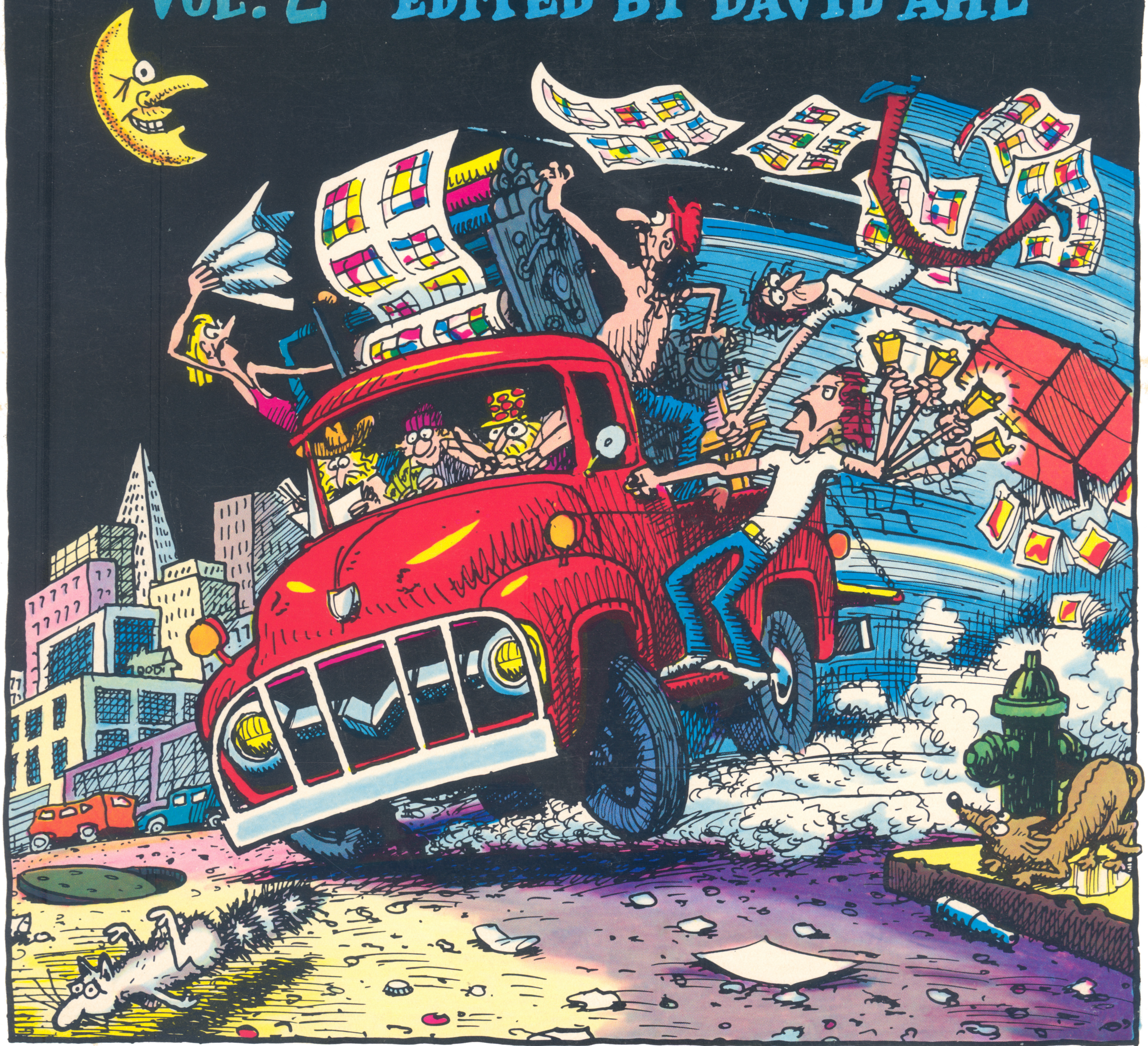
THE BEST OF creative computing

$8.95

VOL. 2 EDITED BY DAVID AHL



ISBN 0-916688-03

The Best of Creative Computing
Volume 2

ORDERING INFORMATION

Additional copies of *The Best of Creative Computing - Volume 2* are available direct from the publisher for $8.95 plus $0.75 postage and handling ($9.70 total). Outside USA, $10.70.

Also available are *The Best of Creative Computing - Volume 1* ($8.95) and *101 BASIC Computer Games* ($7.50). Add $0.75 postage and handling; $1.75 outside USA.

Subscriptions to *Creative Computing* magazine are available for $8.00 per year (paid in advance) in the U.S. or $21 for 3 years. Outside USA subscriptions are $10 per year or $27 for 3 years.

Creative Computing Press, Dept. BC-2
P.O. Box 789-M
Morristown, New Jersey 07960

The Best of Creative Computing

Volume 2 Edited by David H. Ahl

DEDICATION

To Detta, Darcy, and Derek who hopefully will live in a better world as a result of the intelligent use of computer technology.

Creative Computing Press Morristown, New Jersey

OTHER BOOKS
By DAVID H. AHL

101 BASIC Computer Games
Understanding Mathematics and Logic Using BASIC Computer Games
Getting Started in Classroom Computing
The Best of Creative Computing - Volume 1

BY CREATIVE COMPUTING PRESS

Artist and Computer
The Best of Byte - Volume 1
Amazing, Thrilling, Fantastic Computer Stories
The Colossal Computer Cartoon Book

ABOUT THE EDITOR

David H. Ahl received a BS in electrical engineering from Cornell University, an MS in industrial administration from Carnegie-Mellon University, and has done further work in education at the University of Pittsburgh.

For 5 years, Mr. Ahl was a consultant with Management Science Associates, Pittsburgh, PA and for one year with Educational Systems Research Institute. He joined Digital Equipment Corporation in early 1970 where he originated the concept of "EduSystem" (Educational Computer System). He became Education Product Line Manager and wrote numerous books, instructional materials and papers about the use of computers in education.

In August 1975, Mr. Ahl joined AT&T as Marketing Manager-Education where he has responsibility for the development and marketing of Bell System products and services to educational institutions. He publishes *Creative Computing* magazine as a hobby—it is completely independent of outside support.

Mr. Ahl makes his home in Morristown, New Jersey with his wife Sandy, and three children Detta, Darcy, and Derek.

Second printing April 1978

ISBN 0-916688-03-8
Library of Congress Catalog Card Number: 76-438v.2
Printed in the United States of America

Preface

As I was looking over the contents of this book I was struck by the comparison with Volume 1. During the first year we were feeling our way cautiously along, exploring various educational and recreational aspects of computer usage. We had taken the view that hands-on computer usage was a "good thing" and we were promoting the concept, yet when our first issues were put together home computer kits were only a dream. Hence the majority of the Volume 1 focused on educational computer applications and the computer impact on society with a view toward every person learning to use this powerful tool in an intelligent manner. Of course, we ran a liberal number of computer games and other pragmatic applications programs and activities. But only one 3-page article on building a computer of your own.

Now in Volume 2 the tide has begun to turn. No longer do we have to look to educational institutions being the only source of hands-on computer power. Indeed, it won't be long before there is more computer power in the hands of people at home than in all the schools and colleges combined. In the 1976 issues of *Creative Computing,* from which this Volume was assembled, we have increasingly covered computer kits, the microprocessor, and home applications. Not that we have abandoned educational users; rather we see a merging of traditional distinctions. That is, the home or hobbyist user wants and needs to know more of systems design and theory and program structure (traditional educational topics) in order to use his machine more effectively. On the other hand, schools and colleges increasingly are utilizing microprocessor-based kits, terminals and systems (the home turf of hobbyists), particularly with increasing pressure on budgets.

So this volume is the beginning of the merging of school and home applications. Also, we've tried to give all our readers a sense of perspective with commentary, opinion, and future speculations, both factual and fictional. In addition, lest we get too serious about ourselves or our computers, there is a liberal dosage of foolishness, puzzles, and games — just for fun and relaxation and maybe an occasional mental challenge or two.

Where are we going in the future? While semiconductor developments will certainly have a major influence, I look to our readers as providing the real answer. Indeed, *Creative Computing* magazine welcomes articles, programs, graphics, games, reviews, and other material for future issues. We're especially looking for "how-to" material and articles about building various computer kits. We do not want manufacturer puffery, broad generalizations, or writing bogged down in technical abstractions. But perhaps most of all, we're seeking software and applications — real, live, pragmatic, how-to stuff, not "gee whiz" success stories that can't be replicated by other readers. Need a crass, commercial incentive? Here it is: we pay for articles. Please include a self-addressed stamped envelope if you want your contribution acknowledged.

I was tempted to comment about the astonishing diversity of material in this volume. Contributors included middle school students, university professors, hippy poets, and corporation presidents. We've run art, philosophy, programs reviews, fantasy, etc., etc. But to comment about all this is silly. Why don't you just turn to the contents, pick your favorite topic, and take off from there. Have fun!

David H. Ahl
January 1977 Morristown, New Jersey

Acknowledgements

With a book as diverse as this one, it's hard to know where to begin with acknowledgements. The material has all appeared in an issue of *Creative Computing* magazine and, of course, most writers, artists, and other contributors are noted on the article itself.

However the contributions of some people went far beyond simply submitting a single piece for publication. I apologize in advance for anyone I've overlooked but these are some people that clearly were central to the production of *Creative Computing* magazine and to this book.

Steve Gray whose probing comparative reviews of 34 books on BASIC made it one of the most popular features ever run. Steve also documented his experiences building an Altair 8800, using a Tektronix 4051, judging the National Student Computer Fair and numerous other things. Given this wide-reaching diversity, it's no wonder that Steve is now Editor-in-Chief of *Creative Computing.*

George Beker whose fantastic illustrations capture the very essence and spirit of *Creative Computing.*

Pete Kugel who monthly receives a huge pile of books which he gets out to reviewers, collects the reviews together, edits them, and gets them back to us.

Burchenal Green, Managing Editor of *Creative,* who assumed a huge burden of assorted tasks so that I had the time to put this collection together.

John Lees, whose interest in philosophy, ethics and a wider range of topics than I have room to list broadened the scope of the magazine beyond what I ever dreamed.

Steve North, with fresh and sometimes unexpected views, with tenaciousness in getting hardware and software to function, and with attention to detail in proofreading has been a tremendous help.

Jim Reagan whose series of articles on non-usual mathematics gave many a reader many an hour of creative fun.

Bob Taylor whose unusual and thoughtful articles showed that literature, music, and other social activities can all be related to computer systems.

Deanna Dragunas whose concern with the effect of the computer on people kept us honest with ourselves.

Bill Cotter whose huge library of computer games yielded many fine specimens for the pages of *Creative.*

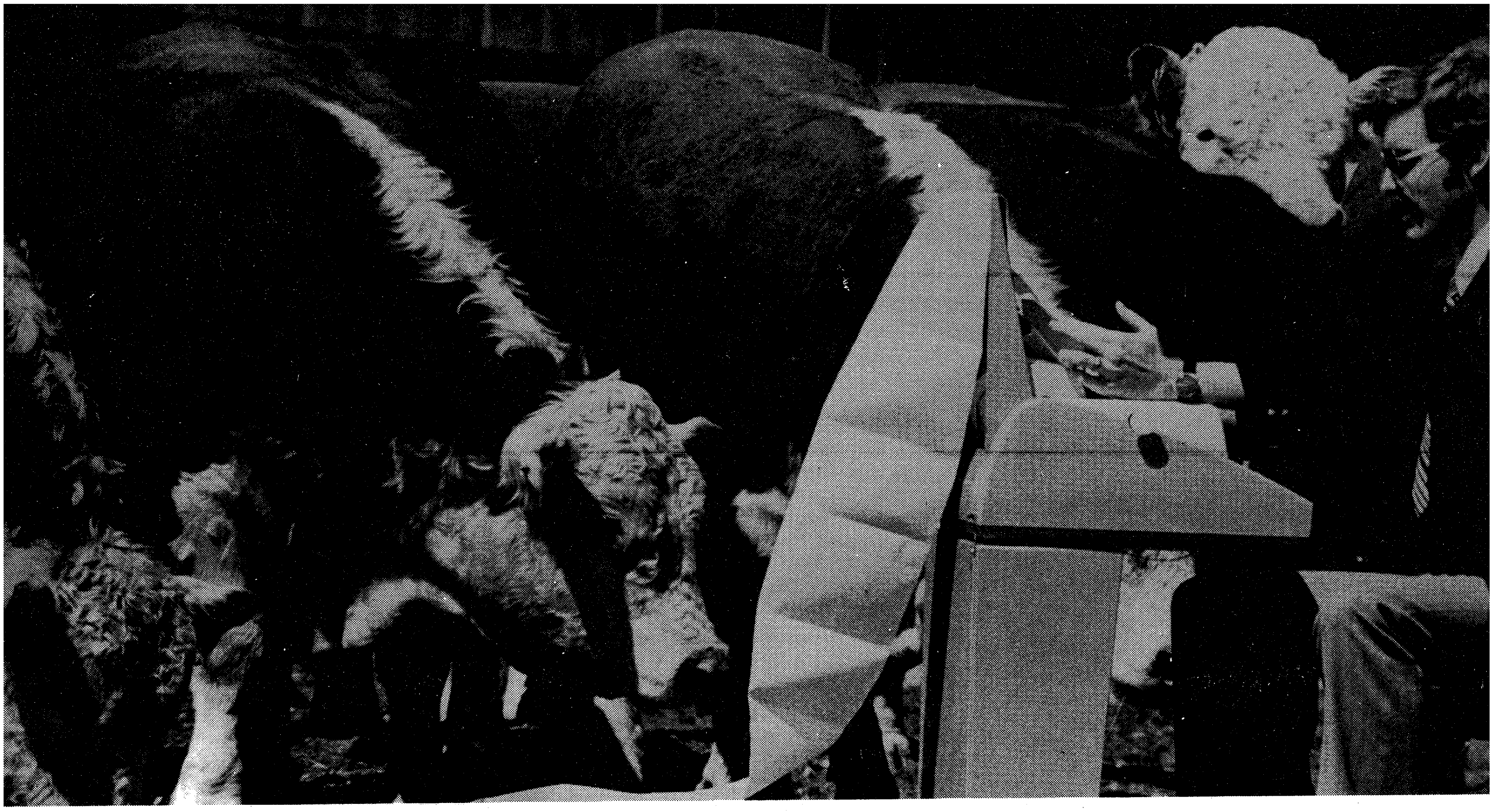
Kent Morton who contacted and kept after many of our educational contributors.

Peter Payack whose incredibly refreshing poetry and short fiction kept us laughing and thinking for hours into the night.

Carol Tick and **Jeanne Tick** whose order processing and handling lets us function as a business instead of a zoo.

Table of Contents

NOTE: Reviews marked with a † are longer feature reviews.



GET THE MOST OUT OF YOUR COMPUTER WITH CREATIVE COMPUTING MAGAZINE

No computer magazine gives you more applications than we do! Games. Puzzles. Sports simulations. CAI. Computer art. Artificial intelligence. Needlepoint. Music and speech synthesis. Investment analysis. You name it. We've got it. And that's just the beginning!

Whatever your access to computer power—home computer kit, mini, time-sharing terminal—*Creative Computing* is on your wavelength. Whatever your computer application—recreation, education, business, household management, even building control—*Creative Computing* speaks your language.

Read through pages of thoroughly documented programs with complete listings and sample runs. All made easy for you to use. Learn about everything from new software to microprocessors to new uses for home computers. And all in simple, understandable terms. And there's still more. *Creative Computing* discusses creative programming techniques like sort algorithms, shuffling and string manipulation to make your own programming easier and more efficient.

We can even save you time and money. Our extensive resource section is filled with all kinds of facts plus evaluations of hundreds of items. Including microcomputers, terminals, peripherals, software packages, periodicals, booklets and dealers. We also give you no-nonsense equipment profiles to help you decide which computer is best for you—before you spend money on one that isn't.

We've got fiction too. From the best authors in the field, like Asimov, Pohl and Clarke. Plus timely reviews of computer books, vendor manuals and government pamphlets. And so much more!

Isn't it about time you subscribed to *Creative Computing?* It's the smart way to get the most out of your computer.

Complete this coupon and mail it today. Or for fast response, call our toll-free hot line.

800-631-8112

(In New Jersey call (201) 540-0445)

I want to get the most out of my computer.
Please enter my subscription to:

creative computing

Term	USA	Foreign Surface	Foreign Air
☐ 1-year	☐ $ 8	☐ $ 12	☐ $ 20
☐ 2-year	☐ $ 15	☐ $ 23	☐ $ 39
☐ 3-year	☐ $ 21	☐ $ 33	☐ $ 57
☐ Lifetime	☐ $300	☐ $400	☐ $600
☐ Vol. 1 Bound	☐ $ 10	☐ $ 12	☐ $ 15
☐ Vol. 2 Bound	☐ $ 10	☐ $ 12	☐ $ 15

☐ Payment Enclosed
☐ Visa/Bank Americard ☐ Master Charge
Card No. ____________________
Expiration date ____________________
☐ Please bill me ($1.00 billing fee will be added; foreign orders must be prepaid)

Name ____________________
Address ____________________
City __________ State ______ Zip ______

Send to: Creative Computing, Attn: Kay
P.O. Box 789-M, Morristown, N.J. 07960

creative computing brings you its best

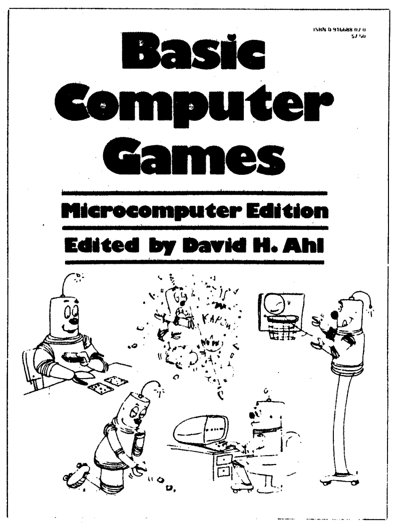
This now classic book contains all the articles, stories, learning activities, games and puzzles that appeared in *Creative Computing* Volume 1—the magazine's first six issues. We've left every bit of the editorial content intact.

Over 200 contributors are represented—from college professor to high school student, from U.S. Senator to underground cartoonist and from corporation president to science fiction author.

The Best of Creative Computing, Volume 1 covers the gamut of computer applications in education and recreation. Its diversity can only be described asstaggering.

"The 'Whole Earth Catalog' of Computers."
American Vocational Journal

New revised edition of our most popular book, *101 BASIC Computer Games*. All you need is a basic-speaking computer.

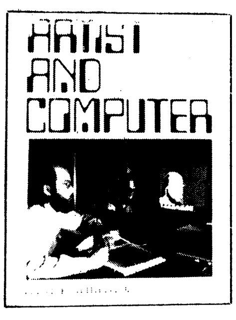


Here are 102 classic computer games, every one in standard microcomputer BASIC. Every one is complete with large legible listing, sample run and descriptive notes.

All the classics are here: Super Star Trek (one of the most challenging versions anywhere), Football (two versions), Black-jack, Lunar Lander (three versions), Tic Tac Toe, Nim, Life and Horserace—to name a few.

Guessing games, matrix games, word games, plotting games, card games, educational games—they're all here. And, they'll all run on your Altair, Imsai, Radio Shack, SWTPC, Xitan, OSI, Poly, Sol, PDP-11 or other micro or mini with extended BASIC.

The delightful cartoons on every page, coupled with highly legible listings, make this revision of *101 BASIC Computer Games* a real must, even if you own the original. 185 pp. $7.50 plus $1.00 postage in USA. $2.00 foreign. Creative Computing. Att: Kay, P.O. Box 789-M, Morristown, NJ, 07960.



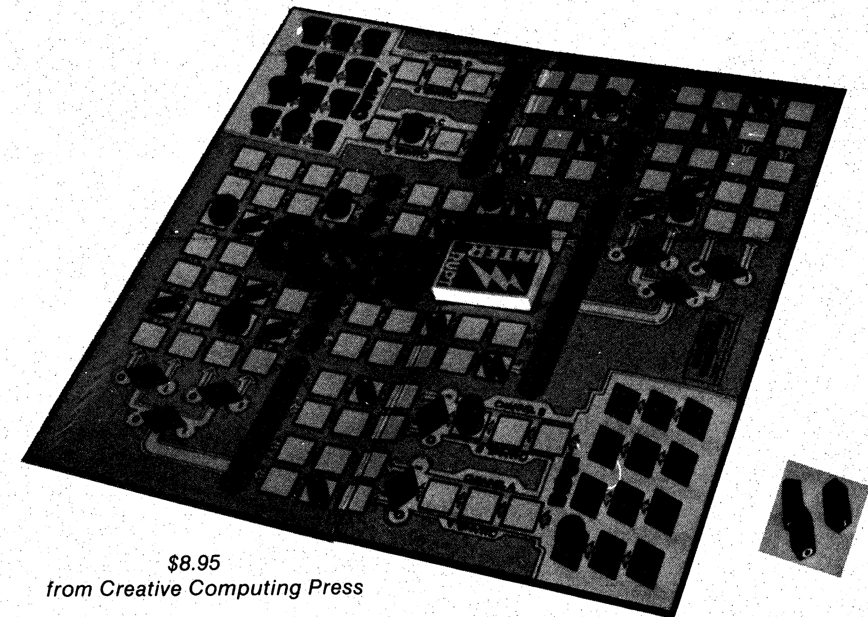
"Get yourself a copy of this book if you enjoy feeding your mind a diet of tantalizing high-impact information."
San Francisco Review of Books.

This unique art book covers a multitude of computer uses and the very latest techniques in computer-generated art. In its pages, 35 artists explain how the computer can be programmed either to actualize the artist's concept (such as the visualization of fabric before it is woven) or to produce finished pieces. Over 160 examples, some in full color. $4.95 softbound plus $1.00 postage in USA. $2.00 foreign. Creative Computing. Attn: Kay. P.O. Box 789-M, Morristown, NJ, 07960.

the best of byte

This is a blockbuster of a book containing the majority of material from the first 12 issues of *Byte* magazine. The 146 pages devoted to hardware are crammed full of how-to articles on everything from TV displays to joysticks to cassette interfaces and computer kits. But hardware without software might as well be a boat anchor, so there are 125 pages of software and applications ranging from on-line debuggers to games to a complete small business accounting system. A section on theory examines the how and why behind the circuits and programs, and "opinion" looks at where this explosive new hobby is heading. 386 pp. $11.95 plus $1.00 postage in USA. $2.00 foreign. Creative Computing. Attn: Kay. P.O. Box 789-M. Morristown, NJ 07960.

not for kids only



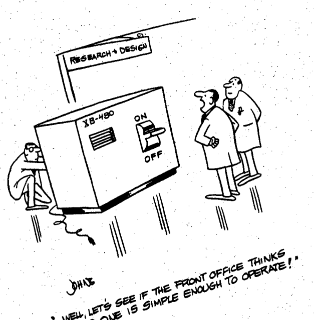
$8.95
from Creative Computing Press

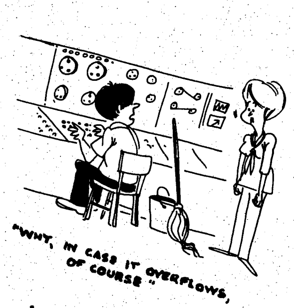
Computer Rage

This fun and educational new board game is based on a large-scale multiprocessing computer system. The object is to move your three programs from input to output. Moves are determined by the roll of three binary dice representing bits in a computer. Hazards include priority interrupts, program bugs, decision symbols, power failures and restricted input and output channels. Notes are included for adapting game for school instruction. A perfect introductory tool to binary math and the seemingly-complex computer.

Binary Dice

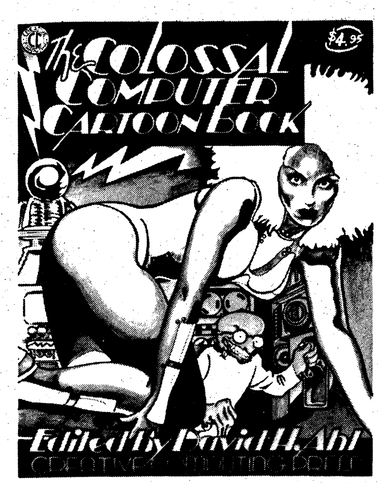
Now, the same dice used in Computer Rage can be purchased separately. Three binary dice (red, green and blue) in a zip-lock bag. $1.25 postpaid [3G].



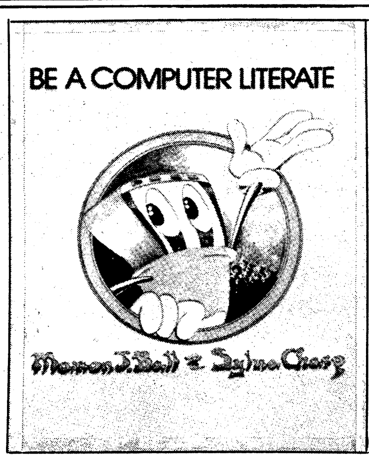


The Colossal Computer Cartoon Book

Take a break. Sit back and
and relax with the biggest and best collection
of computer cartoons ever, hundreds and
hundreds of cartoons about computers, robots, calculators AI and much more.



$4.95
120 pp. softbound
from Creative Computing Press



$3.95
61 pp. softbound
from Creative Computing Press

Be A Computer Literate

This is the most basic, introductory book on computers ever put together for instructional use. Its full-color diagrams, drawings, photos and large, explicit type make this book a pleasure to read. This chapter titles, themselves, best illustrate its contents—

I Introduction
II What Are Computers
III Kinds of Computers
IV What Goes On Inside Computers
V Communicating With The Computer
VI Language Of The Computer
VII How To Write A Simple Program
VIII How Computers Work For Us
Glossary

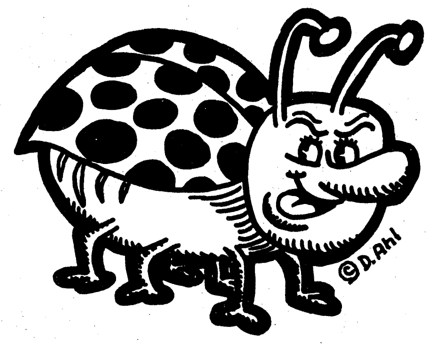
Throughout this book on the section title pages, you'll find . . .

COMPUTER CRITTERS

COLLECT 'EM!

Cut out these cute lil' critters along the heavy black line. Color with crayons, water colors, or "Flair-" type pens. Paste on walls, note-books, IBM machines, etc. Or make a really keen mobile!

LOADS OF LAUGHS!



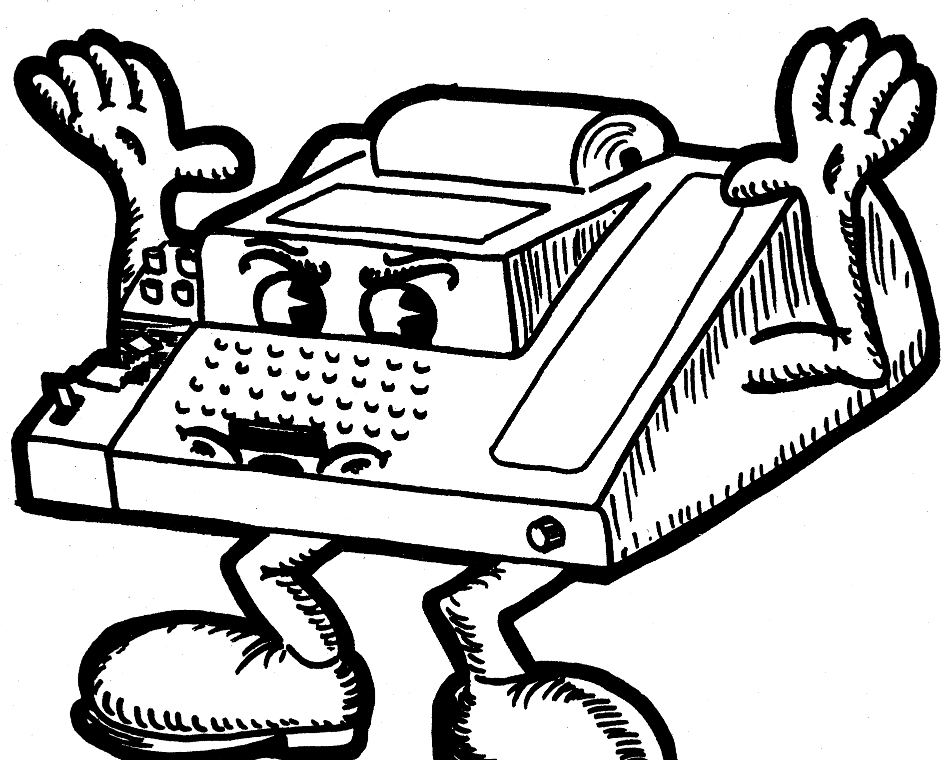
Pester Program Bug

HOURS OF FUN!

SWAP 'EM!

Be sure to collect the whole set

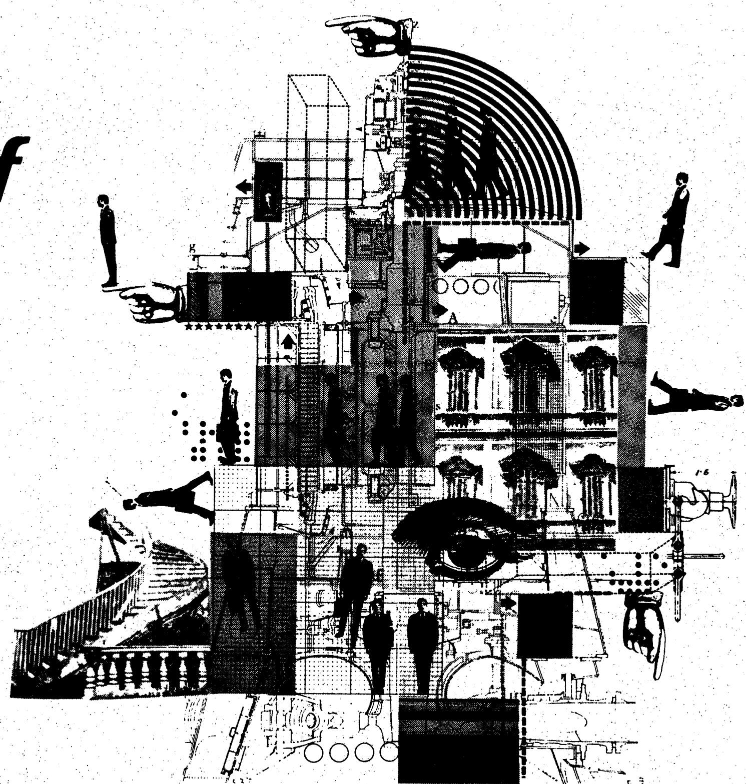
Articles and Commentary



Terrible Tommy Teletype

The Future of Computer Technology

by Deanna J. Dragunas



In a few years, everywhere you turn, a computer will be there to assist, to inform, or simply to play with.

In early 1974, Arthur D. Little, Inc. prepared a comprehensive technological forecast for the Electronic Systems Division of the U.S. Air Force to allow the Air Force to plan for the most efficient use of future data-processing capabilities. That report focuses on what we can expect our computer building blocks and our computer systems to look like in ten to fifteen years. Let's closely examine this techno-forecast.

Three computer building blocks are foreseen, each a different scale of processor. The smallest processor will be similar to today's microprocessor. It will have a small programmable read-only memory to start with; as time and technology advance it will become increasingly more sophisticated and powerful. This processor will be manufactured on one or a few semi-conductor chips as are today's microprocessors.

The second building block will resemble today's small computer system or I/O channel controller. We could call it a complex microprocessor, put together like the smallest building block processor, but considerably more complex.

The third building block processor will be a central processing unit comparable to that in today's medium to high-priced computer systems, but with the speed of today's most complex, largest computers.

Using these three building block processors, four distinct end-user system types are predicted. The first and smallest will undoubtedly have the greatest impact on day to day life, an impact greater than that of the now readily-available pocket calculator. This smallest computer system, the microcomputer, will have one of the smallest processors in it. The microcomputer will be used as an intelligent terminal at first, but then will progress to a small stand-alone system. Microcomputers will be found in televisions, telephones, automobiles, lawn-mowers, calculators (which by then will really be small computers), and in a variety of other such equipment.

The second end-user system, the minicomputer system, will be composed of a single mid-level building block processor and a number of the miroprocessors. This will be a whole computer system which will outstrip today's smaller and medium-sized computers. The simple microprocessors will be contained in the peripheral equipment and terminals, and the complex microprocessor will be the heart of a system which will be able to handle interactive applications and a single background batch stream. This will be the most widely-used computer. This will also be the kind most hobbyists will build in their basements.

The third level of computer system will be a large batch-processing system. Of course, some interactivity will be possible, but this will be primarily a payroll processing, number-crunching, report-generating system. It will have a single or perhaps two central processing units of the largest building-block type, several complex microprocessors for high volume peripheral control, and a scattering of simple microprocessors where needed.

The multi-computer will be the largest computer system. It will be a network of computers within itself with up to four or more of the largest cpu's and as many of the smaller processors as necessary for efficient service for many different users in a number of modes and environments at the same time. For instance, some users may do interactive programming while others run batch processing in emulations of obsolete computer systems, while still others retrieve information from the data bases managed by the multi-computer system. The system will have extremely large data bases and a highly complex executive or operating system to orchestrate its technological complexity. Dial-up data bases for a variety of needs, from menu-planning to medical information retrieval, from trip-planning to news distribution, can be served by such computer conglomerates.

In a few years, everywhere you turn, a computer will be there to assist, to inform, or simply to play with.

Computing Power to the People— A Conservative Ten-Year Projection

by Tien Chi Chen*
IBM San Jose Research Laboratory
San Jose, California 95193

ABSTRACT

The wide availability of low-cost computing power through LSI should lower the communication barriers between the machine and the human user. Intelligent terminals will intercede between the machine and the programmer. The nonprogrammer user will be able to make useful queries of general data bases. Man-machine interaction should be the fastest growth area in the coming decade. The most important development in education should be to expose younger minds to realistic problem-solving.

1. INTRODUCTION

The computer is said to have heralded the second industrial revolution, towards freedom of mankind from drudgery. This freedom has not been realized; the majority of the public have no direct contact with computer systems, and machine users are still adapting themselves to the machine.

Large-scale integration (LSI), or more appropriately, low-cost microelectronics, promises to make a definite start during the next decade.

2. LOW-COST MICROELECTRONICS

Information in the abstract has no mass; its processor and storage can reside in the tiniest physical carrier. But probably the most optimistic computing prophets in 1950 were unprepared for the tremendous shrinking of circuit size in the past quarter-century. The consequent sharply lowered manufacturing cost, in a competitive industrial environment, has led to a drastic price reduction for machine intelligence.

This is most evident in the field-effect transistor (FET) technology, where high density, low power consumption, and low production cost in small packages have brought the pocket calculators, selling at a mere 2% of the price of their counterpart a decade ago, the desk-top calculator. FET microprocessors executing five hundred thousand instructions per second have long been available. Recently the memory density of five million bits per square inch has been reported for a 8192-bit chip, which dissipates 25 milliwatts with a 90 nanosecond access time (1). The trend towards higher density, greater speed and even lower cost is continuing unabated.

A generalized definition of LSI should include the high density, low cost "electronic disks," which use no mechanical parts and hence enjoy much faster access than their mechanical counterparts. Candidates for the electronic disks include charge-couple devices and magnetic bubbles, both under active development.

*This paper represents ideas and opinions of the author, and does not reflect any IBM plans or strategies.

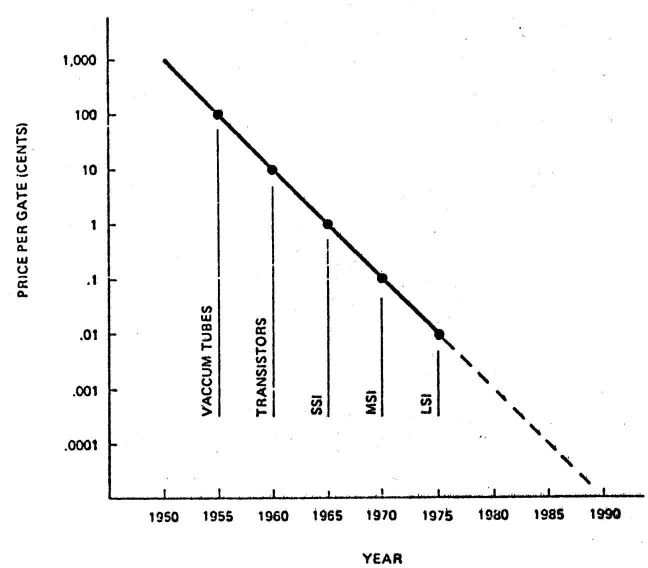
While it is safe to predict the occurrence of at least one unanticipated revolutionary idea, it is extremely difficult to pinpoint this occurrence.

3. A CONSERVATIVE PROJECTION

The following is a projection of the computer scene for the next decade, assuming the steady progress of LSI. There is no intention to cover all aspects of computing evenly; for a more comprehensive projection see (2).

While it is safe to predict the occurrence of at least one unanticipated revolutionary idea, it is extremely difficult to pinpoint this occurrence. The projection here, based only on extrapolations of known trends, is necessarily conservative. On the other hand, the assumption that most of the software problems will be solved in time, is not shared by many expert programmers.

Claims will be made that the wide availability of compact computing power will lead to global equipment upgrading, increased rapport between programmer and machine, large data base for untrained users, effective man-machine interaction for intellectual tasks, and general mind-sharpening in the classroom.



The cost of logic gates is on an exponential decline.

Man-machine interaction, combining the best of the twin worlds of computer precision and human perception, should become the fastest growth area of computer usage during the next decade.

4. GLOBAL EQUIPMENT UPGRADING

Machine intelligence during the next decade will be available at very low cost, lower, in many cases, than the cost of the other devices within the same piece of equipment.

This implies a global upgrading of current "dumb" equipment into "intelligent" equipment, at a small incremental cost. There could also be a corresponding downward price change, but never below the dumb equipment minimum.

Calculators with advanced features (trigonometric functions, memory, and programmability) will only be slightly more expensive than, and will therefore replace, the minimum four-function variety; this trend is quite visible today. Much more importantly, intelligent terminals, housing small computers as subsets, may not cost much more than those with a bare keyboard. This upgrading will be more striking in optical display terminals, where an added intelligent buffer will greatly enhance their ability to handle colors and complex picture-processing algorithms.

Micro computers should approach, and in some respects exceed, current minicomputers in sophistication and performance. Whether this will significantly lower the price of small machines will depend on the attached equipment.

For larger machines, peripheral equipment has long been the hardware cost-determining factor. Upgrading here should lift the system into a new performance category. We can expect the electronic memory to grow in size by more than one order of magnitude; working hand-in-hand with a very fast cache memory, the combined effect is a superlarge, superfast memory, capable of extremely complex management chores, not the least of which is the effective handling of electronic disks and other storage devices to form powerful virtual storage systems.

Micon MCM data terminal measures about 9" square and runs on rechargeable batteries.

There will be upgrading of the I/O and communication interface. Large machines will communicate freely through networks, satellite, or packet radio. When a terminal deals with a "central machine," the latter may actually be a collection of computers reacting correctly under a uniform communication protocol.

It is assumed that most of the outstanding software problems and bottlenecks will diminish through the added LSI computing power, and new systematic programming practices. New software will pay particular attention to data management algorithms and the human interface.

5. LOWERING THE USER-MACHINE BARRIERS

It has been estimated that

$$\frac{\text{the cost to program and debug a line of code}}{\text{the cost to execute the line}}$$

has now reached the astronomical value of 100 million (3). Clearly in a typical installation, the most expensive component is the human cost, which should now be minimized at the expense of machine time. Indeed, human convenience should be maximized whenever possible.

The relationship between the programmer and the machine has seen ups and downs. In the early days of computing, users had physical contact with the machine in order to push the appropriate buttons, but had to state their needs through the unwieldly machine code. The advent of FORTRAN and other procedural languages permitted programming on human terms, but the user was soon ejected from the machine room and had to communicate through a batch-centered job-control language.

The advent of terminals and time-sharing has helped the user to reassert himself, under the desirable illusion of direct machine involvement. But there still remain complex sign-on procedures, difficult control statements varying from layer to layer, incomprehensible error messages, unexplained delays, also unexpected system crashes, destroying the work of innocent users.

The intelligent terminal, provided with powerful monitoring programs, can go far to serve as go-between, much as a resourceful receptionist mediating between an executive and a visitor. The work includes expanding simple sign-on codes into the proper format, explaining unusual happenings, catching and fixing simple errors, keeping statistics, recoding and storing locally for safekeeping security and economy. Small jobs can certainly be handled locally, from start to finish.

With sharply lowered machine cost, interpretive computing on terminals will become common for small problems, especially for students. The conventional compiling process introduces an extra layer of problem transformation into the job, and is a source of misunderstanding. On the other hand, it is easy to learn the use of interpreters. Further, on a terminal every interpretive step can be monitored in terms directly meaningful to the programmer. Compiling and batch processing can be reserved for time-consuming programs, as an economic measure. Optimum interpretation, involving the real-time balancing between interpreting and compiling, should become a reality.

6. DATA BASES FOR NONPROGRAMMER USERS

The computer, far from freeing the average citizen from drudgery, actually generates some resentment in him, because he has no direct use of the computer, yet is often the recipient of its less-desirable by-products, such as wrong bills and junk mail.

With sharply lowered machine cost, interpretive computing on terminals will become common for small problems, especially for students.

Data base query systems do not demand expert programmers as users, and the data base itself could contain material of high interest to the general public. System software delivered thus far has been well received. The future LSI-boosted machine, embodying the fruits of data base research, assisted by the intelligent terminal, should drive down the cost and time per query, to within reach of the layman; and data bases then need no longer be reserved exclusively for executives of large firms. One should see very large, efficient query systems open to the general public, accepting queries in a restricted form of English. Such an effort may require initial sponsorship by the Federal Government.

With data base techniques firmly in hand, the office terminal and the home terminal cannot be far behind. Eventually, a proliferation of display terminals may allow serious attempts at the elimination of printed hard copies, before the decade has run its course.

7. MAN-MACHINE INTERACTION

Low cost machine intelligence will stimulate artificial intelligence research, but probably not enough to solve some of the nagging problems in the field in a decade. True breakthroughs may need to await the new structuring of entire ensembles of logical devices, to create either a semblance of, or an alternative to, biological intelligence. Such breakthroughs will surely be tried, but success cannot be presumed at this time.

Thus problems in natural language comprehension, voice and handwriting recognition, language translation, theorem proving, and deductive reasoning probably will remain incompletely solved.

However, the computer can handle simpler aspects of these problems; human help can be enlisted for the harder ones. Even in conventional data processing, whenever the machine is stuck because of insufficient information or too many alternatives, it can try to supply to the user reasonable guesses and their dire consequences, or just ask a pertinent question. The human then provides the needed direction.

Man-machine interaction, combining the best of the twin worlds of computer precision and human perception, should become the fastest growth area of computer usage during the next decade.

The lack of training of the user here is almost never a problem, the difficulty lies in equipping the machine to handle human-intuitive concepts. The most natural communication channels are of course sight and sound. Voice input may remain limited, but voice output can be quite general. Image processing and computer graphics will permit efficient two-way communication, and intelligent optical display terminals will be wide-spread, many of these in color. The possibility of man-machine dialogue without resorting to written messages should profoundly affect the use of computer systems, especially in education.

Initially, these man-machine projects should rest upon applications, to lend proper weighting of priorities. General man-machine intelligent processing may then be distilled from a number of successful applications.

8. REALISM IN THE CLASSROOM

The wide availability of low-cost computing power, even in conventional packages, can be of untold benefit to education. This is true not merely in the obvious areas such as research in computer sciences, fulfillment of explicit computing needs, or preparation of tutorial material. The computer should be exploited on a large scale as a new vehicle to challenge and sharpen younger minds, by exposing the latter to realistic problem-solving situations.

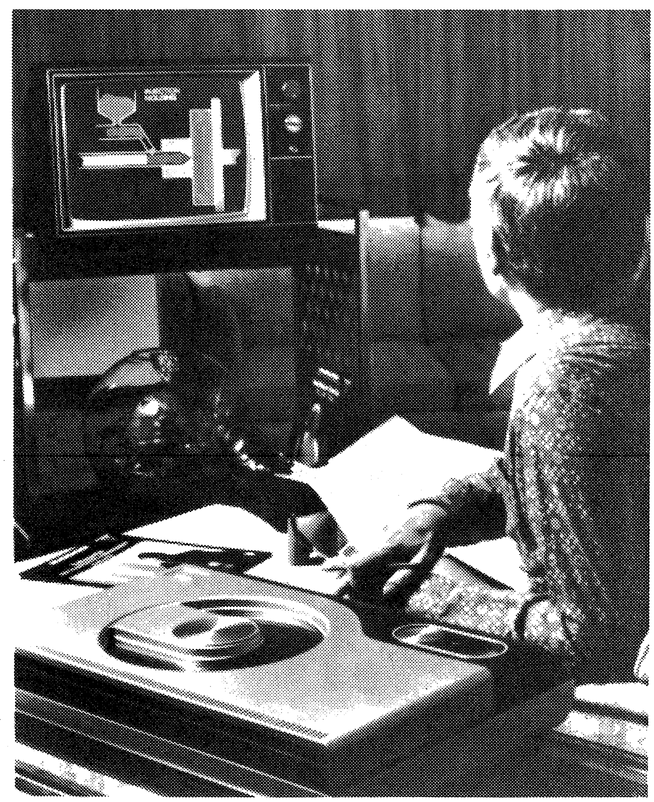
With the arrival of the pocket calculator, examinations in freshman physics no longer need to confine triangles to artificial side ratios such as 3, 4 and 5. A more subtle form of artificiality remains in education, however, in the nature of the problems being posed and solved in schools.

Students today are given only well-formulated problems, the solutions of which requiring a small number of steps. The graduate soon finds, however, that realistic problems seldom come gift-wrapped, with attached answer sheets. Indeed, the identification of the problem is often the major aspect of real problem-solving.

The culture shock of the new graduate will be greatly lessened, if realistically complex situations are treated regularly in the classroom.

Professor Kemeny has pointed out that just the exposure of a student to a data base of statistical facts can stimulate him to draw and verify tentative conclusions (4). Such inductive reasoning is the main ingredient in problem identification. The possible consequence of a problem-solving step can be seen explicitly using a parameterized model.

The ancient Greeks treated geometry as a mind-sharpening device; in the computer we have a new, vastly more powerful tool. An excellent way to practice multilevel reasoning, for example, is by programming and running a computer. The student soon learns to expect punishment and rewards as the multilevel consequence of his decisions. He also learns to separate the forest from the trees, and to shift his forms of attention from one to the other, never losing sight of the final goal. This way he acquires the



Videodisc players coupled with a microprocessor and keyboard will make tremendously powerful and flexible educational and recreational devices in the 1980's.

The ancient Greeks treated geometry as a mind-sharpening device; in the computer we have a new, vastly more powerful tool.

technique to build large structures from smaller modules.

The programmer-student acquires a *modus operanti* for problem-solving. He plans and weighs his actions, making allowance for unforeseen events, and balances the initial programming cost with subsequent debugging effort. He evolves to be a more perceptive, better balanced individual, with a deeper understanding of the machine *as well as himself.*

All of these could have been done ten years ago, but the cost has thus far limited the education through computing to a privileged few. With the projected low cost, one should expect the majority of students in the future will have extensive computing experience. Using optical displays, preferably in color, the computer education process can already begin before the student can spell correctly (5). There should be no compunction about dressing up the optical terminal as a colorful game machine; anyway it is hard to tell where pure play ends and serious learning begins.

9. CONCLUSIONS

The LSI revolution makes machine intelligence generally available at all levels. This chance should be seized to bend the machine to the user.

The programmer should no longer need to learn the idiosyncracies of the machine operating system. The LSI-boosted machine and the intelligence terminal can even adopt human communication channels and rudiments of human semantics to perform semi-intelligent tasks, for human approval or overrule.

The untrained layman should now have the chance to use the computer for queries into very large data bases for facts relevant to himself. Only this way can we claim the war against drudgery is being won in the face of a computer-fueled information explosion.

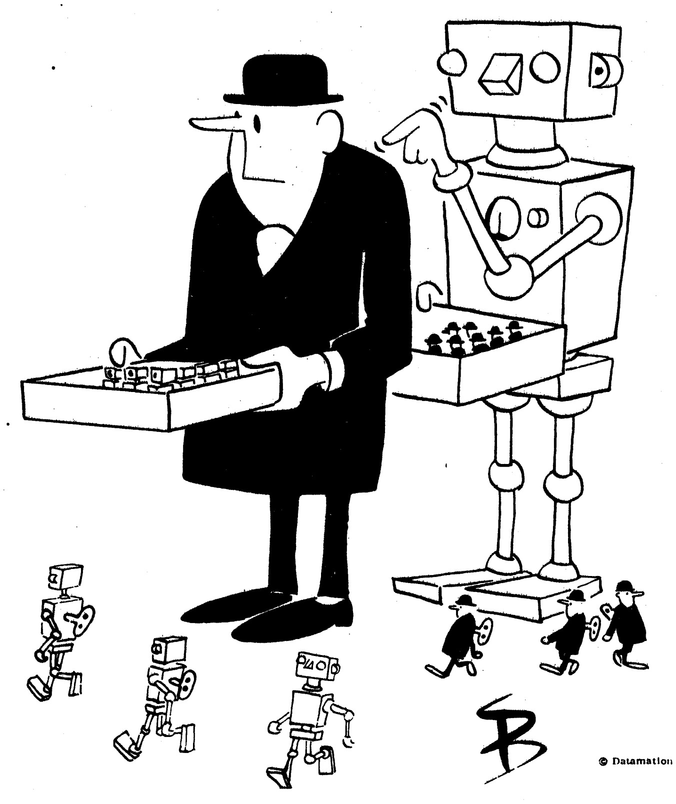
A classroom with abundant supply of computing power can expose the student to realistic problems and encourage him to solve them systematically. He thus becomes better equipped for decision-making in a complex society.

ACKNOWLEDGMENT

The writer is indebted to his colleague, Dr. Juan Rodriguez-Rosell for constructive criticism.

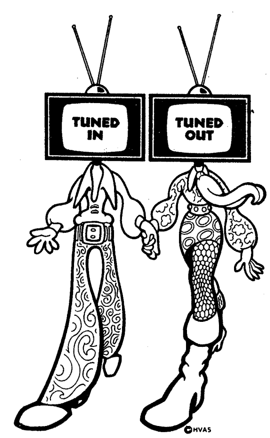
REFERENCES

1. H.N. Yu, R.H. Dennard, T.P. Chang, C.M. Osburn, V. DiLonardo and H.E. Luhn, paper presented at the 13th Symposium on Electron, Ion and Photon Beam Technology, Colorado Springs, May 1975. See Electronics, May 29, 1975, pp 46-47.
2. J. Rodriguez-Rosell, "Trends in hardware and software technology, the 1980 environment," a report from the Center for Computer and Information Sciences and Division of Applied Mathematics, Brown University, Providence, R.I. 1973. 54 pages.
3. E.C. Joseph, Session Chairman's remark at the IEEE Workshop "From calculators to minis," San Francisco, California, February 1973.
4. J.G. Kemeny, *Man and the Computer* (Charles Scribner's Sons, New York 1972).
5. For an early discussion see R.W. Bemer, *Computers and Crisis* (ACM, New York 1972), pp 59-70.



Televisionism Manifesto No. 1

(Selected Excerpts)



by Phil Smith

1. Anything can and will happen.
2. Well
3. Well
4. I am alone and I am not alone.
14. There is no such thing as best.
16. Value is a matter of opinion.
17. Value is not Worth.
19. Dream Life is Vertical
20. Real Life is Horizontal.
25. What I am DOING IS important.
40. Worth is inherent in all things Real and Unreal.
52. Worry is a Horizontal slamming into a Vertical
57. Dream Life is a Fiction.
58. Real Life is a Friction

Phil has written a number of "Alternative Press" booklets. He is currently involved publishing the "Gegenschein Quarterly" which features the work of one artist in each issue. Sample copy $3.00. Phil Smith, 350 East 9th St., Apt. 5, New York, NY 10003

Videodiscs — The Ultimate Computer Input Device?

by Alfred M. Bork
University of California, Irvine

I'd like to discuss with the readers of *Creative Computing* the extremely exciting possibility concerning the marriage of video disc and computer technology, particularly microcomputer and graphic technology, with emphasis on educational applications.

First, it will be necessary to describe the videodisc system, as it is unknown to the general population as well as to people in the computer world. The development has been going on during the past five years or so by a number of major manufacturers around the world. The basic idea is to provide an inexpensive way of showing video through everyone's home TV. The developers hope to sell the unit to many of the people around the world who have television systems, and plan to sell "records," containing films and other things, to be played through home television sets. It should be noted that this is an extremely large market, so all the economies of scale available through large numbers would be present.

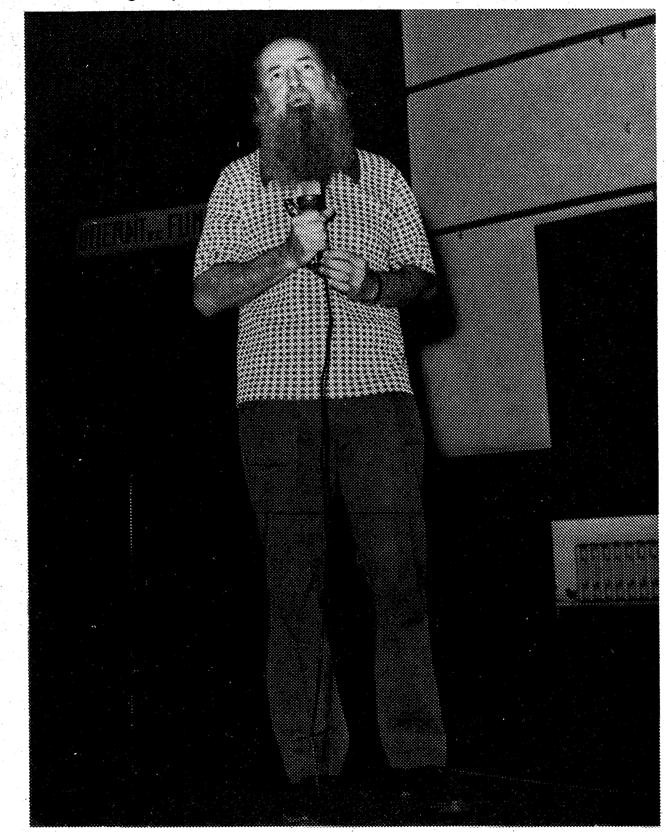
The leading system presently is that developed by Phillips-MCA, although other companies are active in developing similar systems. They have often been demonstrated publicly for several years, and the systems do work. The recording is *not* magnetic—one of the reasons for this development is that magnetic tape recorders for home video have proved too expensive and too subject to problems. The recording media is more like that used in ordinary records, with a number of candidates being used. One possibility is simply pressed vinyl, exactly the same technology used with ordinary records, but with much closer grooves. In many of the systems the grooves are read by laser beams of light modulated by the shape of the groove, rather than mechanically. Typically the discs are rotating at 1800 RPM, so that one track around has the information for one TV picture. Times of approximately one hour of video per disc side are suggested by the developers.

At first glance the videodisc system, while exciting in its own right, does not appear to have any direct connection with computers, and particularly with educational uses of computers. The developers of videodisc systems are not fully familiar with the possibilities of very wide-scale computer-assisted learning systems.

I've long been convinced that the ultimate and inevitable outcome of microcomputer technology, evolving often extremely rapidly, will be that we will eventually find all the processing capability for computer-based learning systems directly in the device itself. While such a system might occasionally connect to a remote computer, for access to large-scale databases, or rapidly changing databases, or for access to massive computational capabilities, these systems would mostly function alone.

This type of system has tremendous advantages. First, a breakdown affects only that particular system; second, no communication costs are involved; and third, none of the present limitations of communications speeds over ordinary phone lines (usually for economic reasons) are present. Fourth, these systems allow highly interactive graphics, because they do allow very rapid speeds, TV bandwidth, impossible today from a practical point of view for remote displays. I believe that graphics is an important component of all educational systems, and it seems to me that the stand-alone system, with its own built-in processing capability is the natural solution with present technology.

The stand-alone system very much needs the videodisc for a variety of reasons. The videodisc used in the stand-alone system, however, would be somewhat different from the one for the home TV system, particularly in terms of the type of information stored on the disc. One of the most interesting aspects of the disc is its very large capacity. An



IFIP World Computers in Education Conference attendees hear about graphics and videodiscs from Alfred Bork.

hour's worth of TV corresponds to about 10^{11} bits of information. Discs are already randomly accessible although at too slow a rate. It does not seem to be too great an engineering problem to increase the ability to access randomly an area of the disc given that all that is necessary is to move a light beam.

I envision that each disc will contain a complete multimedia teaching package. Thus, a particular disc might be an elaborate teaching sequence for physics, having on the disc the computer code for that sequence (including possible microcode to make the stand-alone system emulate the particular machine that material was originally developed for), slides, (one turn around the disc), audio messages, and video sequences of arbitrary length, all of these many different segments. Thus, a teaching dialog stored on a videodisc would have full capability of handling very complex computer logic, and making sizable calculations, but it also could, at an appropriate point, show video sequences of arbitrary length or slides, or present audio messages. Another videodisc might have on it a complete language, such as APL, including a full multi-media course for leaning APL interactively. Another might have relatively little logic, but very large numbers of slides in connection with an art history or anatomy course. For the first time control of all the important audiovisual media would be with the student. The inflexibility of current film and video systems could be overcome too, because some videodiscs might have on them simply nothing but a series of film clips, with the logic for students to pick which ones they wanted to see at a particular time.

The procedure I envision would be something like this. The videodiscs would be prepared by some central sources, either the large educational technology centers discussed in the Carnegie Commission on Higher Education study, *The Fourth Revolution*, or by commercial vendors, perhaps even the current textbook publishers. They would be stamped out by record companies, and they would be sold in stores as ordinary records are. Note that the manufacturing technology for such records is expensive, so there is likely to be little pirating. It is much easier to copy a magnetic tape than it is to produce a new record without access to the master.

Thus, we would have, for the first time in using the computer in instructional ways, a sellable product, difficult to pirate. This would mean that all the usual mechanism of royalties for authors, advertising the materials, etc., would be possible. Students would carry home a stack of records, representing courses they were going to take. The record would be put into a slot in the machine, perhaps using the student's own home TV set and home videodisc unit (although it's not clear that this last would be possible without some modification). The lesson would start up immediately as soon as the start button is pressed.

Although I refer to the device as being in the student's home, it might well be in an educational institution, either a conventional one such as a school or university, or an unconventional one such as a public library. Indeed, one would expect that the records would be available for loan in libraries just as current records are available in many libraries. If record keeping were necessary to insure credit or for taking on-line exams, this could be done either by dialing to a remote computer or by local magnetic storage, perhaps a spearate floppy disc, perhaps a magnetic area on the videodisc.

We should not underestimate the needs of computational capabilities here. It may be that through use of the fixed storage media, the videodisc, we will be able to get by with less "real" storage, and some fast memory will be essential. Some storage will be necessary to refresh a TV screen rapidly; the screen resolution will be very important so that storage will not be trivial.

How do we get all this to happen, and what kind of time scale are we talking about? The time scale seems to be on the order of five years, perhaps a bit longer. The microcomputer technology has perhaps not evolved quite to the point that would make this system economically practical, but it is rapidly approaching such a situation.

The videodiscs are not on the market, and it is possible that there may be competing systems before the issue is clarified. One of the most difficult issues is to bring together people with educational computer expertise and people with expertise in the videodisc technology. Perhaps the first demonstration systems will be developed in university laboratories, or in collaborations between universities and industrial companies, before companies are convinced of the vast possible mass markets for such systems.

So far computers, although useful, are not playing a major role in our educational system. The vast majority of students, both at the K through 12 level and at the university level, never see computers except possibly in courses exclusively oriented toward teaching programming. So the full potentiality of the computer for revolutionizing the way students go about learning is not yet fully appreciated. Nevertheless, this effectiveness is real, and views of the future of learning which do not include extremely heavy use of the computer are inadequate. While one can develop various views of the future (the one in George Leonard's book, *Education and Ecstasy*, is an appealing possibility, somewhat different from the one suggested here), the prospects are nevertheless exciting. Let's get to work on it!

VIDEODISCS
A PARTIAL BIBLIOGRAPHY

"Videodiscs. The expensive race to be first." *Business Week*, Sept. 15, 1975.

"The Videodiscs are Coming." *Science News*, Sept. 27, 1975.

"Videodiscs." *IEEE Spectrum*, Aug. 1975.

"Video's New Frontier." *Newsweek*, Dec. 8, 1975.

"Video in the Round." *Time*, Oct. 17, 1975.

"A Review of the MCA Disco-Vision System" by Kent D. Broadbent. SMPTE Technical Conference, April 26, 1974.

"MCA Disco-Vision" (booklet) 1975.

"A Home TV Revolution" by Robin Lanier. *The New York Times Magazine*, May 25, 1975.

"Round and Round They Go." *Technology Review*, Oct/Nov 1975.

"Videodiscs. The Ultimate Computer Input Device?" by Alfred Bork. *Creative Computing*, Mar/Apr 1976.

"The $2.98 Computer Library" by Arthur Luehrmann. *Creative Computing*, Mar/Apr 1976.

Round and Round They Go

Even a child could operate the video-disc players to be marketed next year, say the manufacturers. The players, which feed television signals from 60-minute phonograph record-like discs into television sets, could allow unprecedented freedom in home television viewing (photo courtesy Philips-MCA).

One pitched battle in the consumer electronics industry over the next few years will be between RCA Corporation and a partnership of MCA, Inc., an entertainment conglomerate, and N. V. Philips, a Dutch electronics corporation.

These two competitors plan to market systems late next year for playing phonograph-record-like video-discs on a $500 player attached to the home television set. The possibilities for such a system are enormous. Not only could viewers select any program they wished, no matter how esoteric (or erotic), but home study would boom. Do-it-your-selfers could actually *see* how the whatzit attaches to the whozit; medical students could play and replay their favorite operations; massive amounts of information could be cheaply stored on the high-capacity discs for use in the home, government, industry, or academe.

The catch is that RCA and MCA-Philips have come up with two different and incompatible methods for home TV records. RCA relies on a sensitive needle tracking a tiny spiral groove as the record spins on a 450 r.p.m. turntable. The video picture and sound signals arise from the changes in electrical properties as the stylus speeds through the grooves.

Philips and MCA combined their formerly rival technologies to develop a system based on a laser beam in the home player. As the disc spins at 1800 r.p.m., the finely-focused blue beam bounces off a succession of tiny pits arranged in a spiral on the record's surface; the resulting reflections constitute the signal for the television. In producing a disc that could store about 30 minutes of television per side, both RCA and MCA-Philips have developed systems capable of storing tens of billions of information bits, and both can give stereo sound along with a high-quality color picture.

The systems represent remarkable achievements, and certainly promise to out-perform the once highly touted videotape players, which have settled at prices too high for the average consumer — $1000 for a player and $30 per 30 minutes of taped program.

The two companies are making subtle and not-so-subtle jabs at one another even before entering the marketing ring. For instance, RCA claims that its system will be reliable and cheap because it is fabricated from conventional components that have been on the market for many years. The sturdy stylus can be replaced as easily as a phonograph needle. No complex beam-aiming mechanism is needed for the needle-in-groove system. And the lower rotational speed significantly reduces the possibility of vibration in the system, say company spokespersons.

Philips-MCA counters with the assertion that all its components have been mass-marketed for years: advanced optics systems in cameras; integrated circuits in computers, lasers in office, military, and space equipment; and high-speed discs in computer storage units. Optical equipment allows higher storage capacity, say company engineers. Philips-MCA has achieved a storage density that could permit up to 60 minutes playing time per side. Because nothing touches the disc and the "pits" are protected by a layer of plastic, the record will last indefinitely. On the other hand, RCA video-discs lasted through about 500 plays, as does the RCA stylus; RCA says that's as many times as anybody would want to play anything anyway.

Philips-MCA has another ace up its sleeve: although both systems can scan the record to replay a desired segment, only the Philips-MCA laser system can freeze the picture — by scanning the same groove again and again. This means that, with the high information capacity of discs, huge amounts of printed information could be put on a single video-disc, with one page per picture frame. The user could search out a page merely by punching in its address on advanced machines to be developed later.

According to the company, the entire Encyclopedia Britannica and all its supplements could easily be stored on a single disc. Philips-MCA systems with computers attached could also be used as teaching machines. The student would proceed through a teaching program, and as his progress warranted, the computer would call up one or another video instruction sequence on the player.

As an interesting aside, Philips-MCA plans to produce laser-read audio records to be played on its system. The scratch-proof disc would allow up to 15 hours of noise-free stereo per side. Remarkably, the disc capacity is so great that each instrument in a 100-instrument orchestra could be recorded on its own separate channel.

MCA also has access to the enormous film library of its subsidiary Universal Pictures, and plans to produce new programming for video-discs once the system is on the market. Whether or not the film library, containing over 11,000 titles, will be an advantage is questionable. Will people pay up to $10 to see movies and television repeats readily available on commercial television? Certainly "Francis the Talking Mule," one MCA offering, will not find a large following.

Whichever system is triumphant, "narrowcasting" — as Philips-MCA calls it — will enable an unprecedented freedom of choice in television viewing. — *D.M.*

The $2.98 Computer Library

by Arthur Luehrmann
Director, Project COMPUTe

A BASIC interpreter plus all the programs in the Dartmouth library, ready to run on your home computer, for $2.98? Space-war, in full color with sound effects and electronic music, right in your living room? A complete, conversational, interactive program to teach you French in 100 lessons—all for $5.00 and ready for your computer?

Does this sound absurdly futuristic? Well, it may not be as far off as it appears. First, let's think about the idea of a home computer. Most people already own about a fourth of one. One's color TV set is a rather good display device and every computer needs one of those.

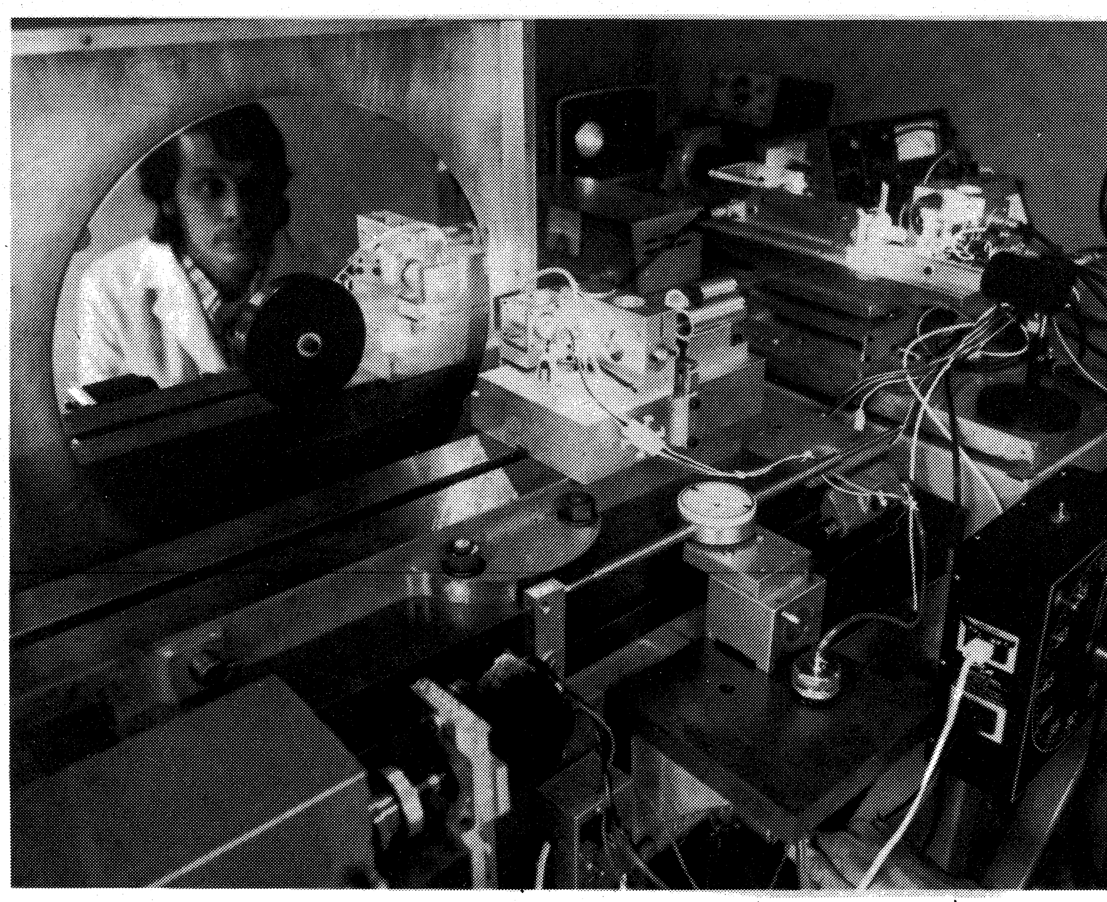
Many people will soon be buying another fourth of their home computer in the form of a videodisc player. Of course, they won't be thinking of it as a piece of a computer, but a videodisc really is a random-access read-only memory with a capacity of about 10^{10} bits. In addition to pictures and sound it can contain computer instructions and data. And it will cost only a few dollars per disc.

Well, that's half the job. The missing part is the processor, memory and keyboard, of course; and it's hard to see why people would want to spend much money for these items. But is it so hard after all? They buy TV sets for their entertainment. They will buy videodisc players for the same reasons. Perhaps entertainment will, in part, motivate buying the rest of the family computer. Everyone knows that computers play pretty good games. People are already paying $100 or more for electronic games like Pong and Odyssey that can only do one or two things. A general purpose game player has all the attributes of a general purpose computer.

The attractiveness of games, augmented by utility computational programs and educational programs will stimulate the spread of the home computer. And the spread of the private computer will create the demand for new software—new games, new utilities, and new courseware. In the course of time widespread demand will provide the economic incentive for authors of software and courseware to write new or improved programs.

The videodisc process may be as important to the publishing and distribution of non-verbal information as movable-type printing was to the written word. In both cases mechanical stamping replaces tracing, whether by monks or magnetic tape recorders.

So don't be surprised in a few years, friends, to find advertisements in the newspaper for Software Specials at your local supermarket. I can see it now: "A Golden Oldie—FORTRAN—a Closeout Bargain at $1.49—sorry, no refunds!!!"



Cutting a videodisc master with a laser. In quantity, discs can be produced for 40 cents although after programming, packaging, royalties, marketing, and distribution, the selling price will be closer to $10. (Photo courtesy Philips-MCA.)

PERSONAL COMPUTERS

Personal computers may now prove to be less expensive and more efficient than time-sharing.

Once upon a time, when computer technology first evolved, users had intimate contact with their machines on a one to one basis. Turnaround was fast, response predictable, and debugging immediate. But unfortunately there was too little computing power to go around, and what was available was too costly for many to afford. When a resource is scarce, it makes sense to share it, even in the face of extra administrative cost. Thus, the concept of time-sharing was introduced.

Time sharing is an idea which has dominated interactive computing for more than a decade. But because of system complexity and integrity considerations, most time-sharing systems offer only very limited access to the capabilities inherent in the machines on which they operate. A time-sharing system's performance deteriorates rapidly with the number of users, and the overhead due to frequent and careful task switching and memory management makes time-shared computing relatively expensive. As the number of users increases, so too does the amount of time the machine spends debugging rather than executing complicated programs.

Overall, the step-wise growth in system complexity made sense while hardware was expensive and software could be cheaply extracted from enthusiastic young people willing to learn. But today software costs are up and hardware costs are plummeting downward. The new computer economy seems to have come full circle. Personal computers may now prove to be less expensive and more efficient than time-sharing.

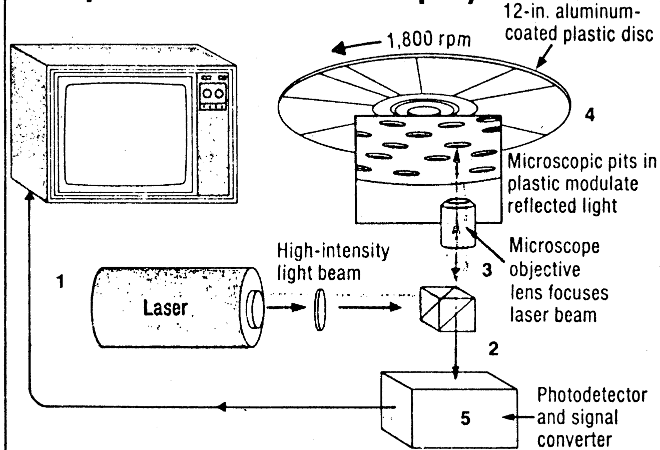
A personal computer is a non-shared system containing sufficient processing power and storage capabilities to satisfy the needs of an individual user. The most advanced of the new personal systems include mass-produced memory and processing modules which can be adapted to a user's specialized needs. Several modules may also be joined and modified to reflect user need, and a group of personal computer modules might even be used somewhat like a secretarial pool, in which each is totally dedicated to a user's need at one particular time, while it resides in a central pool otherwise.

Personal computers, such as the ones being developed now to handle very large programs or to provide educationally-oriented systems, are quite expensive, but after the heavy initial investment in design, turning them out should become much like cutting cookies out of dough. Just as time-sharing originated out of economic necessity, personal computers which can be easily adopted to individual user needs begin to make sense today. It is certainly superior to have 30 systems that give excellent service for 33 thousand dollars each than to have one time-sharing system that cannot adequately support thirty users (doing serious, sophisticated computing) and costs one million dollars. Or for that matter, for specialized tasks how about 200 microprocessor-based systems for $500 each instead of an overloaded 32-user minicomputer time-sharing system costing $100,000? With the new systems, software costs are greatly reduced, and speed and efficiency will prove cheaper and more reliable. Conceivably, as hardware costs continue to decrease, these machines may eventually be distributed like pocket calculators or pencils.

[Adapted from "Personal Computers" by B. Horn and P. Winston, *Datamation*, May 1975.]

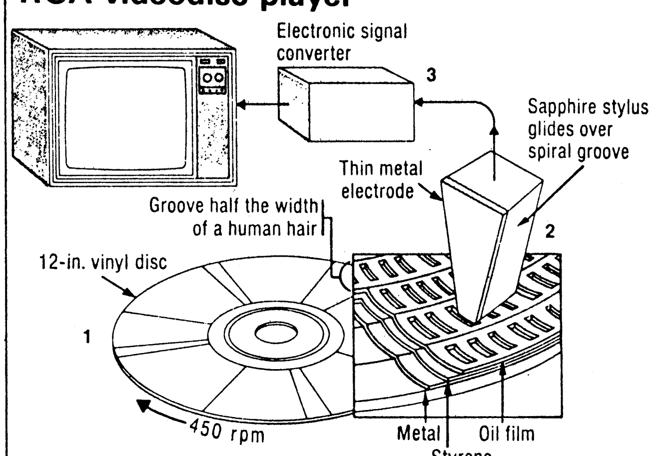
HOW THE VIDEODISC PLAYERS WORK

Philips-MCA Discovision player



OPTICAL PICKUP: A laser (1) generates a light beam aimed by a prism (2) and focused by a lens (3) on a disc (4) coded for picture and sound. Reflected light strikes a photodetector (5) that converts it to signals that are processed and fed to a TV set.

RCA videodisc player



ELECTRICAL PICKUP: The picture and sound code on a spinning disc (1) is picked up electrically by a stylus (2) that transfers signals to a converter (3). The converter processes the signals into electronic form accepted by a TV set.

RUSSIAN COMPUTING— ONE MAN'S VIEW

by
David H. Ahl

On a trip to Russia in April 1974, I took the opportunity to visit the Exhibition of Economic Achievement of the U.S.S.R. Occupying 550 acres, it is similar to a small world's fair. It encompasses some 80 large pavilions each devoted to a different branch of agriculture, industry, or science. It also includes a circular cinema, open-air stadium, and many restaurants, few of which were open during our visit in early April. There was still a fair amount of snow on the ground and what wasn't snow was mud so it was hardly ideal weather for tramping around an outdoor exhibition.

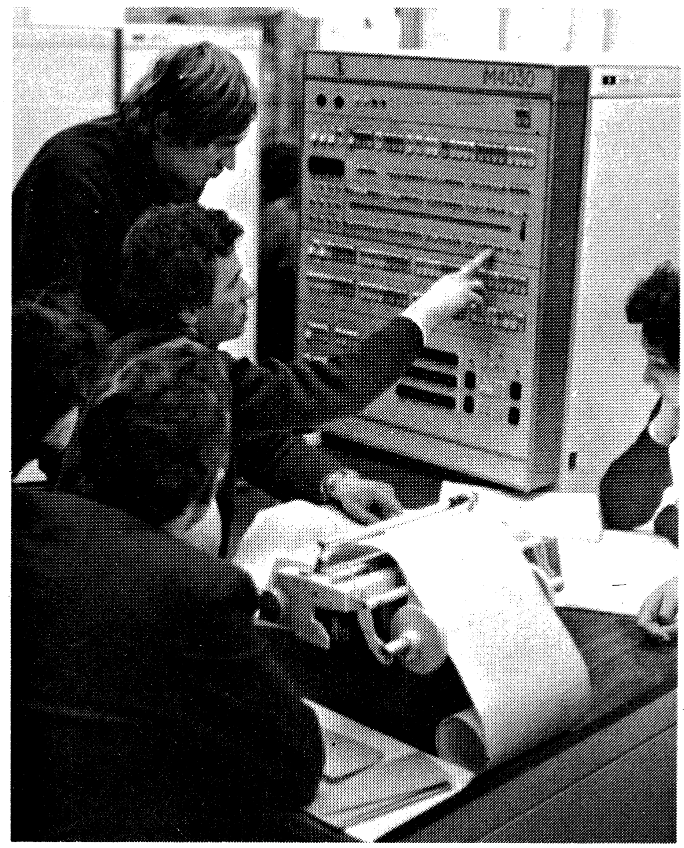
There are pavilions devoted to atomic energy, physics, chemistry, civil architecture, culture, printing, fur breeding, education, public health, radio, space exploration, and many other areas. I visited several of these briefly and found, somewhat to my dismay, that often the three story front facade stood in front of a small one story, 2 or 3 room display. Impressive from the outside, but not much depth.

One pavilion I did visit at some length was the computer pavilion. There were no signs in English, nor could any of the guides inside speak English, hence my account is based strictly on personal observation. There were two large computers in operation, one a batch system (EC-1020) that looked like a cross between a 1401 and 360/30 or 40. The peripherals looked decidedly vintage, particularly the card reader and tape drives. The other system (M-4030) was a time-sharing system with a front panel that reminded me of a flattened PDP-15; the rocker switches were identical. It appeared to use a 32-bit work length. Most of the terminals on the timesharing system were CRTs with quite large screens (12" or more). The terminals were bulky and gave a strong impression of functional utility. The Russian equivalent to our Teletype looks very much like an ASR-37 except the tape reader and punch are nicely recessed to the right of the keyboard. Their 2781 equivalent looks something like an IBM Model B electric typewriter, moving carriage and all. Keypunches look like carbon copies of the 029.

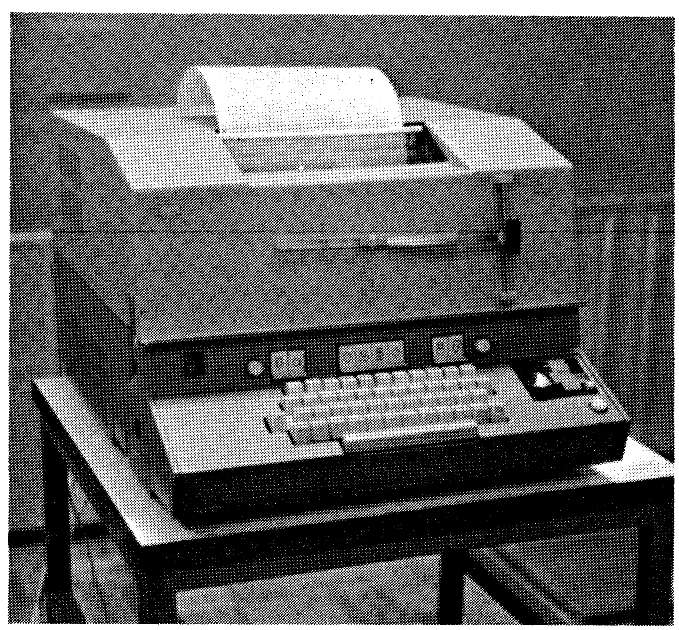
A nice young girl tried to explain text editing to me with much gesturing and pointing. The terminal had almost all the capability of the Dataspeed 40 — scrolling, line insertion, line deletion, etc. When you get your copy OK on the CRT, you press a button and it types out on the attached (local) printer. The major differences between it and the Dataspeed 40 are that intelligence is in the CPU, not the terminal, and the hefty, ungainly size of the unit.

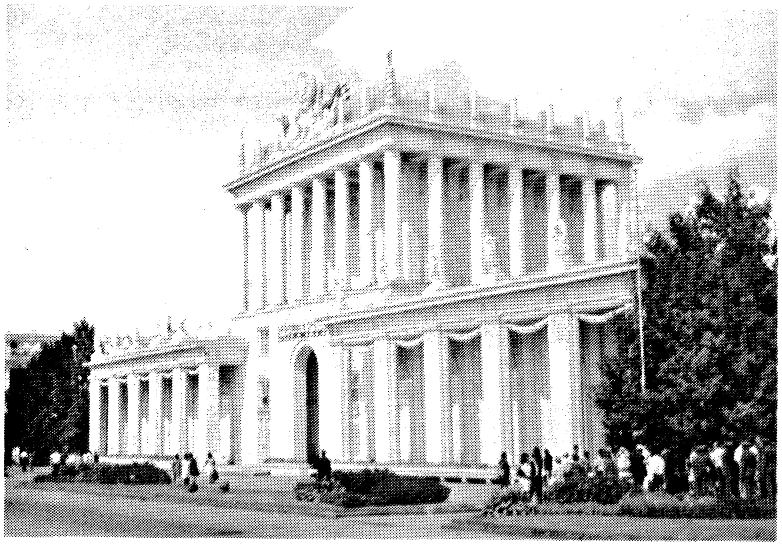
I saw much other hardware, both operational and some just on display. (Most of the center was fully in use—the Russians don't leave expensive hardware unused.) Throughout the display, I saw no signs of any minicomputers or microprocessors. Nor, in my entire trip, did I see a single pocket calculator, not even in GUM, the largest store in Moscow.

I asked one of our guides, a graduate of the University of Moscow, about the use of computers in schools. College students, particularly in mathematics and science, are exposed to them although apparently to a lesser degree than in the United States. High School students do not use computers. Our guide, who had a wonderful sense of humor, told us of an experimental computers system to translate one language to another. When given the English phrase "Out of sight, out of mind" it translated it into Russian as a "blind idiot." Think about it!

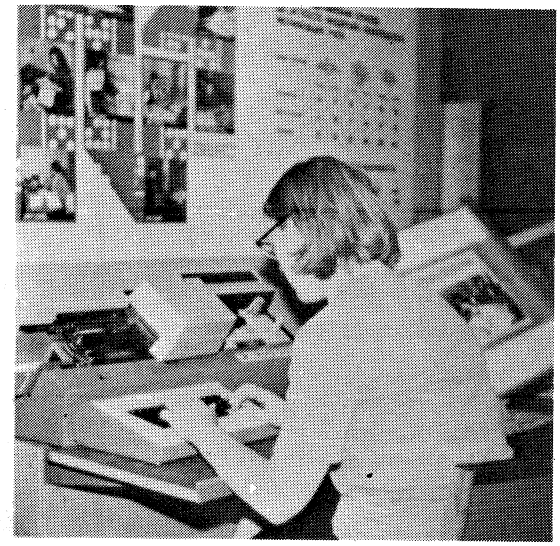


Instructor points to register of Russian M4030 timesharing system.

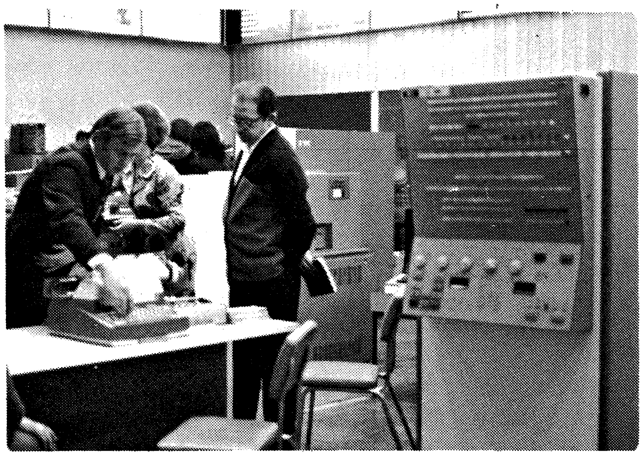

Russian version of the Teletype.



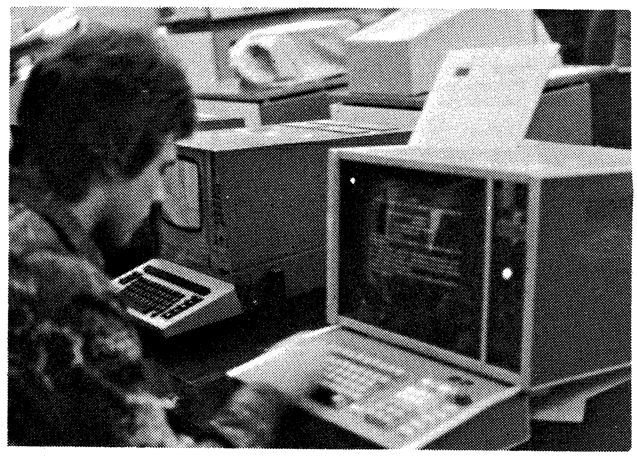
Typical pavilion at the USSR Economic Achievements Exhibition—large facade but only one or two rooms inside.



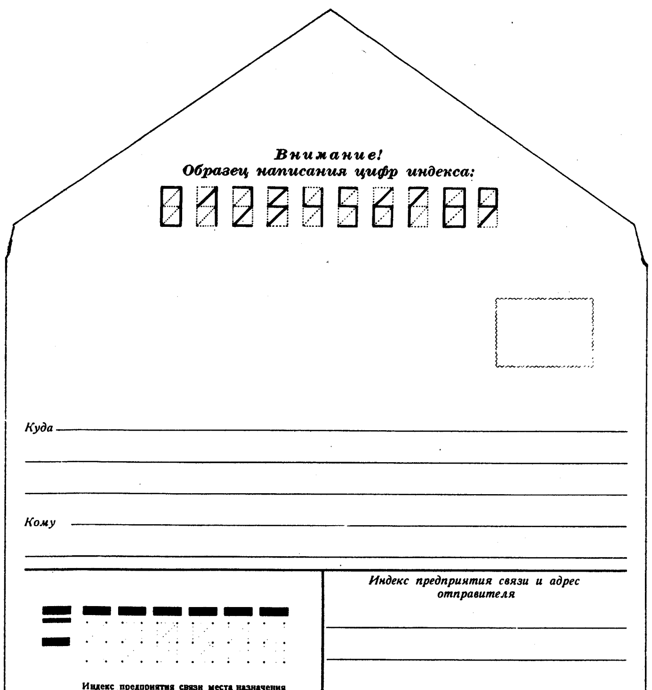
Keypunch machine.



Console of a Russian EC-1020 batch processing computer system.



Typical Russian CRT terminal. Most have very large screens.



Russians are required to put their zip code in machine-readable form on the bottom left side of envelopes.

A postcard view of Moscow University. During my visit the ground was snow covered and the sky cloud covered.

10 GOOD REASONS WHY COMPUTERS CAN . . .

A computer can do more work than
a man.
One reason that's little known
Is that it never has to stop
To answer the telephone.

A computer can do more work than
a man.
One more way to explain
Is that it doesn't stop it's work
To argue and complain.

A computer can do more work than
a man.
Because it never takes
Those dawdling, lengthy lapses
That we call coffee breaks.

A computer can do more work than
a man.
And it's easy to see why.
It doesn't sit with its chin on its hand
And watch the girls prance by.

A computer can do more work than
a man.
One reason it's such a whiz:
It doesn't buttonhole passersby
To tell them how busy it is.

A computer doesn't take nervous pills
All day at the water fountains,
And wastes no time with molehills
Making them into mountains.

A computer can do more work than
a man.
Because, I have a hunch
It doesn't spend three hours
With a customer at lunch.

A computer can do more work than
a man.
And one good reason I've seen is
It doesn't spend the afternoon
Half-conscious from martinis.

A computer can do more work than
a man.
And partly it's a matter
Of not spending all day angling
For the next job up the ladder.

A computer can do more work than
a man.
Here's a final explanation:
It wastes no time on fears of being
Replaced by Automation.

Author unknown

NEW ELECTRONIC DESK CALCULATOR FROM CHINA

KUNG CHANG

THE Great Wall 203, an advanced type of electronic desk calculator, was trial-produced early this year by a plant under the Institute of Mathematics of the Chinese Academy of Sciences. It is 2.5 times as fast as similar calculators produced abroad, has twice the storage capacity and an expanded machine language. It is also slightly smaller and easier to operate than such models.

The Great Wall 203 is of a type more advanced than ordinary electronic desk calculators. It has more functions, greater storage capacity, higher operating speed and under program control can automatically solve complex problems. Programs can be written, debugged and modified conveniently at the keyboard. Equipped with a printer and a magnetic tape unit, it is a complete, independent small computer system that performs some of the functions of a general-purpose electronic digital computer. Its easier handling and maintenance make it suitable for wide popular use.

Most of the people who designed and built the calculator are young mathematicians. Though they were unfamiliar with electronics and computing and their plant was poorly equipped, they drew encouragement from Chairman Mao's teaching, **"The Chinese people have high aspirations, they have ability, and they will certainly catch up with and surpass advanced world levels in the not too distant future."** After studying a lot of material and critically assimilating the good points of foreign and domestic calculators, they boldly created a design in line with the characteristic of Chinese components. Making full use of collective wisdom, the whole plant made suggestions in the course of trial production.

They designed and built this new advanced electronic desk calculator which uses integrated circuits in one year and five months. Now the Great Wall 203 is undergoing comprehensive testing to perfect it for production and distribution.

Testing the Great Wall 203. *Reprinted from a recent issue of* CHINA RECONSTRUCTS, *Peking.*

MICROPROCESSORS & MICROCOMPUTERS

THE STATE-OF-THE-ART

Brian L. J. Callahan • Managing Editor, DataPro Minicomputer Reports

The time of the microcomputer has arrived, forcefully and almost without advance warning. This latest evolution in technology has resulted from the efforts of the major semiconductor suppliers to gain a share of the EDP market. This article defines the microcomputer from several viewpoints – from its innate design, from its uses, and from its impact on the EDP industry.

WHAT IS A MICROCOMPUTER?

The distinguishing characteristic or component of a microcomputer is the microprocessor, one or more large-scale integration (LSI) chips that perform the basic functions of a processing unit. Contained within a typical 0.16-inch square package (thus the "micro" designation) are the usual elements of any processor – the arithmetic logic unit, I/O control logic and general-purpose registers. When memory and a complement of I/O devices accompany or work jointly with a microprocessor, a microcomputer is formed.

Present microcomputers incorporate devices fabricated by metal oxide semiconductor (MOS) techniques. MOS offers extremely high densities of transistors-per-unit-area, but is inherently slower than bipolar devices. Current MOS speeds for a logic element or chip range from 40 nanosec for fast, n-channel silicon gate devices, to 200 nanosec for p-channel metal units. Architectural attributes which exploit MOS technology have been added to increase the speed of microcomputers vis-a-vis bipolar units. They consist of hardware index registers, parallel bus structures, register stacks with programmable stack pointers, and decimal arithmetic.

HISTORY

Since the microcomputer is the apparent successor to the minicomputer as the latest and most advanced evolutionary step in EDP, its lineage will be briefly discussed.

At their commercial introduction in 1965, minicomputers constituted a revolution in data processing. Their compact size and low cost permitted the development of dedicated systems to meet specialized needs in communications, control, data acquisition, and small business data processing.

The potentials of minicomputers were at first not recognized or appreciated by system designers weaned on larger computers who viewed minis in terms of the features and programming languages offered by the larger machines. Program loading was awkward and time consuming, and the shorter word lengths and limited instruction sets made minicomputer programming tedious. Today, systems designers are more familiar with the vagaries and capabilities of minicomputers, and are implementing minis in a myriad of applications.

Microcomputers and microprocessors are following a similar course. In many existing minicomputer applications they offer improved price-performance, compactness and reliability over the mini. Moreover, the characteristics of the LSI microprocessor lend themselves to new applications and system concepts that are impractical with minicomputers.

EVOLUTION

As the minicomputer evolved upwards into the high end of small scale systems, electronic technology was advancing in circuit miniaturization and the use of MOS as a low-cost alternative to bipolar logic. This steady advance in MOS technology has increased the large scale integration of digital circuits from 100 MOS transistors per chip to over 14,000 per chip during the last five years. This increase in chip density has caused a revolution in digital hardware applications. Among the more publicized are the pocket calculator and the digital watch.

A microcomputer uses no more than 10 MOS/LSI packages, each holding more than 500 transistor circuits. A minicomputer would typically require about 100 TTL packages. This simple comparison reflects the prime difference between a minicomputer and a microcomputer – its physical size and the complexity of its components.

A concurrent development which has contributed to the evolution of microprocessors, and thus microcomputers, is microprogramming, where each machine instruction initiates a sequence of more elementary instructions (microinstructions). A microprogramming approach allows, replacing fixed, conventional, CPU control logic with a control memory. Addresses in control memory represent unique states in conventional control logic, and each memory output represents control lines from conventional logic. Stored in this memory are basic microinstructions, including the fundamental control, testing, branching and moving operations.

For an LSI machine to perform higher-level operations with ease, microinstruction sequences corresponding to common higher-level functions are stored in a separate read-only memory (ROM) to be accessed, decoded, and executed on command. These high-level sequences are called macroinstructions, the medium in which system programmers usually code. Macroinstructions in a microcomputer correspond to the basic instructions of a minicomputer.

Microprogramming enables a systems designer to adapt standard hardware to specific applications – perhaps the most useful characteristic of a microcomputer. The designer can construct macroinstructions that are best suited for the particular functions to be performed, and incorporate them into the microprocessor. For example, the instruction set of an existing minicomputer can be completely or partially emulated to minimize software development. Alternatively, a machine can be built to perform functions peculiar to an application such as word processing or data acquisition. This capability to adapt a standard set of hardware modules to a variety of problems combines the cost advantages of high-volume chip production with the computing efficiency of tailored instruction sets.

MICROCOMPUTER vs MINICOMPUTER

Although stark and simplistic price comparisons are sometimes misleading, it is not unfair to say that an LSI microprocessor has a substantial cost advantage over a typical minicomputer CPU. For example, a complete LSI CPU may be purchased for as little as $300, compared to $1000 to $2000 for a minicomputer CPU.

The CPU power consumption of an LSI microcomputer is 66 to 75 percent less than that of a comparable minicomputer. For a system containing but one CPU, the difference would not be significant considering the overall system's power requirements. However, in applications where many CPUs are required, the power difference would be substantial.

An MOS/LSI microcomputer operates at 50 to 33 percent of the speed of commercially available minicomputers. Typical memory-to-memory add times for a moderately priced mini are between 5 and 20 microsec compared to 15 to 60 microsec for a microcomputer. The speed of a microcomputer is derived from the particular MOS process used in fabrication. As these processes improve, so will the speed.

With integrated circuits, system reliability is largely a function of the number of printed circuit (PC) board interconnections. Since each LSI package replaces from 50 to 100 TTL packages, the interconnections required by microcomputers are reduced and total system reliability is increased. The LSI microcomputer can be built into a light and compact configuration because of the higher number of gates per package module and the simplicity of interconnection.

In summary, the LSI microcomputer offers better price-performance, lower power consumption and heat dissipation, higher reliability, and smaller physical size than a minicomputer. The microcomputer further offers the flexibility of microprogramming, which, in a given application, has many advantages. Although execution speeds comparable to today's minicomputer have not yet been achieved, several architectural techniques have emerged which will eventually increase microcomputer speeds.

CHARACTERISTICS

Microprocessor architecture is similar to that of a bus-oriented minicomputer. Applications can generally be categorized by bit width: four-bit microprocessors for calculators; eight-bit units for microcontrollers; and sixteen-bit units for microcomputers. The range of characteristics is broad:

Data Word Size
4 to 100 bits

Instruction Set
40 to 120

Instruction Format
8 to 24 bits

ROM
400 23-bit to 16K 8-bit

RAM
up to 65K 16-bit

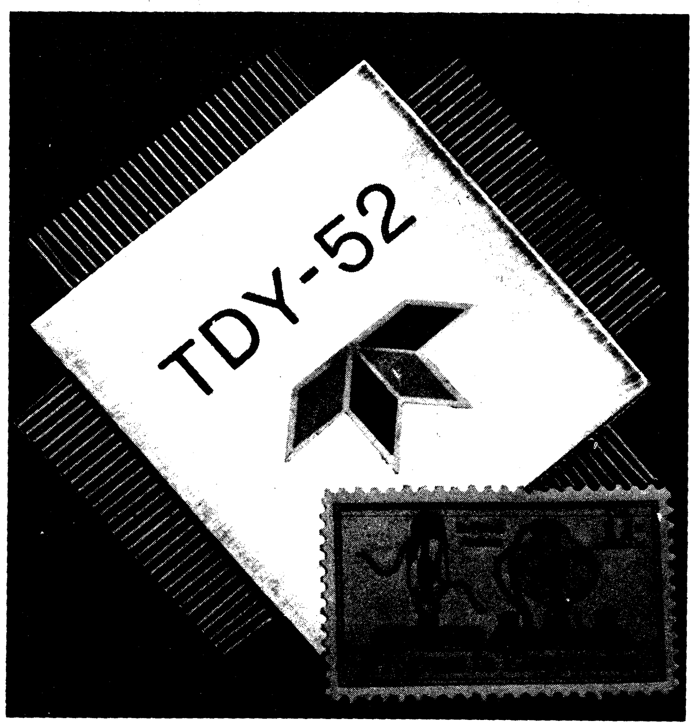
General-Purpose Registers
1 to 16

Cycle Time to Fetch & Execute An Instruction
0.54 to 62 microsec, with 5 to 10 microsec common

Stack Depth
2 to 32 levels

Interrupt Capability
None to full

Parallelism
mostly parallel to serial/parallel



An example of a microcomputer is Teledyne Systems' TDY-52, a programmable microcomputer contained within a 2" x 2" x 0.2" package. Teledyne offers two different configurations of the TDY-52: the TDY-52A, a package holding a CPU with 8 registers, a 4K x 8-bit microinstruction ROM control memory, 4K x 8-bit application program RAM memory, a 2K bit scratchpad RAM, input multiplexer, output buffer registers, priority interrupts, and oscillator; and the TDY-52B, a general-purpose 16-bit microcomputer with CPU and registers, priority interrupt, memory and I/O address register, clock generator, timing and control, and output buffers. Both configurations can also incorporate additional ROM, RAM and ROM/RAM modules, contained within another TDY-52 size package.

CONFIGURATIONS

Manufacturers provide three forms of microprocessors: MOS/LSI chip sets; a single PC card with processor and memory; and a card cage system containing a CPU card, memory cards, direct memory access channel cards, bit interface cards, and connectors for attaching a portable control panel.

Chip sets are suitable for large quantity requirements. The OEM buyer must meet loading restrictions and supply the required clock waveforms indicated in the specifications for the MOS chip.

A PC card approach provides a low-cost CPU that can be incorporated into existing hardware, eliminating most of the problems of interfacing. It is an excellent method to get a new product underway quickly, and can give way to the chip set at a later time if quantities are sufficient.

A card cage system is suitable primarily for breadboarding and prototyping. It comes complete wih power bussing and a breadboarding card on which the user may construct his own interface logic.

APPLICATIONS

The potential applications of microprocessors and microcomputers extend over a broad spectrum of products. Their principal use to date has been in electronic calculators – the extremely high volume quantities required by this market segment dictating the architecture of many microcomputers.

Terminals will be the next major market area to utilize microcomputers. Low-cost data terminals use microprocessors for simple data handling tasks. Remote terminals, by the addition of a microprocessor, become "intelligent," and perform off-line editing, compiling and processing. Point-of-sale (POS) terminals perform calculations, data storage and inventory control functions, and control keyboard, tag reader, display and printer peripherals under microcomputer control.

Microprocessors are useful for tasks normally associated with large-scale systems. In addition to performing channel control functions, they relieve the large central processor of the overhead associated with scheduling, text editing or file management. In a similar manner, microprocessors can be used for sequencing, control, formatting and error detection in tape or disc units. It is probable that more microprocessors will be buried in computer peripherals than will be used as computing devices.

LSI microprocessors, combined with low-cost memory and moderate performance peripherals like floppy discs, CRT displays or medium-speed printers, can provide all the processing power needed for many applications. A large multi-user computer system may soon be needed only for accessing large, on-line data bases or for a few CPU-bound program tasks.

In summary, microprocessors and microcomputers are or soon will be applied to the following types of equipment:

Calculators, both programmable and fixed-function, and small business/accounting computers

Terminals, both keyboard and special-purpose

Measurement systems, from panel meters to full-scale monitoring systems

Automotive systems and traffic controls

Medical equipment

Process and machine control

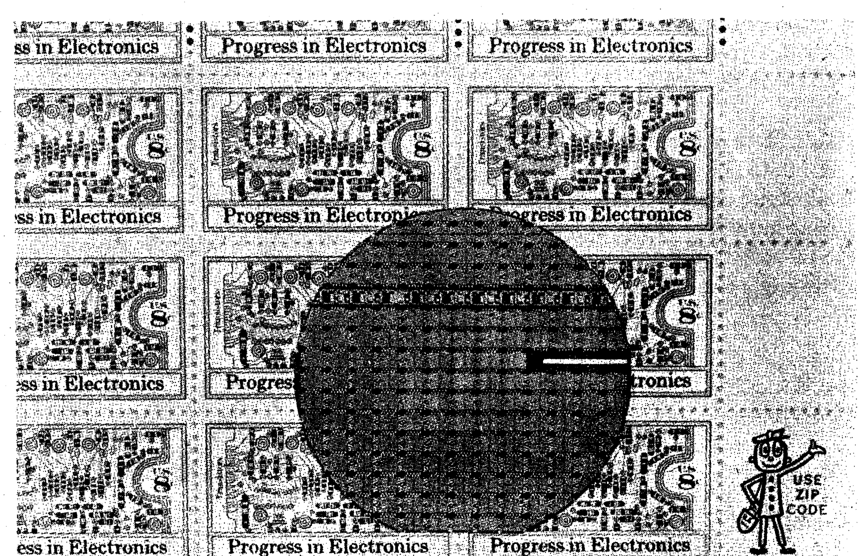
Computer peripherals and system control.

CONCLUSION

LSI microprocessors and microcomputers will soon replace conventional minicomputers and controllers in many applications where a mini or controller is overpriced or overpowered. Microprocessors now satisfy those minicomputer applications where high speeds are not required. Speeds of minicomputers have increased over the years mainly as an enhancement due to changing technology rather than in response to an overall need. While TTL logic was the least expensive technology, very little cost reduction could be realized by producing slower minis. The advent of the LSI microprocessor thus is forcing a reexamination of minicomputer price-performance tradeoffs. ■

HOW SMALL IS SMALL?

Microns and nanoseconds are units too small to conceptualize directly. But the accompanying photograph illustrates graphically, and dramatically, how far we've come in designing microminiaturized computer components. Shown superimposed on a background of postage stamps is a 2-inch wafer containing 130 integrated circuit chips. Each chip contains 4,200 transistors, for a total of 546,000 transistors on the wafer. The stamp is one of four "Progress in Electronics" commemoratives issued by the U.S. Postal Service in July. The chips are manufactured by Western Electric for use in telephone switching systems.



THE REACTIVE ENGINE PAPER

BY TERRY WINOGRAD

This paper, written in October of 1974, originally appeared as a file on the computer at the Artificial Intelligence Laboratory at Stanford University where Terry Winograd is currently leading courses in computer science, linguistics, and the social implications of computing. At the time of writing the people at SAIL were discussing how to design their new timesharing system. Another file was also maintained on the system in which comments, suggestions and objections could be stored. It is an argument for personal computing — not an essay, but a sort of cybernated commentary.

—Marc Le Brun

NOTE:
This is written in an attempt to provoke discussion, so it may overstate, exaggerate, etc. It is written from the viewpoint of the Ivory (silicon?) tower, so the issue is "What is possible?", not "What is now practical?" After all, working at the AI lab you would never realize that there are still people in the world (even at Stanford!!) who talk to computers by cutting little holes in pieces of cardboard.

WHY TIME-SHARING IS OBSOLETE:

Time-sharing is obsolete because it is based on the assumption that a person interacting with a computer large enough to do serious work cannot make good use of its computing power except during a fraction of the time.

GEDANKEN EXPERIMENT:

Imagine that by a decree of God (or ARPA or whoever) we were only allowed to run one user at a time on the PDP-10. All the systems efforts would be directed to making each person's time on the computer as profitable as possible. What kind of system would you organize?

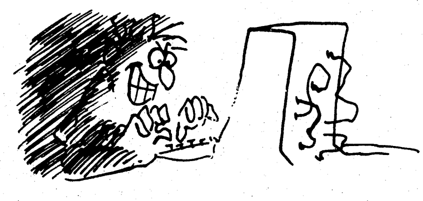
REACTIVE SYSTEMS:

I will borrow a term from Alan Kay (from whom I have also absorbed many of these ideas) to contrast our current "interactive" system with an imagined "reactive" system. A reactive system contains a full-scale processor whose only job is to cater to a single user. It can therefore afford to do relatively large amounts of processing when he or she is doing the simplest of tasks, like editing or giving monitor commands. It can maintain complex reactive graphic displays in real time (e.g. the multiple windows in the current SMALLTALK system).[1]

CLAIM:
The advantages of a reactive system over current interactive systems will be as large as the advantages of interactive over batch.

NATURAL COMMUNICATION:

One of the main advantages of a reactive system is that it can afford to do extensive processing to figure out what the user wants to do, based on both what he or she says and what the current context is. This is the main feature of natural language as a communication system — it is designed (evolved) with the assumption that the hearer will always make use of context and a shared base of knowledge in his interpretation. People are much more efficient at communicating in this style, regardless of whether it is in actual "natural language" or in some artificial language. There are lots of bits and pieces of this in current systems — everything from default file extensions to command completion to the rather extensive set of facilities (like spelling correction) in INTERLISP.[2] On current interactive systems there is a strong tendency to avoid these because they involve running a higher-level program to interpret inputs. With



a reactive system, the bottleneck is the user's typing speed (although we could even imagine some sort of simple voice inputs if we pushed this idea far enough), so we can afford to do lots of processing.

REMEMBERING AND DEDUCING:

Often the problem isn't that we want to specify a command in a way which needs a smart processor to understand what it is, but in a way which takes a smart processor to do it. If we were talking at the command level to a program with even limited deductive capabilities (the kinds now put into robotics programs) it would free us from much of the tedium of converting our desires into "machine code." Why shouldn't we be able to say in some language "put the current who display in a file" and let the system worry about the details of just what needs to be run.

The whole range of things like UNDO, REDO, etc., which are creeping into languages like INTERLISP could be extended to cover every aspect of our communication with the computer — it could remember the context of what we have been doing, and make the necessary deductions to do rather sophisticated things (e.g., redoing some sequence of actions, but tailoring it to apply to a different kind of data object).

EDITING: **Maxim: People Never Edit Character Strings.**

Whenever I am editing, I am editing a document, a message, a program, or some other structure about which I know much more than the simple sequence of characters. The reactive system should always be dealing with my editing in this way. Formatting systems for documents should be interactive and incremental — what I see on my screen should always be what I would see on the Xerox Graphics Printer when I put it out. This includes fonts, justification, diagram placement, etc. etc. Of course the program should be able to reconfigure and modify this in a whole variety of ways, but I should always be working with a document, not a source file (or even a screen-editor!) When I put in a new word or line, things should move to make room for it. Things like spelling-checking could be done incrementally, looking up each word as it is typed in, then when I ask for it, interactively pointing out those not in the dictionary (e.g. by flashing them) and letting me make changes. For programs, the editor should be part of an integrated programming and debugging system (as in INTERLISP), not a separate program at all.

PROGRAMMING AND DEBUGGING:

We have glimpses of integrated systems in SMALLTALK, INTERLISP, ECL[3] and Swinehart's thesis[4] describing a SAIL-based system. Often these are forced into horrible compromises by the fact that they are walking the line between wanting the user to have the full power a program can offer at each step of the way, while having to run them on time-sharing systems which are based on the assumption that you really only want to process part of the time. They fall far short of putting in the kind of integrated knowledge-base and deductive programs which would really allow the system to act as a programming assistant. They do not make full use of the possibilities for incremental compiling (coupled with the editor) which can give the user the feeling of always working with an interpreted system, while actually having the efficiency of careful compiling. They are only beginning to make use of the kind of graphic interactions which can greatly broaden the programmer program bandwidth. I have ranted about this elsewhere[5] so will say no more here, except to note that this is probably the most important area where reactive systems will change our concept of programming.

IS IT WORTH IT?

The same could be (has been and still is) asked about interactive computing. Ask the programmer who has become accustomed to an interactive debugging system and then has to go back to octal (or hexadecimal) dumps. In the short run (and we are always in the short run) this is a difficult question, and I will take the escape given above — it is certainly worth it if our research grants will support it. In the long run I don't think there is any doubt. The costs of processing, memory, etc. are going down and although there may be ultimate limits, we are still far from them. Anything which trades off increased processing for increased ease of computer use by people will be eventually justifiable on practical grounds. Twenty years from now this question will be as outdated as it would be to ask now whether it is really worth having the computer go to the work of accepting programs in symbolic form, since people are perfectly capable of converting them to binary numbers and entering them that way.

CAN IT BE DONE WITH CURRENT TECHNOLOGY?

Researchers at a number of places (including XEROX and MIT-AI) are currently designing and building "personal computers" which have a processing power far beyond standard minis or "smart terminals." In fact many of the ideas above are being implemented at Xerox. By connecting these into a simple network with shared facilities for things like specialized I-O (XGP, A-D and D-A, ARPANET, etc.) and large file storage (allowing file sharing), these computers could currently provide everything we get from a time-sharing system, while opening the way to all of the advantages of reactive systems. At the moment, the cost per user (even if they were commercially available — which they aren't) would be significantly greater than an improved this-generation time sharing system of the kind being planned at SAIL. However the picture is not that clear if we look ahead the 3-5 years necessary to actually get systems running. Careful thought needs to be given to realistically assessing what hardware will (or could if we put effort into it) be available by then, and how much it will really cost.

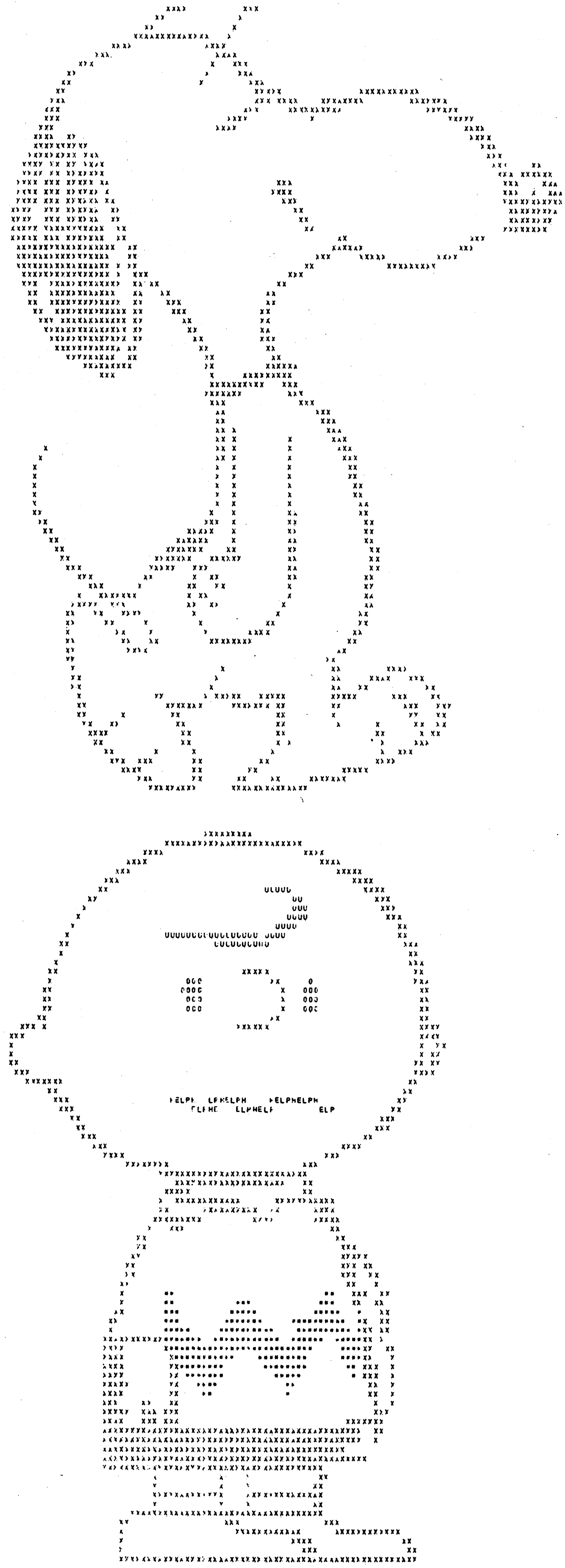
WHY SHOULD WE CARE?

This paper was circulated only partially because there is a general interest in talking about systems. There are particular reasons to think seriously about these issues now. AI researchers (particularly John McCarthy) have always been at the cutting edge of new systems ideas (like time-sharing, list processing, etc.). In this case I think we have even more to offer, since many of the techniques which are needed to build reactive systems are "AI techniques" involving knowledge bases, deduction, etc. We have the option of taking the viewpoint of "wiping the slate clean," thinking in terms of what we would really like to do with the new technologies, rather than incrementally building on the ideas and equipment now available. Even if the appropriate personal computers were not available right now, much of the design and debugging could be done on current systems if we had a clear idea of what we expected from the hardware.

PUNCH LINE: I believe that reactive systems are "where it will be" a few years from now, and we have the choice of leading or waiting to follow. ■

REFERENCES

1. **Personal Computing by Alan Kay, Learning Group, Xerox Palo Alto Research Center and A Personal Computer for Children of All Ages by Alan Kay, ACM National Conference, August 1975.**
2. **INTERLISP Reference Manual by Warren Teitelman, Xerox Palo Alto Research Center.**
3. **The Treatment of Data Types in ELI by Ben Wegbreit, Communications of the ACM 17, 5 May 1974.**
4. **COPILOT: A Multiple Process Approach to Interactive Programming Systems by Daniel Carl Swinehart, Stanford Artificial Intelligence Laboratory AIM 230.**
5. **Breaking the Complexity Barrier (again) by Terry Winograd, ACM Sigplan Notices 10:1 January 1975.**



About Computing

by Geoffrey Chase, OSB
Portsmouth Abbey School, RI

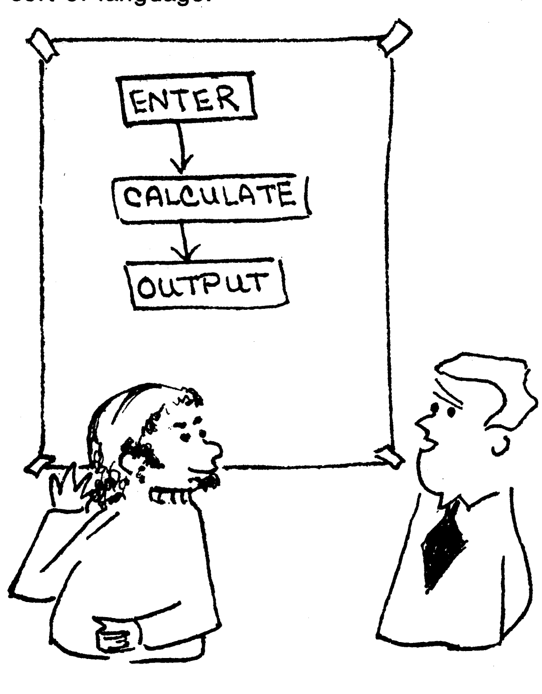
There's a famous "theorem" that runs something like this:

1. Any non-trivial program has at least one "bug".
2. (Corollary) Any program that works is trivial.

Computing—i.e., giving a set of instructions ("program") to a machine telling it what to do and what paths to take *if* this rather than that happens, and when to stop—is different from what most people seem to expect. To begin with, it doesn't necessarily have anything much to do with numbers, arithmetic, or algebra, though students of these subjects often use computers and usually (by no means always) find programming easier than do people who haven't much math background.

Another odd feature is that it is not quite so much a matter of "right" or "wrong" answers as one might think. Almost no one gets his program right the first time he runs it, unless it's copied off a book or does nothing you couldn't do better without the machine. And of those who do get the "right answer"—i.e., a program that finally works—some will write programs that are unnecessarily slow, use up too much of the machine's memory, and stretch an accumulation of small errors into large ones [computers don't make mistakes but they often generate error—figure that one out!]. Someone else may write a program that is lean, taut, elegant, fast and accurate. Both have come up with the "right answer", but the second program is obviously "righter" than the first. Often it is also simpler, or simpler-*looking*, than the inferior program.

The hard part of computing is keeping your head clear. The easy part is what looks hard, programming in some sort of language.



I'd expected a little more documentation.

Some more "theorems", stated without proof:

3. Any fool can write a program no one else can understand. It takes intelligence to write one that is clear and seems easy.
4. Our #3. fool will find, six weeks later, that even *he* can't figure it out.

Write programs so someone else can read them. Use a few comments; make the order of the program adhere as near as can be to the order in which one *thinks* about the problem being solved.

5. Computers are *dynamic*—they "move" things. Many familiar expressions in math and elsewhere are *static,* reflecting an unchanging truth but not giving much hint of how to find or figure. So your job in most instances is to find some sort of repeated *process* (they call computers "processors" sometimes) which will get you what you want in a finite number of "moves" or steps.

Such a process is called an "algorithm". You often don't know, by the way, *how many* moves or steps will be required; merely that there will be only a finite number of them.

It's not at all rare to write an "infinite program" by mistake. This can hang up the processor and the programs of other users, who will be tempted perhaps to hanging of another sort . . .

Example: $Y = 5 \cdot x^3 + 4 \cdot x^2 + 3 \cdot x + 2$ and $X =$ (let's say) 1.2; how do we find the value of "Y"?

"Static" method: multiply X times X times X times 5 and then . . . all the way down the line. A lot of work.

"Dynamic method": let Y = 5. Now, for "K" equal to 2, then 1, then 0, do the following:

a. Multiply Y times X (which is 1.2) and add the coefficient of X-to-the-K-power. When K reaches 0, this is the so-called "constant term", +2 in our problem. Let Y = this new value and *throw away* the old value of Y. In computer talk,

Y = Y*X + C(K), where "*" means "times" and C(3), C(2), C(1), C(0) are the coefficients (from left to right) of our equation. Note that C(K) means "C subscript K"—not "times K"—and that the = sign really means "is replaced by" rather than "is identical to".

b. Your last Y is the answer. Believe it or not, this is by far the easier way to do our problem.

This repeated application of a simple idea is the key to programming.

6. A woman's work is never done, and neither is the programmer's.

You will nearly always find yourself rewriting programs. It might work, *but* you would like another feature added; or, it blows up if the user does something stupid so let's check for stupidity on his part; or, it can be combined with two or three other short programs to make the Pan-galactic Interplanetary Super Solver that cures all ailments.

It's like fine furniture: sanding, resanding, finishing, refinishing, until you really like the looks of it.

7. When in doubt, guess. You don't know what will happen if . . . ? Well, try it. Smart guess-work

David vs. 12 Goliaths

by Monty Newborn

It looked like a sure thing; David would quickly polish off his opponents and we would be able to have an early dinner—most likely before midnight! The slugfests between programs to determine a national champion usually begin at 7 p.m. at the ACM Annual Conferences and continue until almost dawn—certainly well past midnight—and for those participating, this means a late gourmet meal at a nearby twenty-four hour diner over which the evening's activities are rehashed. But tonight, rather than battling each other, the twelve programs were scheduled to take on David Levy, British International Chess Master, in the first simultaneous chess exhibition in history in which a Master could not count on his opponents cringing in respect. Dinner was not too far off.

The scene was a second floor conference room in the Radisson Hotel in downtown Minneapolis; the date was October 19, 1975. The audience of several hundred included both chess experts and computer professionals. On stage were the authors along with teletypes and telephones connecting to remote computers. My role was that of organizer, along with Ben Mittman of Northwestern University, and participant. Data General had been good enough to provide me with a Nova 2 for the event and it served as the only "live" and visible entrant, its lights twinkling as moves for my program OSTRICH were calculated. David has been serving as director of the ACM Tournaments since 1971 and is likely to come into several thousand dollars in 1978 when the computer community fails to produce a program that can defeat him in a match. He accepted a wager in 1968 to this effect from a distinguished group of computer scientists. This simultaneous exhibition would certainly set straight the supporters of the metal monsters!

And so the evening began. It became clear that David intended to take the computers "out of book" as soon as possible and, in general, play somewhat closed positional games. The computers played slowly, taking about 3 minutes per move, while David bounced from one board to another, only seeming to be concerned over his game versus CHESS 4.4 running on a CDC CYBER 175. OSTRICH was holding ground but definitely having the worse of it. One by one David's opponents met defeat, marked visibly by the disconnected telephone and the posting of the results on large display boards in the room. But the games were lasting longer than I expected. David must be playing very safe was my guess; he doesn't want to lose any games or draw any either. The pressure against OSTRICH continued to grow with David building up small gains. But it was now nearing 10 p.m. and there were still about six programs alive! I became completely immersed in my game at this point, losing track of what was happening in the other games. David seemed to be only a few moves from crushing OSTRICH and I was glad to see OSTRICH had made a run of it at least. But then, much to my surprise, David made a weak 25th move giving OSTRICH some chance for equalizing the game and even a chance to gain the lead. OSTRICH saw David's error and made the correct reply, and David was in trouble—but not enough. David, playing at a slight material disadvantage, gradually recaptured the lead and defeated OSTRICH on move 50. Thus, at about 11 p.m. OSTRICH joined the ranks of about 8 other programs that had gone down to defeat. I could now relax and watch David finish off his other opponents.

But it didn't happen that way. David found himself behind in two games and fought to survive until well past midnight when his opponents agreed to draws. CHESS 4.4, the program of David Slate and Larry Atkin of Northwestern University, and TREEFROG, the work of Ron Hansen, Russell Crook, and Gary Calnek of the University of Waterloo, both were ahead but unable to develop a strategic plan leading to a resignation by their mortal opponent.

So once again at 1 a.m. we went to the local diner to rehash the day's events and speculate on how numbered David's days are. On a head-to-head match, ten years may still be safe, but David—beware!

"About Computing" continued —

is much more important in (even) pure mathematics, to say nothing of more practical things, than people realize. Every theorem was once merely a hunch. It is important to find its proof, true; but it is almost more important to find something to prove!

8. Aesthetics (elegance, style) count for a lot, and are a lot more practical than "gimmicks" in the long run.

9. Computer books and write-ups (this one included) are hard to read at first, and full of words that nobody bothers to explain.

Suggestion: If you have trouble on this score, plow ahead and keep on reading, regardless of whether you know what the author is saying. A later page may very well explain an earlier one. Then go back to what bothered you; if it still bothers you, make a note of it and lay the book down.

Another day it may come clear; or somebody may be able to answer your questions.

The writer first met computers when he was 40. The result: near-despair until the above technique was used; then things began to come clear. It's like learning a foreign language; speak and hear as much of it as you can, even if half is obscure.

10. The ingenious chaps who invent computer languages (like BASIC, which will probably be what you start with) put in error "traps" to catch certain obvious blunders. But in general, alas,

a. The machine can and will do what you ask it to do, no matter how stupid or wise your orders may be. It doesn't know the difference.

b. The machine can't possibly tell you *what* you should be asking it to do. Of fact and fiction, value and meaning, it knows nothing. "GIGO" (garbage in, garbage out) is an old IBM motto!

Good luck on your adventure!

Summary of the ACM Sixth U.S. Computer Chess Championship

by M. M. Newborn
McGill University, Montreal, Canada

Playing stronger chess than ever before on CDC's superfast CYBER 175, CHESS 4.4, the chess program of David Slate and Larry Atkin captured the ACM's Sixth U.S. Computer Chess Championship in Minneapolis at the ACM's Annual Conference on October 19-21, 1975. CHESS 4.4 defeated all four opponents in capturing its fifth of the ACM's six tournaments. TREEFROG, the work of Ron Hansen, Gary Calnek, and Russell Crook of the University of Waterloo and winner in 1975, lost only to CHESS 4.4 in the final round and finished in second place. It was TREEFROG (under the name RIBBIT) that dethroned Slate Atkin's program last year. Twelve teams participated in the four round Swiss style tournament.

In addition to the tournament, David Levy, International Master from England and the tournament director, played a simultaneous exhibition on Sunday evening against the programs. SORTIE passed up the exhibition; Slate and Atkin's program played two boards running on Northwestern University's CDC 6400 computer in one game and on a CDC CYBER 175 in the other. Levy won ten games, drew two, and lost none; CHESS 4.4, running on the CYBER 175 and TREEFROG, running on a Honeywell 6080 drew with Levy. Against CHESS 4.4, the game ended with Levy having a Pawn on the seventh rank but down a Knight and probably lost against best play by his opponent (Fig. 1). In his game against TREEFROG, Levy left a rook en prise on the 50th move and was fortunate to gain a draw when TREEFROG was unable to force a win with a Rook and Bishop to Levy's Bishop of opposite color. (Fig. 2). In the exhibition the computers played at a rate of 40 moves in two hours.

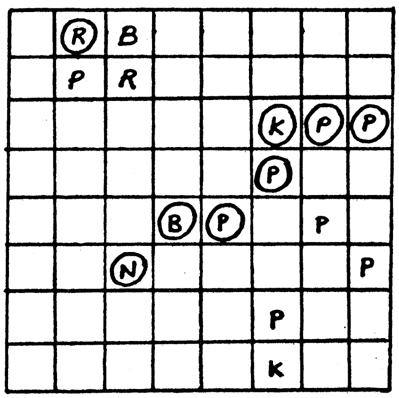


Fig. 1. Position at end of Levy (White) vs CHESS 4.0 (Black) White to move.

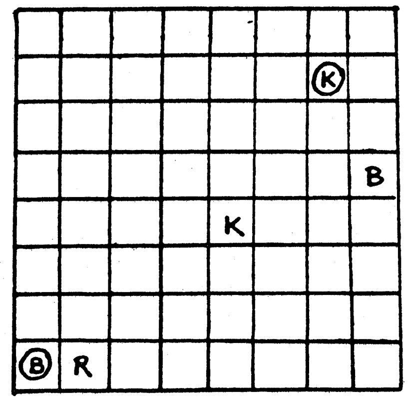
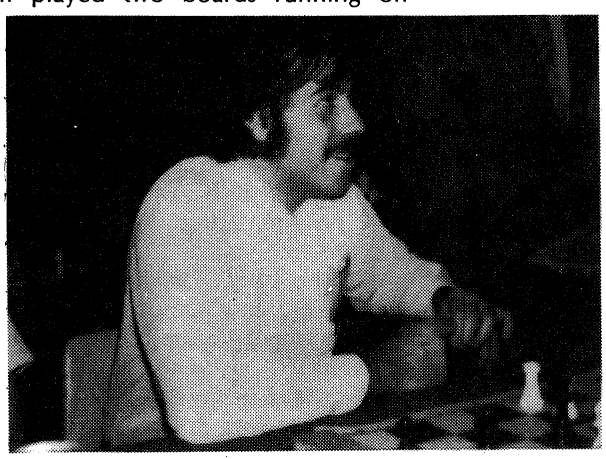


Fig. 2. Position at end of TREEFROG (White) vs Levy (Black) Black to move.

*Black pieces are circled



David Slate, programmer of CHESS 4.0, winner at ACM-76.

"The minute you say that a thing cannot be done, you are through with that thing. And no matter how much you know — even if you are an expert — if you say it can't be done, you are all through. And someone knowing nothing about it, but thinking it can be done, now is a better man for the job than you."

Harry Myers

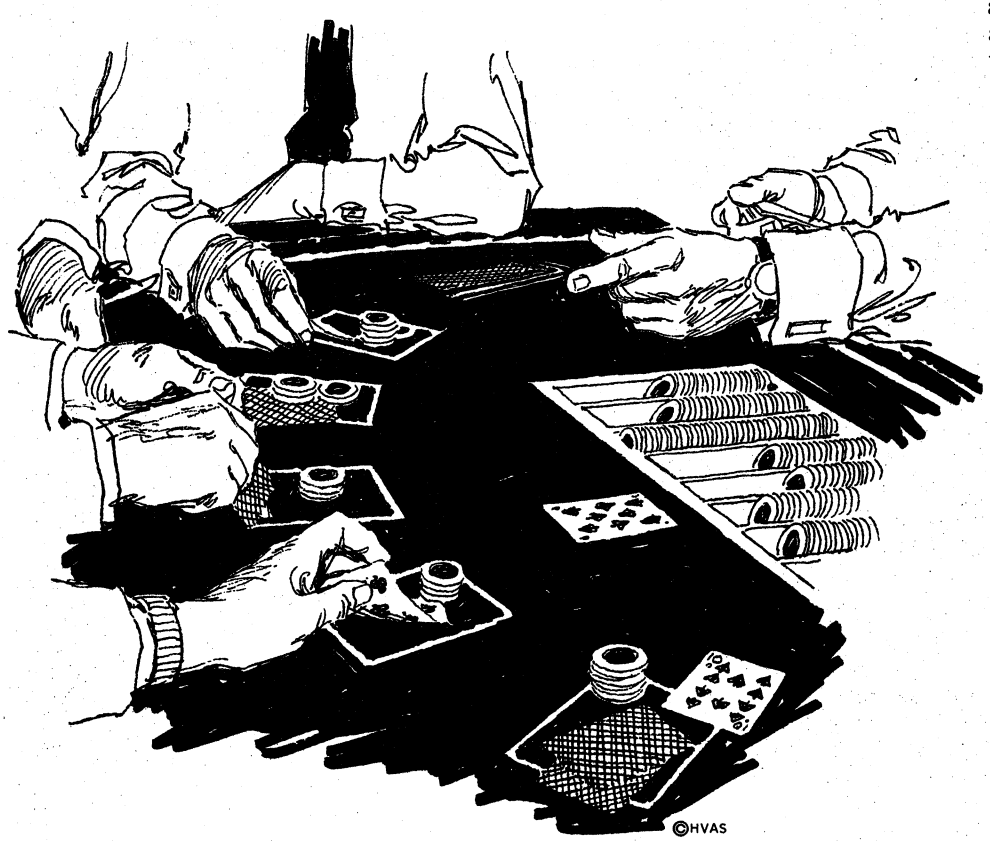
Beating the Game

Game theory compares blackjack systems and proposes to teach a computer backgammon

by Dietrick E. Thomsen

The man who broke the bank at Monto Carlo is a musical fantasy that grew out of the avid interest many people have in the things that happen on green baize tables. The man who, according to folklore, was told not to return to Las Vegas because he had won too much money there is real. He is Edward Thorp, a professor of mathematics at the University of California at Irvine.

Lately Thorp has been looking for the best way, in a theoretical and practical sense, to beat the blackjack table. He has devised a way of comparing the several blackjack systems against each other and a theoretically best possible system. At the same time his interest has turned to that ancient, but recently trendy, game, backgammon. He shared some of his latest insights on these topics with fellow mathematicians at the recent National Mathematics Meeting at Washington.



The blackjack systems depend on counting the cards as they fall. As play proceeds, the deck is depleted, and with the fall of each card the player's expectation of success changes. Removal of different denominations from the deck changes the expectations in different amounts. Removal of two nines does not have the same effect as removal of three fives.

From the way the expectations change, a particular numerical value can be assigned to each denomination of card. As the cards fall from the deck a running total of these numbers is kept. Different systems assign different numbers to different denominations. They also differ in how they use the running value total and the number of cards remaining in the deck. There are various combinations of addition and division. There is also a difference in whether the system keeps a separate count of aces. Aces have two values in blackjack, 1 and 11, and therefore some system makers like to tally them separately. The result of all this arithmetic is used to advise the player how to bet.

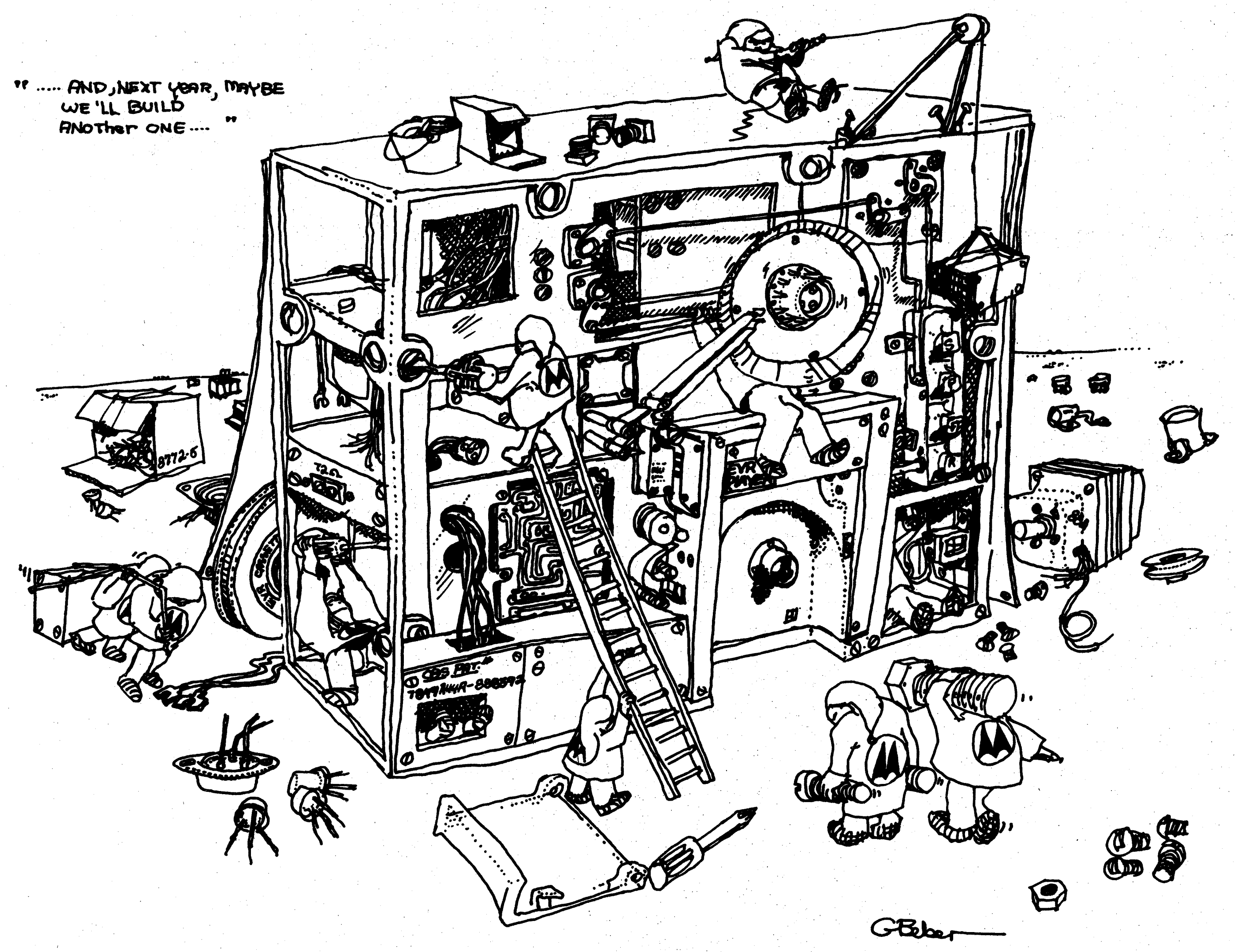
The question Thorp set himself was whether there is some method of comparing the different strategies without doing a massive computer simulation of a million hands. He finds one and he finds a criterion to compare them with each other and see how close they come to a theoretically possible optimum system.

First he needs a definition of "more advantageous." It may seem obvious that it means a greater chance of winning, but the case is complicated because a given system may give a greater expectation of winning when the play is in a particular stage, but it may be surpassed by another in a different situation. The final working definition of advantage is a system that gives at least as good a chance of winning over as wide a range of situations as an alternative with at least no more risk to the player.

Thorp finds that he can compare the quality of systems by defining an expectation function for each one that expresses its relative betterness. The expectation changes as play proceeds. It depends on the fraction of cards remaining in the deck, and it varies as they fall.

Graphically the expectations define a surface called a simplex, and the falling of cards causes motion from point to point on this surface as the expectation changes. Working with the geometry of the simplex Thorp can compare system to system, and he finds that he can define a single number, which he designates with the symbol lambda, that expresses a system's betterness relative to others and its closeness to a theoretically possible optimum system. Thus he has an analytic method for ranking blackjack systems and no longer has to simulate a million hands on a computer to compare them. But he does not tell us which is the best possible system.

Backgammon is among the most ancient games. A set dating to 2600 B.C. has been found. From the game theore-



tician's point of view, Thorp says, the hope of solving the problems that it presents is quite good.

The game consists of a ladder of 26 cells. Of these, 24 appear on an actual backgammon board. For the analysis, Thorp adds, the two "off the board" spaces at each end into which counters that have successfully completed their journeys are put. Each player's counters start at one end of the board, and the object is to get them all across the board and off it, passing the other player's counters coming in the opposite direction, before he gets his across. Moves are determined by throwing dice.

There is an important complication. If counters of both players arrive in the same cell, there are situations where one can be sent back to the beginning of its trip. This possibility of repeated restarts makes the game in a theoretical sense potentially infinite. In principle a backgammon match could last forever. It is "a fact which will impede analysis slightly," Thorp concedes.

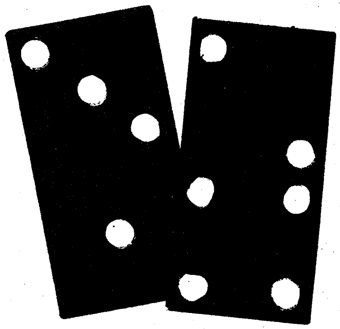
The way to analyze the game is to set up partial models that are simplified, removing complexities of the real game, especially the one that makes it infinite, and then gradually to add back the complications. In Model I each player has one counter, and the bounce-back rule is suspended so that the counters can freely pass each other. When this is properly set up it produces a game of 167 steps. The first to reach step 167 wins. Computer simulation shows that when goals are equal, the first player to roll has a slight advantage, but this declines toward even chance as play proceeds. This is a very crude approach to a real backgammon game, but it leads to interesting insights, Thorp says.

In Model II one sets up an end game. Again there are two counters, but they have already passed each other so there is no further chance that they could be sent back to their starting points. This too is a finite game and is amenable to solution.

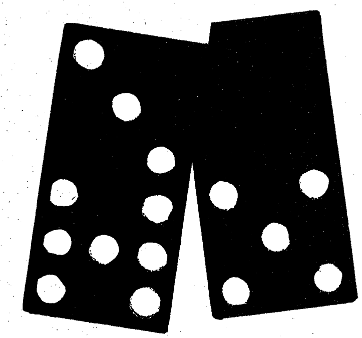
One complication of the real game is, the doubling cube. As the game proceeds, if one player gains a certain advantage, he can use the doubling cube to double the stakes. This changes the consequences for the loser and alters the expectations and strategy of play. Recursion schemes can be devised to solve both Model I with the doubling cube and the end game with the doubling cube. (A recursion scheme is a system for calculating a series of related values. Knowing the first number in the series and the recursion scheme you can calculate the second. Putting the second number into the recursion scheme gets you the third. And so on.)

In actual backgammon play it is possible that the game might come down eventually to Model I or Model II, but these highly restricted situations are still far from the complexity of a full game, in which each player has several counters on the board at once and the bounce-back rule can operate. Still enough has been learned so far for Thorp to conclude that backgammon "can be played better by computer than by any person." But suppose the computer refuses to go to Reykjavik? □

In computer play, the aggressive method of play is significantly better than the defensive.



SIMULATED STRATEGIES OF GAME PLAYING



by Dr. S. Reisman
IBM Canada, Ltd. Laboratory

Game theory enables one to classify competitive games according to characteristics of rules of play of the game. Consequently, the game of chess is described as a *two-person, zero sum* game of *perfect* information. This classification describes a competitive situation between two players in which the total game situation is open to observation to both players. Both players have opposite interests in the outcome and one player's win is the other's loss. In addition, the game can be terminated in a finite number of moves. Games of perfect information differ from games of imperfect information by the fact that, in the latter, there is no best strategy. However, in the former, sometimes called strictly determined games, the player with the best strategy will win regardless of his opponent's play.

The game of Draw and Match Dominoes is classified as a *two-person, zero sum* of *imperfect* information and according to game theory there is no best strategy of play which can be employed.

It was hypothesized while this might in fact be true, techniques of simulating cognitive processes might be used to determine if better strategies do exist. Consequently, player protocols were gathered from opponents while they played a simplified version of Draw and Match Dominoes. An analysis of the protocols indicated that human players use a combination of a number of components of play to form their playing strategy. These components are described as: (1) the defensive component; (2) the aggressive component; and, (3) the statistical component.

The defensive component is characterized by a player's being more apt to make assumptions about his opponent's game situation and as a result attempting to block his every move. The aggressive component is different in that the player using it makes no assumptions about his opponent's situation and instead makes his move only on the basis of his own known situation with the objective of playing his longest chain of tiles. The statistical component is used rarely, and only as a last means of decision making if the other two components are unsatisfactory. It is characterized by the player's counting the various tiles already played and making decisions as to the likelihood of matches being drawn from the bank.

In an effort to determine the validity of this classification system an interactive Dominoes-playing program was written in the list processing language IPL-V.* The program was altered so that the strategy employed in a game could be varied to combinations of the above components. Results of the computer play indicated that the classification system is, in fact, an operational one.

In order to determine if there are optimal strategies of play, the interactive program was altered to allow the play of the game to be between two computer programs rather than between a human and a computer, as in the original version. In this way, one program using one type of strategy could be played against a program of another strategy. The programs were each loaded onto a different interactive terminal and the output of one terminal's program was used as input to the terminal with the second player-program.

In this attempt to determine a better strategy, one program was set to use only the defensive component, and the other, only the aggressive component. The results of a number of games played in this manner indicate that the aggressive component of play is significantly better than the defensive component. An analysis of the games played indicated the reasons for this. The strictly defensive player makes assumptions concerning his opponent's situation while ignoring his own game situation and the harm he may be doing to himself by blindly attacking his opponent.

The typical human player does not rely on only one component of play, but both makes assumptions about his opponent and considers his own situation. For this reason, a better strategy would probably consist of a combination of these components in a particular game situation. Although the search for a better strategy of play in the game of Dominoes is not of earth-shaking importance, the potential spinoff of the techniques used and results obtained may be of value in other situations of a competitive nature.

---

*For a complete description see: Dominoes—A Computer Simulation of Cognitive Processes, *Simulation and Games*, Vol. 3, No. 2, pp. 155-164.

BEYOND BASIC

by Alan B. Salisbury

INTRODUCTION:

There are many levels of exposure to and interest in computers. It's probably safe to assume that the majority of Creative Computing's readers are familiar with BASIC and the computing power which it makes available to them. An equally safe assumption is that many readers do not have a very deep knowledge of the computer beyond that which they obtain through BASIC. This level of computer interest and understanding is sufficient for many purposes and represents a very large community of computer users.

The vast majority of users of computers have little or no knowledge at all of the computer. In fact, they may not even be aware at times that they are using a computer! These "users" include, for instance, the people who drop a quarter into a slot to play a tennis or hockey-type game using a video screen and hand controls. Slightly more involved, perhaps, is the person who sits at a terminal to play STAR TREK after someone else has loaded in the correct program and set the system up ready for play. Countless other applications could be listed in which the fact that a computer is doing the behind-the-scenes work may or may not be apparent to the user.

This article looks in the other direction, beyond BASIC, toward a deeper level of understanding of the computer. The purpose of twofold: first, it will help the reader to put BASIC in its proper perspective in relation to the many other types of computer software; second, it may (hopefully) stimulate many readers to expand their knowledge of computers and thus open up whole new areas of excitement and challenge.

BASIC IN PERSPECTIVE . . .

To begin with, let's review exactly what BASIC really is. Simply stated, BASIC is a programming "language." Like any language, it has a vocabulary with precise meanings and a set of rules as to how that vocabulary may be used (semantics and syntax). BASIC can be used to express problems and their solutions in a form that is readily understandable by humans—it looks reasonably close to English and algebra for instance.

Generally, BASIC is *not* directly understood by computers! (There are a *very few* special computers which actually are built to directly understand BASIC, however.) Hence, a translation is required from a user program written in BASIC into another form which is directly understandable by the computer. BASIC is therefore usually referred to as a "higher order language" (HOL) when compared to a "machine language."

There are other higher order languages, of course, in addition to BASIC. The most popular of these are FORTRAN (FORmula TRANslation), ALGOL (ALGorithmic Oriented Language) and COBOL (COmmon Business Oriented Language). A host of additional languages exist, highly tailored to specific uses. All of these languages are similar in that they require translation into a machine language before the computer can actually run the program.

COMPILERS . . .

Higher order languages wouldn't be of great value if the programmer was saddled with the job of doing the translation from BASIC, for instance, to machine language. Fortunately, this is the type of job that a computer can do very well. Programs called "compilers" have been written to do this translation for us. A compiler takes as its "input" a higher order language program (called the "source program") and produces as its "output" a machine language program (called the "object" program) ready to be run on the computer. We thus have a two-step process including a compile (translate) phase and an "execute" (run) phase.

One of the important and very nice features of higher order languages is that they are generally "machine independent." That is, when writing in BASIC, the programmer doesn't need to know if his program will be run on an Interdata 7/16 or a DEC PDP-11/40 for example.[1] Different translators will, of course, be required, one for the Interdata machine and one for the DEC machine, since their machine languages are quite different. These translation programs are usually provided by the manufacturer of the hardware.

MACHINE LANGUAGE . . .

We have mentioned machine language many times without really addressing how it differs from higher order languages. This can best be understood by looking at the familiar four-function calculator, which in many ways is a very simple computer. This calculator can do four functions: add, subtract, multiply and divide. This is, in effect, the vocabulary of "instructions" which the calculator understands. We can use the four-function calculator to solve any problem whose solution can be reduced to a series of steps (a "program") using the +, -, X, ÷ operations.

As an example, let's consider the BASIC statement:

```
LET X = A+5
```

We can translate this into a series of steps (using the calculator) such as the following:[2]

```
Clear
Enter A
Depress +
Enter 5
Depress +
Read X
```

Here we have expanded the vocabulary to include the human actions (clear, enter, depress for input, and read for output). In the language of the machine/human team, the above program is a translation of the BASIC statement.

Moving from the four-function calculator to a minicomputer is not difficult. One major difference is that the

[1] In fact, some differences in BASIC may occur between manufacturers.

[2] The exact sequence will depend on the logic of the particular calculator. The sequence shown is typical.

vocabulary, or "instruction set," is greatly increased; a typical mini may have 70 or more functional instructions which it understands, including the familiar add, subtract, multiply and divide. New capabilities here include such things as "shifting" numbers left or right, or "comparing" two numbers to determine if one is bigger, smaller, or the same as the other.

Another significant difference is in the amount of memory available. Our calculator example, above, did not include memory. Many calculators have one or more "memory" locations in which a number may be "stored" and later "recalled" by use of appropriate keys (which actually add to the instruction set vocabulary of the calculator.) Computers generally have *thousands* of such memory locations which can be used to store not only numbers (or "data"), but they can also be used to store instructions so that the entire program can be stored within the computer.

It is this last fact (program storage) which makes the computer the powerful instrument it is. With the program stored inside the computer, the operator need only depress the "start" or "run" switch and the program can then be executed at electronic speeds, typically in *millions* of instructions per second. Add to this the fact that the computer can make "decisions" (for example, using a "compare" instruction, go on to *different* parts of the program depending on the results of the comparison) and you have a good explanation of the power of the computer.

Returning to our previous example, a sequence of minicomputer type instructions to perform the LET X = A+5 function might be

```
CLA A
ADD Five
STO X
```

CLA, ADD and STO are abbreviations (called "mnemonics") for the full instruction names. For example, CLA could be "Clear and Add," meaning "clear the working (accumulator) register to zero and add to it the contents of the memory location indicated" (in this case, A). ADD would simply be "add to the accumulator register" without clearing beforehand, and STO would be "store" the contents of the working register in the indicated memory location. Each of the many instructions of the computer has a similar detailed and precise meaning.

BINARY NUMBERS

In the binary number system, only two symbols (digits) may be used, 0 (zero) or 1 (one). The decimal system, on the other hand has ten symbols: 0 thru 9. Just as in the decimal system, when counting causes us to run out of symbols in one column, we carry 1 into the next column and start over, the same process occurs in the binary system. The difference is that a carry in the decimal system occurs for every count of ten, while it occurs in the binary system for every count of two. Therefore, "place values" in the binary system are powers of 2 (1, 2, 4, 8, 16, etc.) as compared to place values in the decimal system which are powers of 10 (1, 10, 100, 1000, etc.).

Conversion from binary to decimal is a simple process requiring only that the place values corresponding to 1's in the binary number be added up. As an example:

64's	32's	16's	8's	4's	2's	1's		
								32
								8
								4
0	1	0	1	1	0	1	=	1
								45

To complete the picture, we have to now point out that even the above three instruction program sequence is not really in *machine* language. Computers don't normally understand letters and words such as CLA. In fact, everything within the machine must ultimately be in the form of binary numbers, consisting only of combinations of ones and zeros. This is the fundamental unit of information within the machine, known as the "bit" (for binary digit).

Memory locations and working registers within a computer generally contain a fixed number of bits which is the "word" length of the computer. Minicomputers typically use 12 or 16 bits per word; alternatively, as few as 8 bits (the "byte") or as many as 64 bits may be handled at a time. The significance of word length lies primarily in the magnitude of the numbers that can be stored in a single word; fewer bits means smaller numbers or, more important, greater round off error since, in effect, fewer significant digits can be saved. Speed is also affected, since a wider word length usually means that more bits can be processed at one time.

It was mentioned earlier that memory is used to store both data and the program. Program instructions are stored in memory words according to precise formats. As an example, a 16-bit word may be divided into 6 bits for an "operation code" (specifying the particular instruction to be performed) and 10 bits for a memory address (the location of the data to be used). Our CLA A instruction would thus include a 6-bit code for CLA (it might be 001010) and a 10-bit binary address for the location we chose to call A (this could be 0000101101 if location 45 were used for A).[3] The complete machine language instruction would then be:

001010	0000101101
Opcode	Address

ASSEMBLERS . . .

Once again we are faced with the problem of translating from the mnemonic (or symbolic) form of instruction into the numeric machine language form. The computer comes to the rescue as before with a special translation program (available from the hardware manufacturer) which "assembles" machine language programs from the symbolic instructions. An "assembler" program takes as input symbolic assembly language source statements and produces as its output a binary machine language object program ready for execution. Unlike higher order language programs which are largely machine independent, assembly language programs are very machine dependent since each type computer has its own, generally unique, machine and assembly language.

In addition to relative machine dependence or independence, there are other considerations involved in understanding the differences between compilers and assemblers. Our example showed that a single BASIC statement resulted in *several* assembly language-type instructions. Each assembly language instruction, on the other hand, usually causes a *single* binary machine language instruction to be generated. It is not uncommon for a good assembly language programmer to write a program requiring fewer machine instructions than one written in a higher order language and compiled into machine instructions. The assembly language programmer may therefore find that his program requires less memory and executes faster. The price paid for this possible bonus is usually the extra time (and training) required to do assembly language programming. As compilers become more efficient, this difference may narrow.

[3] See inset box for an explanation of binary numbers.

LOADERS ...

We have talked about compilers, assemblers, source programs, and object programs. How do these programs get into the machine? One (horrible) alternative is for an operator to load the binary instructions through the console of the machine, one word at a time by setting switches or entering them through a small calculator-like keyboard. Preferably, the program could be punched onto paper tape or cards beforehand and then simply read into the machine. The capability to "read" a program into memory requires that another program already must be in memory to cause the reading to take place! Such a program, called a "loader," must include all of the many detailed instructions required to read a card, for instance, and then move all of the 80 characters or other binary information read from the card into designated locations in memory. In fact, all input and output operations are fairly complicated at the machine language level and the manufacturer usually provides "utility" programs so that the programmer doesn't have to write these himself. For the same reason, a "library" of other useful routines is generally available including such things as math functions.[4]

SYSTEMS SOFTWARE ...

Compilers, assemblers, loaders and utilities are collectively referred to as "systems software" and are usually written by the manufacturer's "systems programmers." The using programmer is, in contrast, referred to as an "applications programmer" and his "applications programs" are the programs written to solve the user's problem.

Starting with a program written in BASIC, there are many steps required before the output is available. First, a loader must be in the machine. (Loading the loader is itself a job to be done by the operator!) Then the BASIC compiler can be loaded in, the source program read, and the translated object program produced as output. Now the object program is read in (again by a loader) and then executed producing the final results. Input/output utilities and math routines may well have been required and loaded in along with the object deck.

The smallest mini (or micro) computers are often operated in just this fashion, with the operator handling many separate programs and manually controlling the sequences of loading and execution. This job, too, is one in which the computer can lend a hand.

OPERATING SYSTEMS ...

The most important of "systems" programs is one called an "operating system" (sometimes "executive," "monitor," "control program") which takes over these tasks of scheduling, allocating space in memory, calling other systems programs, etc. With an operating system (OS), the using programmer can simply state the kind of job he wants to do (using a special "job control language" requiring a card or statement of the beginning of his program), and the OS will handle many of these details for him. Here, then, we have another form of computer language, that which is used to talk with the operating system.

With an OS in control of the computer, our sequence of events is simplified. The source program deck, written in BASIC, together with a job control card, may be all that the user must place in the input card reader. After reading the job control card, the OS will call in the BASIC compiler (probably stored on a disc file), load it, translate the source program to a machine language program, load the machine language program together with all required utility and library programs, and finally execute the object program. What we have described here can be called a "Compile-and-Go" scheme, in which the system does not have to stop between the translation and execution phases.

INTERPRETERS ...

Our brief tour of systems software has omitted one important type of program, the "interpreter," which is particularly important to BASIC users. The use of a BASIC compiler to first translate a BASIC program into machine language for later execution is one method available for the BASIC user to execute his programs. It may be the only method, depending on the computer and its available systems software. Frequently, however, a BASIC interpreter may be available to execute BASIC programs, either in addition to a compiler or in place of it.

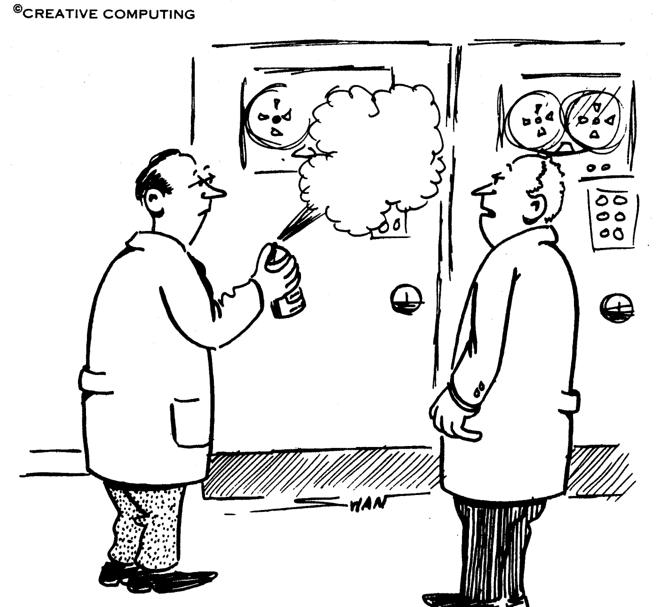
An interpreter differs from a compiler in one major area: an interpreter does not *translate* the source program, but rather it effectively *executes* it directly. Each line of a BASIC program is examined by the interpreter to determine what actions will be required to execute (or, more accurately, evaluate) the statement. The interpreter then performs those actions by immediately executing the appropriate portions of its *own* program, and then moves on to the next source BASIC statement. This is in contrast to the compiler process in which the entire BASIC program is translated to a unique set of machine language instructions which are then executed after the compiler has finished its job.

To illustrate how the two systems differ, let's consider how they would process the BASIC statement:

LET X = 2+3

A compiler would produce as its output a machine language *program* consisting of several instructions, which *when executed later* would calculate the desired result and assign it to the variable X. An interpreter, on the other hand, would produce as its output the new *value* of X which is 5; no instructions or new program would be produced by the interpreter.

What differences does the user see between a compiler and an interpreter? Not many. Both methods ultimately produce the same end results. If the program is to be executed many times, there is a benefit to using the compiler method; in this method, it is possible to obtain a copy of the object program produced by the compiler, on punched cards or tape for instance. Then when the user



"Where did you learn to debug a program, Haverstraw?"

[4] Hardware math instructions are normally limited to add, subtract, multiply, and divide. Trig functions, square roots, etc., require programs that compute these higher math functions using the basic instructions.

wants to execute his program again, he can merely load it and execute it, without going through the time consuming translation phase. Compilation therefore offers the potential of time-saving efficiency.

INTERACTIVE SYSTEMS . . .

Interpreters generally offer an advantage of their own in that they may be "interactive." This allows the programmer to sit at a teletype, for instance, with the interactive BASIC interpreter in control, and get immediate "feedback" from the interpreter as he enters his BASIC statements. Errors can be immediately identified by the interpreter and corrected by the programmer. Also, the results of executing each BASIC statement can be available immediately, thus making the system react to the user as if it were a sophisticated calculator. For a new program, it may therefore be possible to get an answer faster using an interpreter than using a compiler, since the interpreter does not have to go through the separate translate and execute phases.

DEBUG PROGRAMS . . .

A special type of interactive program often available for mini's and micro's is the debug program, sometimes called a monitor.[5] A debug program is used to aid in debugging a machine language program. It typically permits the programmer to insert "break points" in his program which will cause the program to pause in its execution and then allow the programmer to examine the contents of key registers and memory locations. In this way, the programmer can walk through the execution of his program and if there is a "bug" in the program, he can isolate it (hopefully).

TEXT EDITORS . . .

Other programs besides interpreters may be interactive. One of the most common is a "text editor" which can be used to help create a new program or "file." If a BASIC compiler were available through a time-sharing system, an interactive text editor could be used to build or create a BASIC program using an on-line terminal; the text editor would permit the programmer to make changes such as adding or deleting individual characters or whole lines. When the programmer is satisfied with his program, he could then ask the system to compile it and execute it. The software performing the actual time-sharing operations serving many users simultaneously is essentially a more complex type of operating system as we have previously discussed.

SUMMARY . . .

We have now completed a very general introductory overview of systems software. The reader should now have a reasonable understanding of the various types of programs involved, the functions they perform, and how they relate to one another. The types of systems programs available and the functions they perform are summarized in the accompanying table.

[5]This use of the term "monitor" is somewhat different from the OS type "monitor."

TABLE OF KEY SYSTEMS PROGRAMS

Systems Program				
Name	Function	Input(s)	Output(s)	Comments
Compiler	Translate HOL Program to Machine Language Program	HOL Source Program	Listing. Machine Language Object Program	Object Program may be punched out or loaded into memory
Assembler	Translates Assembly Language Program to Machine Language Program	Assembly Language Source Program	Listing. Machine Language Object Program	Object Program may be punched out or loaded into memory
Interpreter	"Executes" HOL Program	HOL Source Program	Problem Solution	Output is only what the source program produces. Interactive
Loader	Loads Machine Language Programs	Machine Language Object Program(s)	Machine Language programs ready to be executed	Loaded into memory
Text Editor	Creates or edits files	Source Programs, data, etc.	Listing or copies of files when requested	Interactive
Debug Program	Facilitates isolation of program bugs	Commands to Debug Program	Responses to commands, e.g. register contents, memory contents, etc.	Runs along with user's object program to be debugged
Operating System	Overall control of computer system and its resources	Job Control Language or equivalent. All other inputs to computer	Responses to JCL commands. Log of system status, error messages to operator, etc.	Schedules jobs, allocates memory, etc., to minimize need for operator

The Computer "Glass Box"

Teaching With *A Programming Language*

Howard A. Peelle
University of Massachusetts

Introduction

The COMPUTER GLASS BOX is a bold new approach to teaching with A Programming Language.[1] In this approach, short and quickly comprehensible computer programs are given to students for their direct viewing. Each program embodies a concept, a procedure, or a relationship and is written as simply and clearly as possible. The inner workings of such a program are visible and, hence, become the basis for learning.

This approach utilizes a computer program more as a "glass box" than a black box. The program's formal definition – expressed in the explicit terms of a programming language – serves to elucidate and reveal understanding. By observing the structure of a program as well as its behavior, key concepts may become *transparent* to the student.

Related Research

The glass box approach represents a synthesis of ideas put forth by three other researchers. MIT's Seymour Papert has recommended that children study procedures actively by using a computer programming language (called LOGO) as a conceptual framework [1]. Kenneth Iverson of IBM has persistently stressed simplicity and generality in using APL to expose fundamentals in a variety of mathematical and scientific disciplines [2]. IBM's Paul Berry first advocated open use of APL as a strategy for teaching in what he called the "functional approach" [3].

Characteristics of the COMPUTER GLASS BOX Approach

In contrast to conventional computer-assisted instruction (CAI), the glass box approach allows the student significant *control* over his own learning processes. This control is achieved through the activity of programming. Programs can be entered independently by the student via a computer terminal, and their use requires no other pre-stored curriculum material – as do most CAI applications. Indeed, making the full power of the computer accessible to the learner is 180° from the kind of CAI characterized by programmed instruction, tutorial, or drill-and-test sequences.

This approach is pedagogically suitable for a wide range of educational levels – from elementary school children to university graduate students. Especially for children who have been held powerless in lock-step educational systems, use of the computer in this way opens up new worlds of learning – *active* learning, learning with *power*.

Using glass box computer programs, students can proceed to learn during several complementary activities. Specifically, they can:

- *examine* the program's definition (intuitively)
- *analyze* the program's definition (logically)
- *predict* the outcomes of the program
- *execute* the program on a computer
- *scrutinize* the program's behavior
- *experiment* with different applications of the program
- *modify* or expand the program
- *generalize* the program
- *invent* new or related programs, and
- *discuss* implications with teachers and peers.

These student-initiated, student-responsible, success-oriented activities differ dramatically from frantic hand-waving about abstract concepts often seen in classrooms.

The ideal glass box program is also expository – it 'speaks' to its reader, explicating concepts and procedures in concrete terms. Desirable characteristics of such a program are:

Simplicity
Comprehensibility
Flexibility
Generality
Elegance
Provocative Implications

By "simplicity" I mean that a single idea of modest scope is to be taught using a brief program (about 10 lines of APL coding, taking less than 5 minutes to type). By "comprehensibility", I mean using clear, readable commands (usually one per line) with well-chosen mnemonic identifiers. By "flexibility" I mean a program design which is easily modified and which can be used with other programs in modular structuring (nested sub-programs with explicit resultants). By "generality" I mean developing mathematical models which can extend to a class of cases. By "elegance" I mean choosing expressions which strike one's aesthetic chords. And, finally, a glass box program is "provocative" when its implications suggest interesting follow-up discussions.

To the extent that these characteristics foster insight and learning, a glass box program is, itself, a *pedagogical agent.*

Examples of Glass Box Programs

To illustrate this approach, some sample glass box APL programs are described below, with accompanying suggestions for extending their use in teaching-learning settings.

COMPUTER – ASSISTED INSTRUCTION

In order to emphasize the contrast with conventional uses of computers for teaching, the first glass box program illustrated is from the area of computer-assisted instruction. Instead of concealing the CAI program – usually designed to control the child's behavior – we show him the mechanism itself so that he may see how it works and ultimately *control the computer.*

Consider the APL program below which exposes the essence of drill-and-practice in multiplication skills. In drill-and-practice, typically, a student is given a series of problems to solve, is asked for his answers, and the answers are judged for correctness, etc. Indeed, the computer is an excellent vehicle for administering drill-and-practice, but a programming language can also *describe* this process clearly.

The DRILL program begins with a NEWPROBLEM and prints 'MULTIPLY', a simplified message telling the student what to do with the two numbers that will follow. The FIRST number is an integer randomly chosen between 1 and 20, and the SECOND number likewise.

[1] *A Programming Language* (abbreviated APL) is a new multi-purpose computer programming language developed by Kenneth Iverson of IBM. Originally conceived as a unifying mathematical notation, APL has since been used successfully in fields such as business, scientific research and education.

```
    ∇ DRILL
[1]   NEWPROBLEM:
[2]   'MULTIPLY'
[3]   ⎕ ← FIRST ← ?20
[4]   ⎕ ← SECOND ← ?20
[5]   ENTER: ANSWER ← ⎕
[6]   →NEWPROBLEM IF ANSWER=FIRST×SECOND
[7]   'NOPE. TRY AGAIN.'
[8]   →ENTER
    ∇
```

The student may ENTER his ANSWER which is then judged for correctness by the program. IF the ANSWER equals the FIRST number times the SECOND number, a NEWPROBLEM is given; otherwise (if ANSWER is wrong) 'NOPE. TRY AGAIN.' is printed, and the student may ENTER his answer again.

NOTE: IF is a sub-program used to facilitate the reading of branching commands. Its definition is:

```
    ∇ BRANCH ← LINE  IF  CONDITION
[1]   BRANCH ← CONDITION/LINE
    ∇
```

Its syntax is → (line number) IF (condition)

It means that IF the condition is true (evaluates to 1), the program branches to the line number (or line label) given; IF the condition is false (evaluates to 0), the program branches to the next line.

In order to use the DRILL program, its name is typed. The following is a sample:

```
      DRILL
MULTIPLY
 19
 2
⎕:
      38
MULTIPLY
 16
 18
⎕:
      248
NOPE. TRY AGAIN.
⎕:
      288
```

```
MULTIPLY
 8
 12
⎕:
      96
MULTIPLY
 6
 2
⎕:
      12
MULTIPLY
 14
 18
⎕:
```

Students notice immediately that this program has a flaw. It does not stop! Scrutinizing the program's definition reveals that after getting a multiplication problem correct, one always gets a new problem – ad infinitum. Also, after getting a problem wrong, the student must answer that same problem again – another potentially endless loop. The student's first task, then, might be to build in an option to stop the program at will.

DRILL is, of course, only a *prototype* program. With other modifications of one's choosing, DRILL may become considerably more sophisticated. Possible extensions include: (a) displaying pictorial feedback – like a "smiley face" for positive reinforcement or a "grouchy face" instead of 'NOPE. TRY AGAIN.', (b) presenting a pre-specified total number of problems, (c) limiting the number of allowable mistakes on individual problems (or all problems), (d) generalizing the multiplicands to create a more flexible range of problems (including negative numbers, decimals, etc.), (e) gathering performance data, (f) using performance criteria to make diagnoses, (g) automatically adapting level of difficulty based on diagnoses, (h) adding personalized instructions, and (i) building in timing components, jump-ahead options and hints.

```
 *****        *****
* o o *      * x x *
*  ∇  *      *  Δ  *
* \_/ *      * /‾\ *
 *****        *****
```

PSYCHOLOGY

With computer programs suitable for viewing, students may learn some fundamentals of psychology. In studying behavior, for example, consider the following APL program[1] which models – albeit crudely – an emotional reaction.

```
    ∇ TEMPER
[1]   EMOTION ← 0
[2]   NEW: EMOTION ← ⎕ + EMOTION÷2
[3]   →MAD IF EMOTION>10
[4]   →NEW
[5]   MAD: '**!?!**?!!'
    ∇
```

TEMPER is a program which will, under certain conditions, get mad at you.

The program begins with zero EMOTION and then encounters a series of numbers, representing 'events' in the life of the program. A low number is low in emotional significance; whereas high numbers are highly emotion-producing.

Each time a number is entered, the program generates a NEW EMOTION based on a simple mathematical model: EMOTION becomes the number just entered plus one half of the previous EMOTION. (In the course of human events, this might be akin to the ameliorating effect of time on emotional burdens, i.e. 'sleeping on your troubles'.)

This process continues until a test condition – the "threshold" for mad behavior – is exceeded. The program goes MAD if EMOTION ever becomes greater than 10. (**!?!**?!! is the computer's programmed vernacular.)

To use the program, the child types its name (TEMPER) and then enters a sequence of numbers. For example:

```
      TEMPER
⎕:
      4
⎕:
      6
⎕:
      8
**!?!**?!!
```

This sequence produced MAD behavior.

But, suppose one tries entering the same numbers in a different order:

```
      TEMPER
⎕:
      8
⎕:
      6
⎕:
      4
⎕:
      →
```

Here, the program does not display MAD: '**!?!**?!!'. Apparently, (for this model) the sequence 8 6 4 is "tolerable," whereas the previous sequence 4 6 8 clearly was not tolerable!

[1] This program is similar to one written in a simplified FORTRAN by John Loehlin in *Computer Models of Personality,* Random House, NY, 1968.

Again, this suggests an analogy with human behavior: experiencing the most emotion-packed events first and then tapering off may be more tolerable than the reverse.

Other variations of input also suggest interpretation in terms of human psychology. Sandwiching a low-emotion event between two high-emotion events, say 7 2 7, can make the total sequence tolerable; by contrast, the events 7 7 2 and 2 7 7 produce mad behavior.

The mathematics underlying this TEMPER model can be exposed quickly and naturally. For example, after some experimentation with the program, one might wonder: How many 5s can the program take before it 'blows its top?'

```
      TEMPER
⎕:                         ⎕:
      5                          5
⎕:                         ⎕:
      5                          5
⎕:                         ⎕:
      5                          .
⎕:                               .
      5                          .
```

A sequence of 5s builds up EMOTION to higher and higher values, but never reaches 10. This process parallels the well-known geometric series 1 1/2 1/4 1/8 1/16 1/32 . . . the sum of which converges to 2. Exploring in this way, a child may gain some insight into the nature of infinite series in an active and interesting (at least less abstract) setting.

Some simple modifications of the TEMPER program students might make are to: (a) change the threshold, e.g. from 10 to 25 for higher tolerance, or to ?25 (a random number) for unpredictable behavior; (b) modify the model, e.g. from EMOTION ÷ 2 to EMOTION ÷ 3 to express stronger 'forgetting'; (c) adapt the program for use by others, e.g. inserting conversational statements such as 'ENTER NUMBERS FROM 1 TO 9' or even 'CAUTION! THIS PROGRAM MAY BECOME EMOTIONAL. . .', and (d) make the program dynamic, e.g. automatically resetting EMOTION to 0 after an emotional catharthis.

Possible extensions of TEMPER include: (a) writing related programs, such as a version with multiple emotional dimensions like ANGER, FEAR, and LOVE, and (b) writing companion programs, such as two TEMPER-like programs which interact with each other so that one's output is the other's input.

CYBERNETICS

In the area of cybernetics, students can be introduced to some sophisticated ideas by using simple computer programs. Scene analysis, for example, is an important part of robotics research. In designing vision machines, it is important to know what types of scenes can be computationally distinguished. Consider the two scenes below:

```
      SCENES

*********         *********
*                 *       *
*                 *       *
*                 *       *
*********         *********
        *
        *
        *
*********         *********
```

One scene is "connected"; the other is not connected. Note that the same line segments comprise the two scenes, but that they are in different positions.

Suppose one of these two SCENES is PICKed at random. Call it MYSTERY.

```
      MYSTERY ← PICK  SCENES
```

Further suppose that you are permitted to PEEK at small portions of the MYSTERY scene -- called "microscenes" -- but you are not told *where* the microscenes came from. For example,

```
      PEEK  MYSTERY

****
*
*

      PEEK  MYSTERY



****

      PEEK  MYSTERY

*
***
```

PEEKing at MYSTERY is like using a flashlight to illuminate small unidentifiable places on a much larger unknown scene.

After a period of probing, the question arises: *Can* you determine which scene it is that you are looking at?[1] (The answer is postponed so that the reader may ponder this question.)

The APL programs which facilitate exploration of this question in scene analysis are simple indeed:

This program will PICK one of 2 SCENES at random for the result called MYSTERY.

```
      ∇ MYSTERY ← PICK  SCENES
[1]     MYSTERY ← SCENES[?2;;]
      ∇
```

This program will PEEK at some two-dimensional SCENE and produce a random 4 by 4 portion for the result called MICROSCENE.

```
      ∇ MICROSCENE ← PEEK  SCENE
[1]     MICROSCENE ← SCENE[(⍳4)+(?10);(⍳4)+(?10)]
      ∇
```

The enterprising student might elect to automate the production of MICROSCENEs.

```
      AUTOPEEK                      ∇ AUTOPEEK
                               [1]    ''
***                            [2]    PEEK  MYSTERY
  *                            [3]    ''
                               [4]    →1
 ***                                ∇
 *
 *
 *

 *
 *
 ****

 **
  *
  *
  *


   **

   .
   .
   .
```

Soon it should become clear that these two SCENES *cannot* be distinguished on the basis of random microscenes

[1] This question is treated as a theorem by Minsky and Papert in their book *Perceptrons,* MIT Press, 1970.

Perceptrons are theoretical machines which can be trained to detect features of a scene by computations in a layered network of logical elements.

alone. (Of course, if one could trace sequentially through a scene, its "connectedness" or "non-connectedness" could be determined easily.)

Possible extensions of this excursion into scene analysis include studying perceptrons and related questions about "spatially local evidence."

COMPUTER ART

The world of computer art can be opened to students through a few simple APL programs. Beginning with a foray into automated design, they can proceed to engage matters of aesthetic judgement and artistic technique.

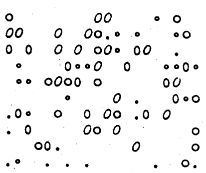
For example, consider the following DESIGN program.

```
     ∇ PICTURE ← SIZE  DESIGN  COLORS
[1]    HOWMANY ← ρCOLORS
[2]    PICTURE ← COLORS[?SIZEρHOWMANY]
     ∇
```

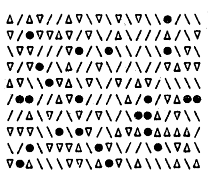
DESIGN uses some COLORS (symbols on the keyboard) and some SIZE (two dimensions of a matrix) to produce a PICTURE.

A simple program like DESIGN goes a long way with children. They seem never to tire of it, for it can produce quite a variety of designs:

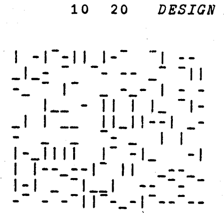
```
      10  20  DESIGN  ' O  o  ○  ∘  . '
```



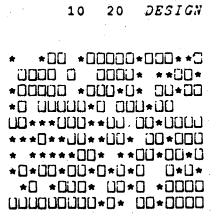
```
      10  20  DESIGN  '//Δ●∇\\'
```



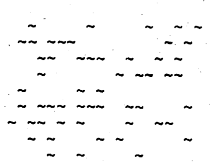
```
      10  20  DESIGN  ' ¯-||-_ '
```



```
      10  20  DESIGN  '⎕* *⎕'
```



```
      10  20  DESIGN  ' ~ '
```



While these "computer hieroglyphics" may have dubious aesthetic appeal, one can imagine -- instead of these typed symbols -- randomly generated swatches of color, perhaps displayed on a televsion-like screen.

Extensions of this approach to computer art include: (a) automating DESIGN, (b) weighting the selection of COLORS, (c) asking for human judgement (Do you like it or not?) in order to adjust weights on COLORS or other aesthetic factors, and (c) piecing together several computer-generated PICTUREs into a montage.

Another approach to computer art involves viewing programs which simulate artistic technique. For example, consider the program MONDRIAN below (named after the Dutch abstract painter).

```
     ∇ MONDRIAN
[1]    CANVAS ← 30 50 ρ ' '
[2]    DAB: COLOR ← 'o⎕*'[?3]
[3]    SIZE ← 3 5 ⌈ ?6 10
[4]    PICK: PLACE ← ?30 50 - SIZE
[5]    OVERLAP ← +/+/CANVAS[PLACE[1]+⍳SIZE[1];PLACE[2]+⍳SIZE[2]]≠' '
[6]    →PICK IF OVERLAP>2
[7]    CANVAS[PLACE[1]+⍳SIZE[1];PLACE[2]+⍳SIZE[2]] ← COLOR
[8]    →DAB IF (PERCENT ' ' ON CANVAS) > 67
[9]    CANVAS
     ∇
```

MONDRIAN begins with a blank canvas (arbitrarily set at 30 by 50). Then the program chooses a random COLOR, SIZE and PLACE to DAB.

OVERLAP measures the extent of overlap with DABs already on the CANVAS.

IF OVERLAP is greater than 2, then it will PICK another PLACE. (This is tantamount to finding relatively open space on the CANVAS).

IF, however, OVERLAP is not too large, the COLOR is put on the CANVAS at the PLACE and in the SIZE selected.

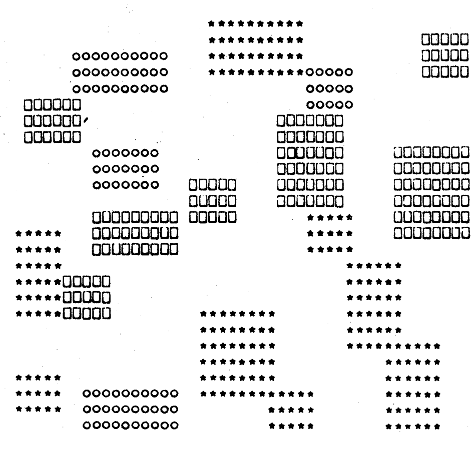
The program continues to DAB IF the PERCENT of blank spaces ON the CANVAS is greater than 67. In other words, as soon as it is 1/3 filled up, CANVAS is displayed.

Note: MONDRIAN uses two simple sub-programs (mostly for readability). They are PERCENT and ON:

```
     ∇ HUNDREDTHS ← PERCENT  N
[1]    HUNDREDTHS ← ⌊0.5 + 100×N
     ∇

     ∇ DENSITY ← SYMBOL  ON  PICTURE
[1]    DENSITY ← (+/+/SYMBOL=PICTURE) ÷ (×/ρPICTURE)
     ∇
```

Now, MONDRIAN at work:

```
      MONDRIAN
```



Possible extensions of this kind of program include: (a) simulating and combining additional artistic techniques (those that can be operationalized), (b) computing abstract measures of difference between random "paintings" produced by the computer, (c) converging to minimal differences from a previously specified "ideal" painting, and (d) developing a model for aesthetic judgment -- perhaps one which "evolves."

Conclusion

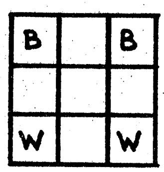
These are but a few APL "glass box" programs designed to stimulate students to think about selected concepts. Each of the sample programs shown here can be used *as is* and, of course, can be extended in a myriad of directions. Other topics well-suited for this pedagogical approach include some drawn from linguistics, statistics, mathe-

Creative Chess

by Walter Koetke
Lexington High School, Mass.

The game of chess was introduced to Europe near the middle of the thirteenth century. As one might expect of a game with over 197,000 ways of making the first four moves with over 71,000 resulting positions, many people have always been fascinated with the game but so very few have mastered it. This article is not, however, about the game of chess but rather about a few of the recreations it has spawned.

According to W.W. Ball (1), one of the oldest known chess related recreations was proposed in the early 1500's. The problem consisted of a 3 by 3 board with two white knights (W) and two black knights (B) positioned as shown.



The object is to move the pieces so that the squares occupied by the white knights are occupied by the black knights and vice versa.

When one is learning to play chess, some type of point value for each piece is usually assumed. For example, I was taught to value a queen as 10, a castle as 5, and a bishop and knight each as 3. You may have noticed that different introductory texts are likely to suggest different point values. How can this be? How are these point values assigned? If the relative point values are valid, then why do the experts disagree?

Suppose we choose to evaluate a piece by computing its "checking power". We can define the checking power of a piece as the probability that the piece will have the king in check if the piece and the opposing king are placed at random on an empty board.

To compute the checking power of a castle, the castle is placed on an empty board leaving 63 empty squares. Of these 63 squares, 14 are controlled by the castle. A king placed at random on the board would, therefore, have a probability of 14/63=2/9 of being in check. Thus the checking power of a castle is 2/9.

Computing the checking power of a bishop is a bit more tedious. A bishop placed on any of the 28 squares in the outer ring of the board commands 7 of the remaining 63 squares. However, if the bishop is placed on any of the 20 squares in the ring of squares adjacent to the border ring, then it commands 9 of the remaining 63 squares. Similarly, if placed on one of the 12 squares of the next inner ring the bishop commands 11 of those remaining, and if placed on one of the 4 center squares the bishop commands 13 of those remaining. Thus the checking power of a bishop is:

$$\frac{28}{64} * \frac{7}{63} + \frac{20}{64} * \frac{9}{63} + \frac{12}{64} * \frac{11}{63} + \frac{4}{64} * \frac{13}{63} = \frac{5}{36}$$

Using similar logic, the checking power of a knight can be computed as 1/12 and the checking power of a queen as 13/36. Converting the computed checking power to integers yields:

Piece	*Checking Power*	*Relative Point Value*
Queen	13/36	13
Castle	2/9	8
Bishop	5/36	5
Knight	1/12	3

What do these relative point values suggest? Perhaps that my own assumed values of 10, 5, 3, and 3 aren't very good. Perhaps that our definition of "checking power" should be improved. Perhaps that the chess masters are supplying relative point values from their experience rather than from reproducible computation.

Try computing some relative point values on your own. Can you define "checking power" so you can produce the relative point values you've assumed? One alternate definition of checking power begins as ours did, but then excludes these squares from the pieces control from which the king could take the piece. For example, consider the computing power of a castle using this alternate definition. If the castle is placed in any one of the 4 corners, it controls 14 of 63 remaining squares. However, if the king was placed in either of the 2 squares adjacent to the castle it could take the castle. Thus the castle really only controls 12 of the 63 remaining squares. If the castle is placed in any of the other 24 outside squares it again appears to control 14 squares, but 3 of these, those adjacent to the rook, must be excluded because the king could take the rook from these positions. Similarly, if the

Computer Glass Box continued –

matics, engineering, ecology, and physical sciences.

The challenge to educators, then, is to identify such topics suitable for embodiment as glass box programs, to search out the kernel concepts to be taught, and to lead students to better understandings of those concepts using a programming language.

References

[1] Papert, S. "Teaching Children Thinking", M.I.T. LOGO Memo No. 2, Oct. 1971.
[2] Iverson, K. E. "APL in Exposition", IBM Tech. Report No. 320-3010, Jan. 1972.
[3] Berry, P. et. al. "APL and Insight: The Use of Programs to Represent Concepts in Teaching", IBM Tech. Report No. 320-3020, March 1973.

castle is placed in any of the 36 remaining squares it appears to control 14 squares 4 of which must be excluded. Thus the revised computation power of a castle is

$$\frac{4}{64} * \frac{12}{63} + \frac{24}{64} * \frac{11}{63} + \frac{36}{64} * \frac{10}{63} - \frac{1}{6}$$

and so forth. The remaining computations and further modifications of the definition of checking power are left to you.

An article discussing chess related recreations would be incomplete without considering some of the many "tour" problems, particularly since they offer interesting programming challenges as well. There are many different types of tour questions—the fewest number of moves to accomplish a task, covering all squares once and only once, the minimum distance traveled, . . . Here we'll consider just three different questions.

First, can you determine a path traveled by a king such that each square is occupied once and only once? No, it's not that easy. There's one additional catch. As the king moves, number each square consecutively starting with 1. When you've finished, the squares will each contain one of the integers 1 thru 64. What's the catch? The resulting board with numbered squares must be an 8X8 magic square. A solution will be published in a future issue.

As a second problem, try moving the castle from one corner of the board to the diagonally opposite corner, again passing through each square once and only once. This one isn't quite as easy as a quick reading suggests, but you should be able to solve it in a reasonable period of time.

Finally, consider the standard "Knight's Tour" problem that is considered in so many articles on recreational chess. A knight must cover the entire board occupying each square once and only once. Don't, however attack this one with a knight and a chessboard—try something different. Write a problem that will find a solution for you. Kemeny (2) offers a BASIC program that attempts to find a solution by making random moves. The small sample of runs he published doesn't include a complete solution, in fact not one of 25 runs exceeded 50 squares before no moves were possible. Perhaps surprisingly, however, in 15 of the 25 runs the tour did pass thru over half the squares before terminating. Actually, when a knight's tour is attempted using only random moves, long tours are quite common but a complete tour is very unlikely indeed. How then can a program help? By adding a little bit more to the move selection than simply the choosing of a random number. When selecting a move, add one additional criteria. Always move to a square from which the knight will command the fewest squares that have not been occupied. Fewest is not a misprint, even if it does contradict your intuition. If several squares fit this criteria equally well, then select one of them at random. This additional criteria is reasonably implemented in a program, and although it doesn't guarantee a successful tour, you are apt to be surprised by the results!

Bibliography

1) Ball, W.W. Rouse, *Mathematical Recreations and Essays,* Macmillan and Co., London 1940
2) Kemeny, J.G. and Kurtz, T.E., *BASK Programming* Second Edition, John Wiley and Sons, New York, 1971

Big Surprise From Small Computers in Chess Matches

Good things come in small packages—especially when computers compete in chess matches. In two matches here and abroad, small computers performed surprisingly well against giant competitors.

David actually conquered Goliath in an intramural chess match at Columbia University when a small Data General Supernova—about the size of an attache case—checkmated an IBM System 360/91—one of the largest in the world—in just 25 moves.

The Supernova, owned by Columbia's Department of Electrical Engineering and Computer Science, had a memory capacity of 32,768 bytes; the IBM system, part of the university's computer center, had a capacity of over 2 million bytes.

The Supernova's chess program was written by Professor Monroe Newborn of the Electrical Engineering and Computer Science Department and by student George Arnold. It is written in assembly language and uses a technique that determines the best move by searching between four and eight half-moves ahead, selectively analyzing about 1,000 terminal positions. A move is determined in about one minute—well within chess tournament rules.

The game lasted more than 90 minutes, but the Supernova gained a decided advantage on the sixth move: the System 360/91, playing white, blundered, and traded a knight for a pawn. One of the program authors noted that the computer saw the correct move (bishop takes bishop) but didn't realize that exchanging bishops would save the knight. Once the big computer decided it could not save the knight, it decided to pick up a pawn.

"Having exclusive use of the smaller computer, along with running the program on-line, helps offset the greater speed and capacity of the larger machines," Professor Newborn said.

In another surprising match, a Computer Automation Naked Mini computer using only 16,000 words of 16-bit memory came in 12th in stiff competition in the World Computer Chess Championships held in Stockholm, Sweden.

"I'm no expert at chess; in fact, I'm just an average amateur, but I love to play with computers. Even so, I was surprised, indeed," said Bob Prisen, Interscan Data Systems, Ltd., United Kingdom, who programmed the Naked Mini. He spent approximately 300 hours in an eight-month period and used BASIC assembler language.

The first-place winner, by contrast, was a large-scale English-built ICL 4/70 computer entered by the Moscow Institute of Control Science. The machine and its programming had been prepared by a team of 10 fulltime staffers for more than two years. The computer used a program called KAISSA.

During the Swedish match, the Naked Mini stayed in England. Communications between Prisen and the computer were established using international telephone lines and an acoustic coupler with a Teletype and an on-line visual display unit.

Prisen said he hopes that a European or British Chess Computer Championship can be arranged in the future. "If it occurs, I'm confident the Naked Mini computer will greatly improve its position in the Chess Computer League with a bit more core memory and programming effort," he said.

SNOBOL

by David Touretzky

SNOBOL is a string processing, pattern matching language, that breaks all the restrictions associated with numerical languages like FORTRAN or dp languages like COBOL. Here are some key points:

1. No fixed data types. A variable's type is determined by its values. Changing the value changes the type. The datatypes available are: string, integer, real, double precision, array, table, pattern, name, and compiled code. The table is a type of hash table: its subscript is a string instead of an integer.

2. The heart of SNOBOL is the concept of pattern matching. A SNOBOL statement is made up of one or more of the following fields:

```
label  string  pattern  =
   string  branch
```

Every pattern match either fails or succeeds. The branch section causes branching to a specified label based on the results of the pattern match. For example:

```
MYLABLE  CARD  'CAT'  |  'DOG'  =
   'ANIMAL'   :S(L1)
```

In statement MYLABEL, the variable CARD is searched for any occurrence of 'CAT' or 'DOG'. If the test succeeds, the string which was matched is replaced by the string 'ANIMAL', and, since the match succeeded, the program would branch to L1. If the test failed, the program would continue with the next statement. Pattern matching can be much more complex, and can include many levels of alternatives, calls to user or system functions, recursive pattern definitions, and self-modifying patterns. Many special characters are used in pattern matching operations. For example, the $ is an assignment operator.

```
TEXT SPAN('A')  $  C  =
```

will match the first contiguous string of A's in TEXT and assign that string to the variable C. In addition, the string will be deleted from TEXT, since a null expression appears to the right of the equals sign.

3. Concatenation is accomplished by writing expressions next to each other. For example:

```
MESSAGE  =  'THE BILL IS'
   (COST  *  1.05)  '.'
```

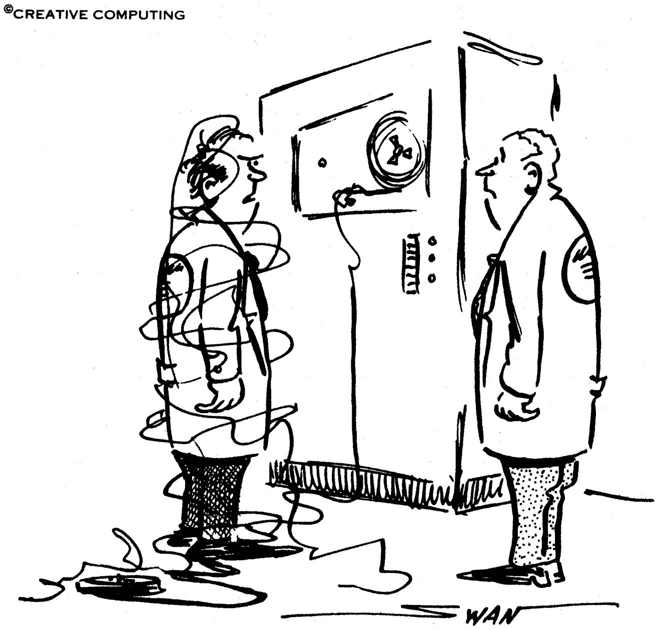
In this case, COST would be multiplied by 1.05, converted to a string, and concatenated with the other strings in the expression. Strings may be of any length. They expand and contract through pattern matches or assignments.

4. Complete debugging tools are available, including traces, variable dumps, and program controlled error handling. Interactive implementations, such as SITBOL, allow convenient debugging as the program executes.

5. As you have seen, SNOBOL has no verbs, no data restrictions, and no artificial constructs like blocks, procedures, or cases. The language is based on special symbols, such as =$.★?&:()@, and system functions which generate patterns (such as CONVERT). There are also some system variables, such as &ALPHABET which contains every possible character, &STLIMIT, the maximum number of statements that may be executed and &ERRNO, the code number of the most recent error interrupt.

6. SNOBOL is available on many machines, including the IBM 360, the DEC system 10, and UNIVAC 1108. Two extended versions are commonly in use: SPITBOL and SITBOL. The best introduction to the language I have seen is the *SNOBOL 4 Primer*, by Ralph and Madge Griswold, published by Prentice-Hall.

7. SNOBOL can be used for many things, including file manipulation, database editing, music generation, language analysis and translation, game theory, computer assisted instruction, and the simulation of automata.



"And where were you when the FORTRAN hit the fan?"

A Smalltalk Airplane Simulation

by Bruce Horn

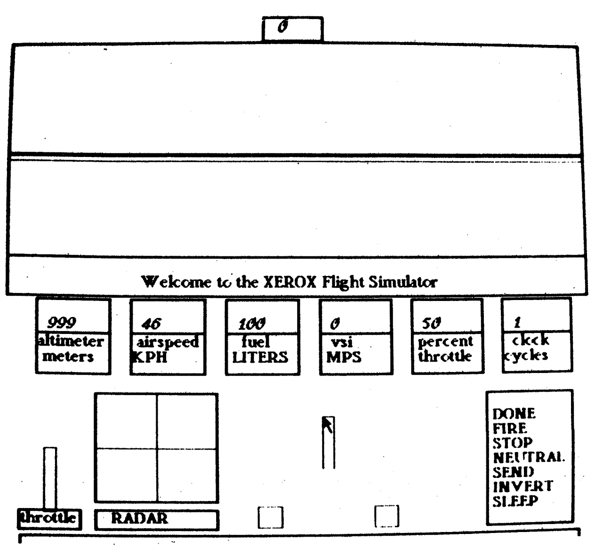
My name is Bruce Horn, and I am a junior at Gunn Senior High School in Palo Alto, California. I have been programming for about 3 years, mostly in Basic or Algol on Hewlett-Packard systems, such as the HP-3000. I would usually write simulations of other languages in Basic, or some type of mathematical program such as a program to plot equations.

At the end of the first semester, one of the math teachers asked me if I would like to get some Exploratory Experience credit (a program for work experience without pay) for working with computers at Xerox. I remember visiting the Xerox Palo Alto Research Center (PARC) one time before, so I decided that it would be a great experience. Since then, I have been working on one major project: an airplane simulator.

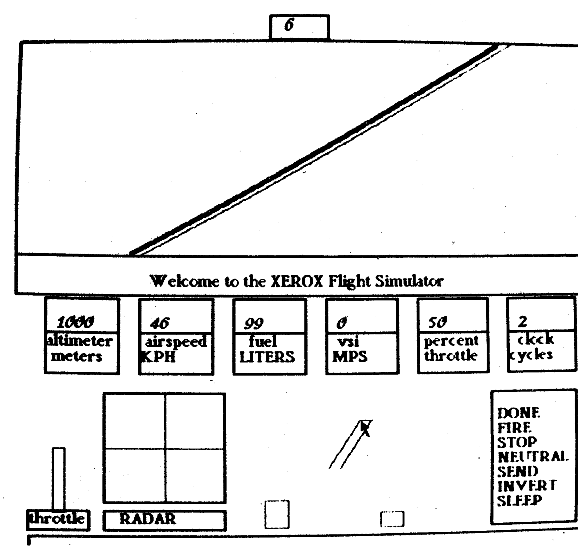
The computer that we use at PARC is a single user computer with a disc drive, keyboard, a graphics display, a five finger keyset and a pointer called a mouse. Since it is not timeshared, the machine responds quickly. It uses a totally new language called Smalltalk.

My idea of an airplane simulation is a program that would totally imitate the movements and attitude of an actual airplane, including the instruments used and the actual position of the horizon that you see. An ideal simulation on a computer would have to be realistic enough to be interesting. This would involve realistic output so that you could really see what is going on, and convenient input (stick, throttle, rudder) so that you can control the simulated airplane easily.

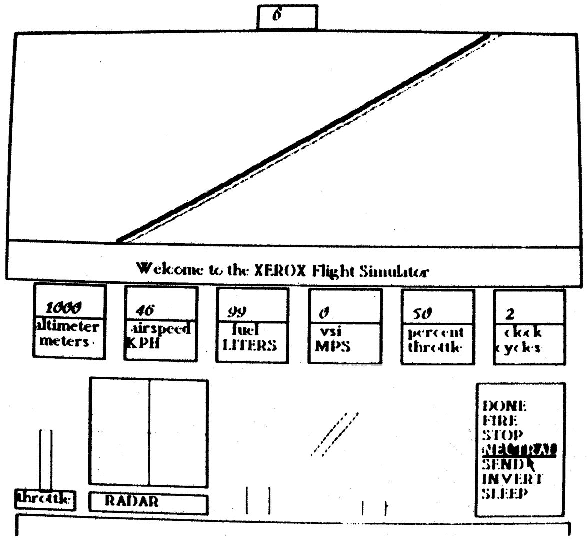
The basic idea behind Smalltalk is the concept of *classes* and *instances*. Programming in Smalltalk means that you



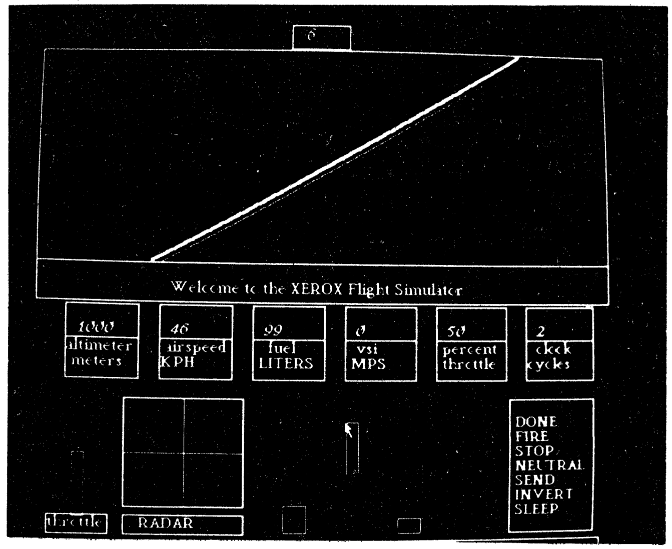
1. Photograph of the display screen showing the airplane flying straight and level (with no bank).



2. Airplane is in a turn — banking and depressing the right rudder pedal.



3. Selecting a command from the menu (bottom right of screen) — command word is inverted when pointed to. I am pointing at 'neutral,' a command to reset any control.



4. The whole screen is inverted — black changes to white and vice versa.

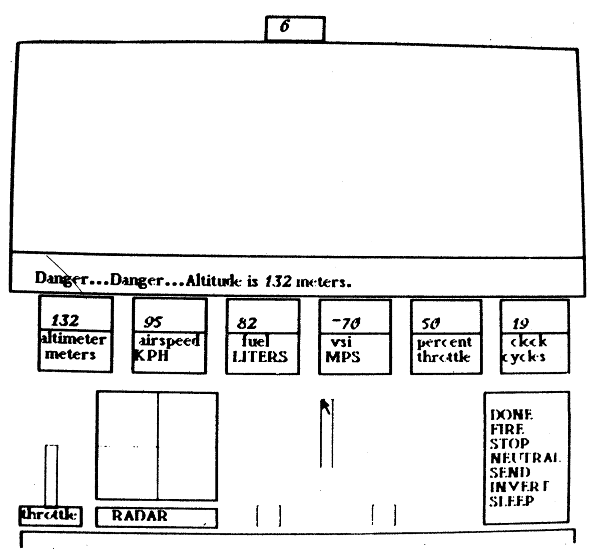
define what you mean by a group of objects by specifying the action each member of the group can take and describing properties that distinguish each member of the group.

Since there are many different instruments in an airplane, but they do nearly the same thing, I defined one class called *instrument* and created instances of that class for the different types of instruments. Each *instrument* is totally individual, but still retains the characteristics of the class. For example, each has its own location on the screen, a label for the instrument, and a value to display. Each *instrument* understands certain messages. Each does the same thing when it receives that particular message, but each instrument can be controlled individually. Each instance is different from another instance in the sense that each has different characteristics. However, each instance of the class called *instrument* has the same general form. A common example is the class of human beings. Every person is a human being, but each person may have different hair color, height, and weight.

The airplane simulator uses the mouse, keyset, and the keyboard together to send messages to the airplane to control its instruments. The mouse controls the position of the stick for pitch and roll, and it is used to select items from a menu of commands. A menu is a list of commands that can be executed by pointing at the command word with the mouse and pushing a button (Picture number 3). The keyset controls the setting of the rudder pedals and throttle, and uses different combinations of the five keys to control other operations in the simulator. At this time, the user sets the controls and requests a display of the state of the airplane with these control settings.

Later, I will have the airplane continually change state. Now that I have created the class *airplane* (the set of instruments and their display), I can create many instances, each running on a different computer. I hope to change the program so that other computers in the room can communicate with each other for a dogfight situation, complete with missiles and a working radar. The ultimate version of the simulator will have a more insteresting horizon that will change with the heading of the airplane (i.e., a total 360 degree panorama can be seen, including mountains and rivers) complete with perspective so that the actual position of the airplane relative to objects is more apparent.

Smalltalk is a very easy language to work with, with no limits to the complexity of programs. The concept of *classes* and *instances* is a very powerful idea and can be used very easily to simplify very complex problems. This is just not possible in Basic, no matter how advanced the version is. Therefore, a normal Basic program will probably look more complex than the same program written in Smalltalk, and, therefore, most likely harder to understand. The line-by-line execution of Basic is harder to understand, with the problem of if . . .thens and gotos interrupting the flow of the program. If this simulation was written in Basic, just for the instruments there would have to be vectors storing the values of the instruments and their labels, since all variables are local in Basic, and any part of the program that would access the instrument's data would have to know which vector and what element of the vector. In Smalltalk, only the class would have to know the explicit details of where all the data was. By using nesting and different levels in the program, Smalltalk eliminates the problem of having to jump around in the program because of a conditional statement or goto. Overall, I think that Smalltalk is easier, more efficient, and more interesting to use than Basic.



5. Just before a crash. You can't see the horizon because the airplane is facing the ground.

Non-Human Intelligence

Pushing into the future it is inevitable that we humans will be confronted with a much more challenging array of choices, problems, and technology than we have today. Not only that, but we will very likely be confronted with additional types of intelligences—from machines that we ourselves build to extraterrestrial intelligent beings. It is appropriate, therefore, that this issue of CREATIVE COMPUTING examine both artificial (machine) intelligence and also extraterrestrial intelligence. (Other than the name, they bear no relationship to each other.)

Somehow it seems appropriate that it is the computer that is helping us to leap ahead in our quest to seek out extraterrestrial intelligence. It was a computer aided prediction, for example, that recently helped S. Christian Simonson of the University of Maryland identify the closest galaxy to the milky way. It's ironic that astromers have identified thousands of distant galaxies and galaxy clusters up to 350 million light-years away, yet the light from our own milky way obscured our nearest neighbor, a small galaxy only 55,000 light-years distant.

The largest-scale project seeking life on other worlds is CETI (Communicating With Extraterrestrial Intelligence) sponsored by the Soviet Academy of Science. There's no telling what frequency an alien civilization might use to broadcast so listening will be done over the entire short-wave portion of the radio astronomy frequency range (1 to 100 gigahertz). The project will last from 1975 to 1990 and will use three kinds of search. The first will check each star within 100 light-years of the sun, and possibly if time permits, out to 1,000 light-years. The second will examine different galaxies in the local cluster. Finally, there will be an all-sky survey for signals from anywhere.

Turning to more distant observations, consider for a moment the light from the Coma galaxy cluster that has taken 350 million years to reach us. In other words, we are now observing that cluster as it was 350 million years ago. Conversely, if an observer on a planet in the Coma cluster had a telescope pointed towards the milky way, he would be seeing it as it was 350 million years ago, eons before the earth that we know today. But 350 million light-years is nothing; the Arecibo radio telescope has recorded signals from quasars more than 7,000 million light-years away, that are receding away from us with velocities of more than 150,000 miles per second. From these observations and the relation between the distance and the speed of recession, we can calculate how long ago all the matter of the universe was concentrated in one, immense, incredibly dense mass. The answer is about 10,000 million years ago.

Mind boggling, isn't it, that our observation of quasars today takes us back three-quarters of the time to the beginning of the universe? But if we can observe objects three-quarters of the way back to the beginning of the universe, why not go further? And quite accidentally that has happened. Bell Labs, in testing new ultra sensitive radio receiving equipment working at wave length of 7 cm., found radio noise 100 times stronger than the expected noise level of the equipment. More recent tests in a rocket and high altitude balloon confirmed this radiation. This noise, in fact, is the emission from the original primeral fireball of the universe. Talk about a big bang!

Perhaps we can't **travel** back in time but we can **look** back. Way back! The next time you gaze up at the stars, why not ponder some of these questions. Was there space in which the embryonic universe existed? How and why did the universe fly outward? (The dense concentrations of matter that existed at the beginning of the universe are similar to those existing in a black hole from which nothing can escape.) Will the universe expand forever? Or will it again collapse to a mass of infinite density? Is there a mirror universe composed of anti-matter which is contracting as our universe of matter expands? And what is the position of humankind in this vast scheme?

David H. Ahl

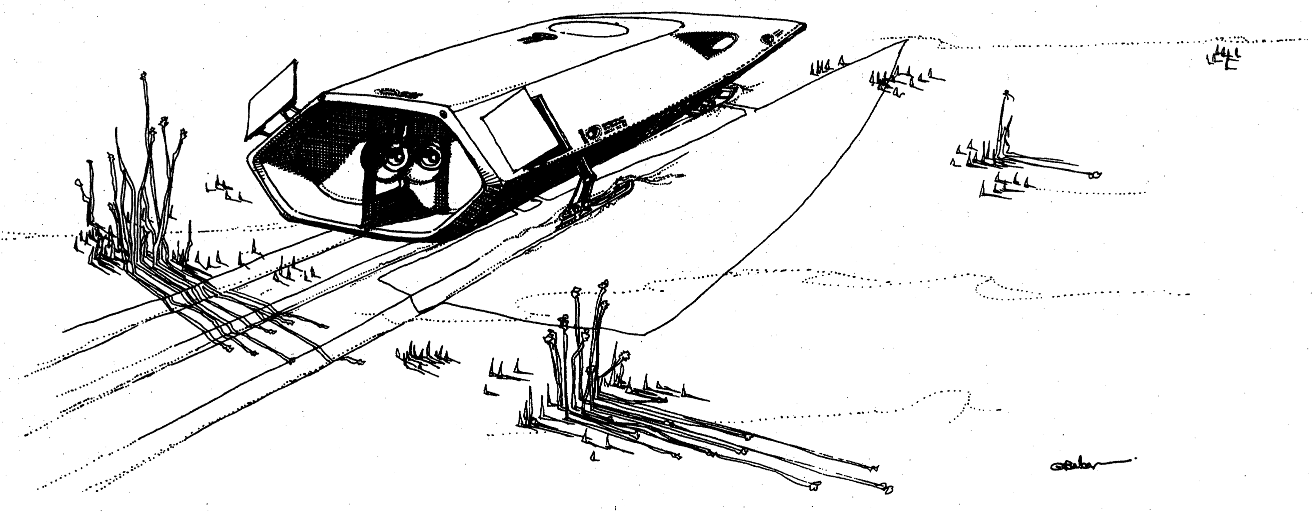


Illustration by George Beker

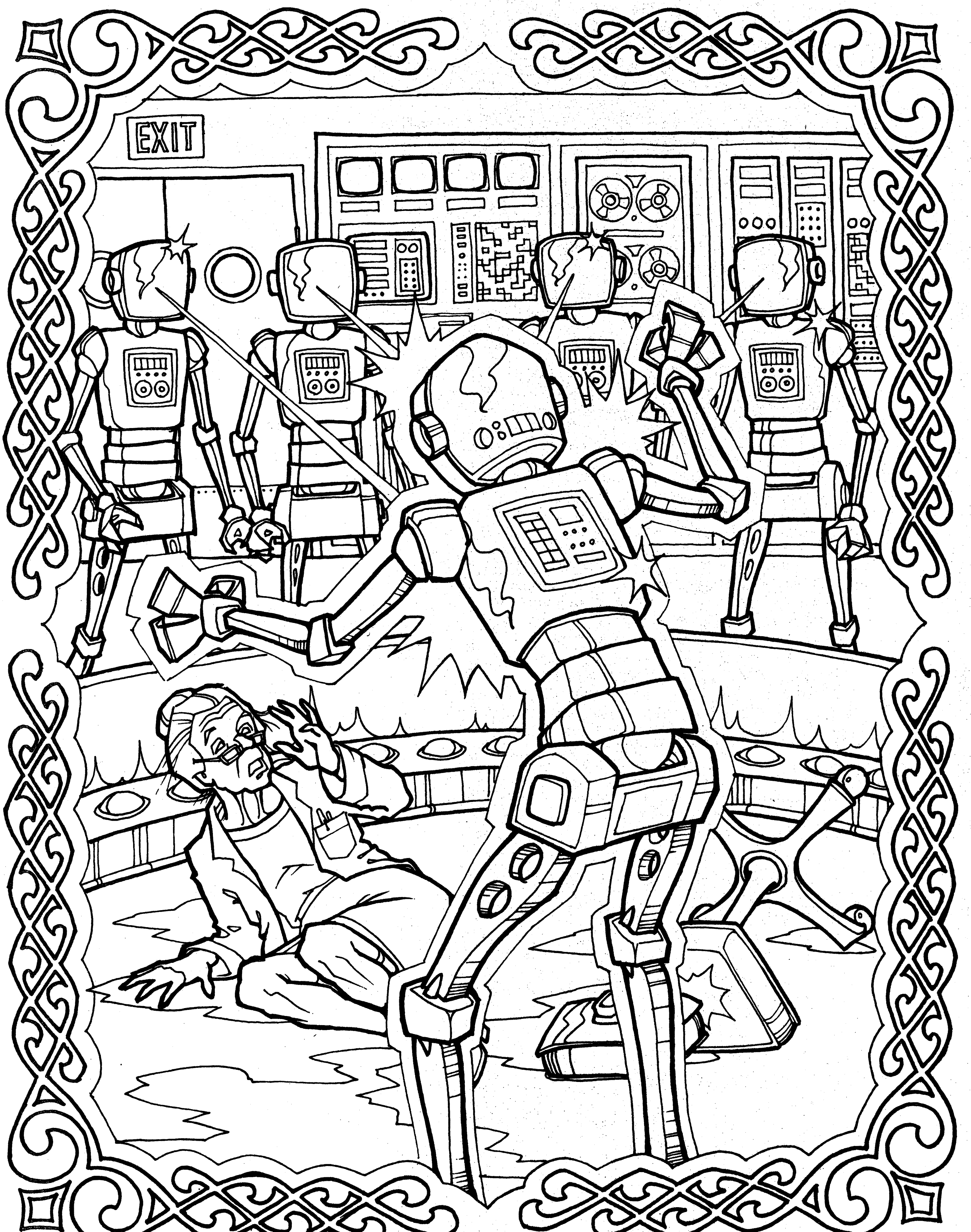


Illustration by Mark Savee for *I, Robot* from the coloring book, *Science Fiction Anthology*, Troubador Press, San Francisco, CA.

An Esoteric Ethical Excursion

by John Lees, University of Missouri-Rolla

I had volunteered to review Robert Heinlein's *The Moon Is A Harsh Mistress* for *Creative* because one of the central characters in the book is an intelligent computer, capable of speech and clearly possessing "free will." Since I have been an avid reader of science fiction for as long as I have been reading anything, rereading *Mistress* continually brought to mind all the other science fiction stories I have read which had as characters intelligent computers. After a while I realized that a great many of the stories I could remember contained some kind of reference to intelligent machines, computers, androids, cyborgs, robots or some type of artificially constructed sentience. [This probably represents a bias on my part—this is one type of fiction which appeals strongly to me.]

Now science fiction writers have had a great deal of luck predicting what path our technological evolution will take. Nuclear power, lasers, synchronous communications satellites and of course space travel have all been predicted well before they became realities. Needless to say, a lot of worthless, totally impossible predictions have also been made; hindsight always excels foresight. Anyway, I am convinced that hidden somewhere in all the garbage and noise of science fiction is the form which our future sentient companions will take. What will it be?

I think I may know, and I'm afraid the credit may have to go to Isaac Asimov for his 1940s creation of the positronic robot. [Isaac already has too much fame for his own good.] A quote from the introduction to *I, Robot,* Asimov's 1950 collection of his robot stories:

> "All that had been done in the mid-twentieth century on 'calculating machines' had been upset by Robertson and his positronic brain-paths. The miles of relays and photocells had given way to the spongy globe of plantinumiridium about the size of a human brain."

When I reread that a few days ago, I sat back and thought, "hmmmm."

I realized that Asimov had started writing his positronic robot stories before even the transistor had been thought of! I looked for a real-world parallel to the above quote and it was not hard to find. We don't have "positronic brains", but we're not too far away from having massive computer power in a globe about the size of a human brain.

Compare the ENIAC vacuum tube computer, which filled a room with 18,000 tubes and became operational in 1945, with Hewlett Packard's HP-65 hand-held card reading calculator. Or compare Digital Equipment Corporation's original mini, which filled a cubic meter, with their recently introduced PDP-8 on a single circuit board. Look at the direction of technology: microprocessors, miniature densely packed memories, low power high efficiency circuits. Throw in the opinion of Capt. Grace Hopper and others that the computer of the near future is going to have an architecture of interlinked but asynchronous microcomputers (the human brain has *got* to work this way) and what do you have?

You have a generation of very small computers that perhaps begin to approach the complexity needed for "sentience." Lets say we have a circuit board covered with microprocessors and micro-program stores and another thingie, probably more of a block, which is a very dense high speed random access memory, no doubt one of the new storage technologies. Now take the microprocessor board and "crumble" it around the memory. Maybe it's a flexible circuit board, maybe just a wiring network encapsulated in potting compound, who knows yet? It will take up less space this way and provide equal access time to the memory for all the microprocessors.

Does humankind have the right to create a race of slaves? For make no mistake—if it is merely a question of technological development—we can do it.

Now to cool this hardware the easiest thing to do would be to simply immerse the whole thing in a container filled with coolant. It's a delicate and expensive creation. So put it in the strongest type of container, a spheroid. Attach I/O gear, run power leads to the power supply, run coolant pipes to the refrigeration unit—these can be conveniently housed in a box below the "brain." Add locomotion. Energize. Presto Chango! Welcome to the age of intelligent robots!

There are a few technological problems to be overcome before this updated fiction becomes reality, but there is an even larger problem which *must* be solved before my scenario comes alive. Fellow sci-fi fans will realize that I've failed to include the most important aspect of Dr. Asimov's creation: the Three Laws of Robotics. I am very much afraid that I do not see how to include them.

The Three Laws of Robotics

1. A robot may not injure a human being, or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings except where such orders conflict with the first law.
3. A robot must protect its own existence as long as such protection does not conflict with the first or second law.

According to Dr. Asimov, those three laws are inherent in the positronic brain, and such a brain without the First Law is fundamentally unstable. Unfortunately, here in the real-world parallel, things don't work that way. All computers built to date have some form of the Second and Third Laws, although not always in that order. Of course no one has yet manufactured a computer or developed software that remotely qualifies for the label of intelligent or sentient.

But it will happen. How do we instill the First Law in a computer? Remember that Asimov himself hedges around the First Law in some of his later stories. Should the First Law be applicable to your run-of-the-mill intelligent computer, or only to robots; computers with locomotive capability? And how about this one: If we succeed in creating another intelligence, a fellow sentient being, do we have the moral right to ourselves impose on it such a set of laws?

Does humankind have the right to create a race of slaves? For make no mistake—if it is merely a question of technological development—we can do it. There is already at least one other semi-sentient species on Earth with us, the Dolphins. Will we treat another species any better than we have treated the Dolphins?

Now I will admit that this is a set of highly speculative questions, to say the least. But it is a set of questions that I would prefer that we have answers to when the time comes. One way or another, we are going to run into another intelligence before too much longer. It may be an intelligence which we create, it may be contact with an extraterrestrial intelligence, it may be the simple realization that there is already another intelligence on Earth, but we will not remain alone. I hope that we will not be completely unprepared when the times comes.

THE THINKING COMPUTER

by Bertram Raphael
Director, Artificial Intelligence Center
Stanford Research Institute

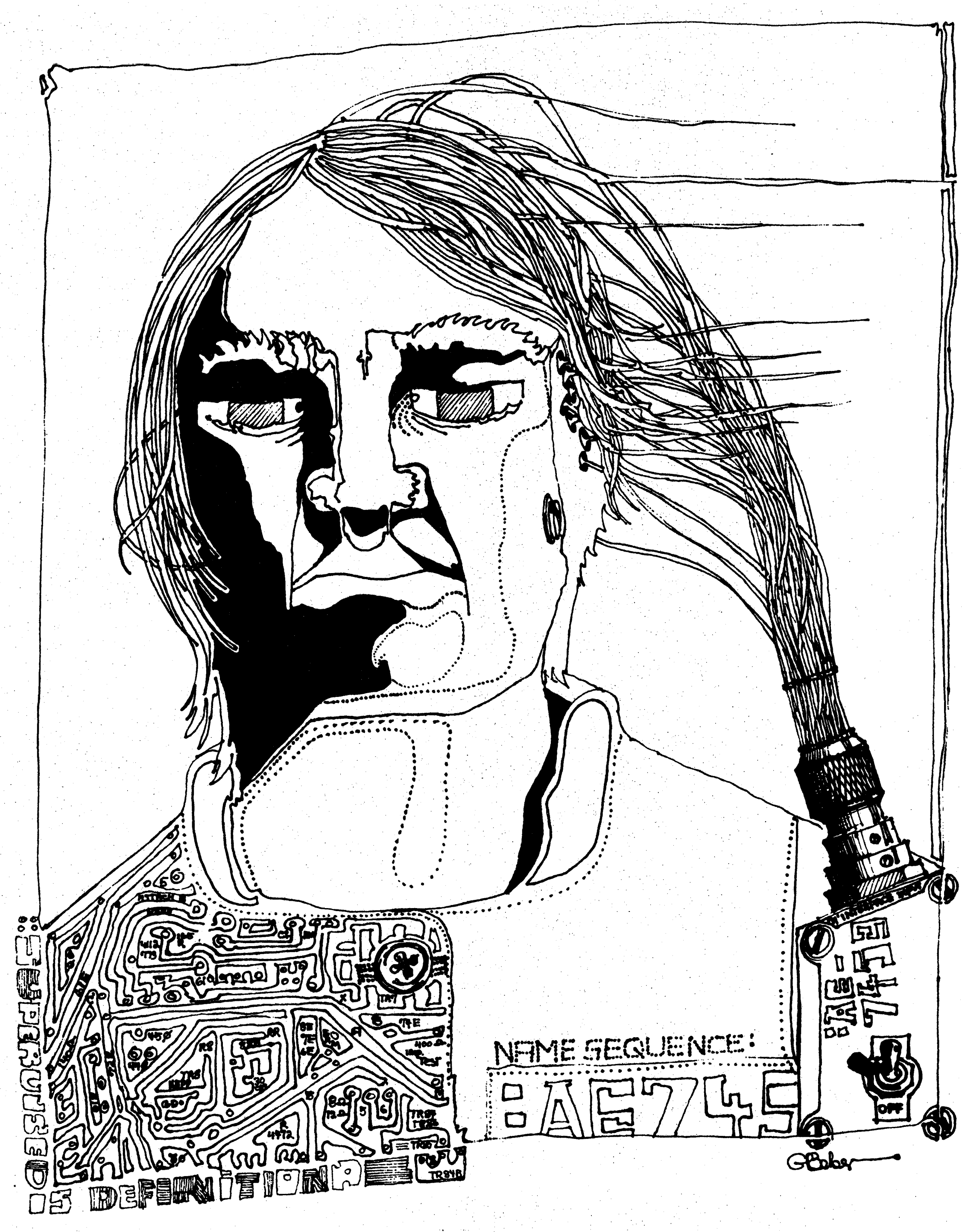


Illustration by George Beker

Misleading Myths

Many people believe that computers are inherently stupid, and think that even a suggestion that computers might be made smarter is ridiculous. This belief is so widespread that most people never even consider the many ways in which smarter computers might help them. Misconceptions about a computer's limitations seem to be based upon two widely accepted but basically untrue premises. Let us examine these myths. By pointing out some of their fallacies, perhaps I can open your mind to the fascinating prospects for smarter computers.

THE ARITHMETIC MYTH. *A computer is nothing but a big fast arithmetic machine.*

Computers are arithmetic machines, certainly; almost every computer has wired into it the ability to add and subtract. But are they "nothing but" arithmetic machines? Certainly not. Take the reference manual for any computer, and scan through its "instruction set": the collection of basic operations it has been designed and wired to perform. You will see a few, perhaps as many as ten or twenty, operations that bear some close resemblance to arithmetic—e.g., ADD, DIVIDE, FLOATING SUBTRACT, MULTIPLY STEP, and so on—but you will also see many, perhaps one or two hundred, operations that have relatively little to do with arithmetic—e.g., STORE, LOAD, TEST, SHIFT, READ, WRITE, REWIND TAPE, SKIP, MOVE, MASK, MATCH, TRANSFER, and so on. Much of the time that any computer works on any problem, the computer is positioning, comparing, moving, choosing, copying ..., but it is not doing arithmetic. Rather than calling a computer "nothing but a big fast arithmetic machine," it is much more accurate to say that a computer is *a big, fast, general-purpose symbol-manipulating machine.*

THE STUPID COMPUTER MYTH. *A computer is an obedient intellectual slave that can do only what it is told to do.*

This second myth is even more persistent than the first one, and even more damaging in the way it tends to constrain our thinking. Suppose I gave you the pieces of a jigsaw puzzle and told you, "by the way, these pieces cannot be fitted together." Would you try very hard to fit the pieces together? Why should anyone try to build a smart computer, if he is told over and over again that computers are inherently stupid?

The stupid-computer myth has been repeated and generally accepted for more than a hundred years. In 1842, after Professor Babbage of Cambridge designed his Analytical Engine, a large-scaled mechanical digital computer (which unfortunately was never completed), his friend Lady Lovelace wrote, "The Analytical Engine has no pretensions to *originate* anything. It can do *whatever we know how to order* it to perform." There is no question that Lady Lovelace's argument, and all the subsequent versions of the stupid-computer myth, are true, in a certain literal sense: a computer must be given its program of instructions, and it will always do exactly what those instructions tell it to do (unless, of course, one of its circuits fails). And yet this basic truth is not a real restriction on the intelligence of computers at all.

This article consists largely of material from the book, *THE THINKING COMPUTER: Mind Inside Matter*, by Dr. Bertram Raphael which will be published early in 1976 by W.H. Freeman and Company.

As novel sources of information, amusement, or artistic experiences, the potential for us to benefit from thinking computers is limited only by our imaginations.

The claim that a computer "can only do what it is told to do" does not mean that computers must be stupid; rather, it clarifies the challenge of how to make computers smarter; we must figure out how to tell (i.e., program) a computer to be smarter. Can we tell a computer how to learn? To create? To invent? Why not? I'd bet even Lady Lovelace would have agreed that the task of figuring out "how to order" a computer "to originate" something would be a fascinating and meaningful research challenge.

Progress in "artificial intelligence," the study of how to make computers smarter, is now enabling computers to apply a wide range of problem-solving methods; to communicate in ordinary English; to perceive the physical world; and to combine such abilities into flexible systems that perform useful tasks. The following paragraphs review some of this progress.

Problem-solving Methods

How do most people solve common, everyday problems? Suppose Mr. Pollack is driving to a ski resort in his little foreign car. On the way he encounters a snow storm, and finds he must mount his brand new tire chains on the wheels of his car. This problem—how to mount the chains on the wheels—can be divided into many little subproblems. Do the chains go on the front or the rear wheels? Should they be wrapped around a wheel by jacking the wheel off the ground, by driving the car onto the chains, or by figuring out how to use the funny little "mounting tools" that come with the chains? Which side of the chains should be up? How does the peculiar linking mechanism work? And so on. Mr. Pollack must solve these problems as quickly as possible, so that he can accomplish the task without freezing his fingers and soaking his clothes, and so that he can still get to the ski area without missing too much of the day's activities. Well, exactly how is this kind of problem usually attacked? By encoding the known facts into mathematical axioms, and using theorem-proving methods? Not likely! Instead Mr. Pollack (and millions of others) use *informal* problem-solving methods.

Informal problem-solving methods are especially intriguing because of their extreme generality. Problem-solving methods that most scientists develop work only in highly specialized, highly technical areas: e.g., a method for solving second-order linear differential equations with constant coefficients, or a method for estimating the distances of stars less than twenty light years away. Informal problem-solving methods, in contrast, seem to be applicable to a wide variety of problems, most of which may be brand new to the person using the methods: even though Mr. Pollack may never have been called upon to mount tire chains before, he need not be at a total loss as to how to proceed. Is there some standard way of viewing any task, that enables people to apply their reasoning abilities with flexibility to any problem that arises? Psychologists have developed various approaches to explaining such complex cognitive behavior. One approach that has been embodied in a computer program, called the General Problem Solver (GPS), demonstrates a way to redirect a single central mechanism to a variety of different tasks with a minimum of effort.

An important feature of GPS is that it separates task-dependent information—the detailed description of a particular problem, the actions or "operators" available for use in solving the problem, and the desired results or "goal"—from task-independent reasoning methods that may be useful for many different types of problems. Various strategies for finding an effective sequence of operators can be proposed, and the GPS computer program, developed at Carnegie-Mellon University, is a tool for comparing and experimenting with such proposals.

GPS has a major advantage over more formal search algorithms. It does not have to select operators in sequence from the first operator, which is to be applied to the initial problem, to the last operator, which is to reach the goal. Instead, it looks at the initial problem and the goal, and goes to work on trying to reduce the most important difference between them. The operator that succeeds in reducing this difference might eventually have to be applied somewhere in the middle of the complete sequence of operators that solves the problem. By deciding upon this operator first, the problem solver overcomes a major hurdle and replaces the entire task by two simpler subtasks: getting from the initial situation to one in which the chosen key operator can properly be used, and getting from the state that exists after that operator is used to a final solution.

Consider Mr. Pollack's chain-mounting problem again. A straight-forward search procedure for figuring out how to put on the chains would consider first the alternative actions that are immediately possible when he stops his car: wait for the snow to melt, or try to drive on without chains, or turn around and go back home, or get out of the car. If we assume he gets out of the car and is standing in the snow storm, his next choices might include: get back into the car, or open the trunk, or jump up and down to keep warm. If he opens the trunk, then he can get the chains out, or get the jack out, or get the suitcases out, or crawl in, and so on. Eventually, if he follows the most direct course of action, he will find himself lying in the slush under the car, with the chain wrapped around the wheel, and his fingers jammed up between the freezing axle housing and the hot exhaust pipe, trying to figure out how the new-fangled linking mechanism works.

The GPS approach might begin by observing that a key difference between having no chains on the wheels and having chains that work correctly on the wheels is that each chain must be linked onto a wheel. Therefore an understanding of the linking mechanism may be singled out as the first problem that must be solved. This problem can be tackled by studying the manufacturer's directions or by experimenting with the actual chains, while seated in the warm dry car. Once the linking mechanism is understood, the next problem is how to get to a situation in which it is appropriate to close the links, so he might then focus attention on the wrap-the-chain-around-the-wheel problem. Thus the overall task is broken down into a sequence of progressively less crucial subproblems whose solutions each fill in a different portion of the overall solution.

Around 1969 scientists at Stanford Research Institute developed a problem-solving system to control an experimental robot. This system, called STRIPS, combines some of the best features of deductive theorem-proving methods, with informal GPS-like problem solving. One of the difficulties with using GPS was that the user had very little guidance as to how to represent his problem in the computer. GPS dealt purely with abstract notions such as objects, operators, and differences, and its success depended to a great extent upon special characteristics of those objects, operators, and differences, that the user invented for himself. The principal contribution of STRIPS is to embed into a GPS-like framework a set of specifications for the nature of objects and operators, and a resulting automatic method for obtaining differences. Of course, the user must still construct a specific representation for each specific problem, but STRIPS at least tells him the form that that representation must take.

Current directions in problem-solving research include the development of new programming languages that incorporate earlier problem-solving techniques; studies of ways to use man-machine systems more effectively in cooperative problem-solving activities; methods to allow problem-solving systems to use computer simulation techniques when appropriate; and the growth of large data bases so that problem-solving systems can have access to the general knowledge they require.

Natural Language

Linguists are actively working on theories to explain the nature of language and its semantics. However, technology does not usually wait for theories to be completed. While the linguists carry on their theoretical studies, computer scientists have also been studying how computers can be made to understand natural language. These studies have been conducted from an experimental engineering point of view.

Many of the experimental language-processing programs fall in a general category called *question-answering systems*. A question-answering system may be defined as any computer program that understands the information typed into it, and demonstrates that it understands by answering questions about the information. The ideal question-answering system should be able to: (1) accept facts and questions, and make appropriate responses, all in the form of natural English; (2) store, remember, and make efficient use of a large amount of data—at least thousands of elementary facts; (3) answer questions that require it to figure out the logical consequences of the facts stored explicitly in its memory; and (4) operate conversationally—e.g., via a time-sharing computer terminal—without frustrating delays. Although no system yet developed has all four of these capabilities a significant degree of success in each of the four areas has been separately achieved by various systems. In the next few years we should begin to see these capabilities combined and improved, producing the first true, complete question-answering systems.

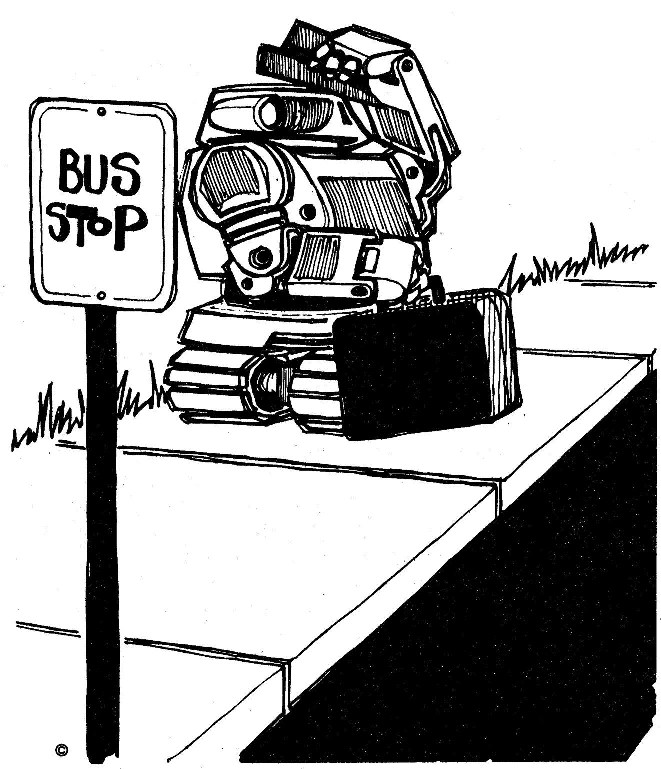
We would like computers to be able to understand not only typed and printed, but also *spoken* natural language. For many years research in the field of "speech recognition" focused upon identifying individual words purely on the basis of their sounds, and progress was limited. Now scientists recognize that understanding spoken language involves using many sources of knowledge—such as knowledge of vocabulary, syntax, and subject matter—in addition to the perceived sounds themselves. Speech-understanding systems now under development integrate such multiple sources of knowledge about language in order to come up with an accurate understanding of what has been said.

Perception

Suppose you are about to open some presents. Do you need to unwrap each item completely and look at it in a strong light in order to recognize what it is? Not usually. One handlebar sticking out of a large, formless wrapping is enough to identify the bicycle you had been expecting. If it's Christmas morning and a small flat box has a tag showing that it came from Aunt Agnes, you might know it contains another hideous tie. On the other hand, if the occasion is your bar-mitzvah, then you can be pretty sure that every small flat box contains another pen-and-pencil set. If you don't come to any such quick conclusion, you might examine the mysterious package more closely, look at the tag or postmark, lift it to feel its weight, shake it to see if it rattles or sloshes, and you usually will have a pretty good chance of perceiving what's inside without ever seeing it. Similarly, it is unfair to expect a computer to recognize real objects unless it first knows something about the expected characteristics of the objects, such as their size, shape, color, and the normal physical relationships among them.

Many of the past computer-vision projects tried to "simplify" their tasks by aiming their TV cameras only at artificial objects like boxes and wedges, which had straight-line edges and clear mathematical descriptions. Unfortunately, such objects have few expected sizes or shapes, no normal physical relationships, and rarely any context to guide the recognition process. Because of this, paradoxically, the attempt to simplify may actually have made the recognition problem more difficult.

Current research is turning to more natural pictures that may incorporate curved objects and complex surroundings. "Scene understanding systems" are now being built that coordinate the use of several kinds and sources of knowledge in order to solve complex problems. For example, knowledge of illumination, distance measurements, color, spatial relationships, and physical constraints, can all contribute to the accuracy of the interpretation of visual data.



Robots

I am not going to define the word "robot" here, because of the wide range of interpretations it has. The following examples indicate the general kinds of devices that we shall consider. Without getting enmeshed in the technical details of how they work, let's look at what some of these systems were capable of doing a few years ago.

At Hitachi Central Research Laboratory a TV camera was aimed at an engineering plan drawing of a structure built out of various-shaped blocks. A second camera looked at the blocks themselves, which were spread out on a table. The computer "understood" the drawing, reached towards the blocks with its arm, and built the structure.

At MIT, the camera was not shown a plan; instead, it was shown an example of the actual structure desired. The computer figured out how the structure could be constructed, and then built an exact copy.

At Stanford University, the hand obeyed spoken directions. For example, if someone said into the microphone, "Pick up the small block on the left," that is precisely what the arm would do.

At the University of Edinburgh, a jumble of parts for two wooden toys was placed on the movable table near the camera. "Freddy," the Edinburgh hand-eye-table robot system, carefully spread out the parts so that it could see each one clearly, and then, with the help of a vise-like work station at one corner of the table, assembled first the toy car and then the toy boat.

At SRI, Shakey the mobile robot was told to "PUSH THE BOX OFF THE PLATFORM." Shakey had no arm, and realized that he could not reach the box unless he was on the platform with it. He looked around, found a ramp, pushed the ramp up against the platform, rolled up the ramp, and then pushed the box onto the floor.

Recently, robot researchers have been concentrating their efforts upon specific technical problems that must be solved in order to create more powerful robot systems. Major developments coming out of current work include: (1) new hardware technology that is leading to more reliable and less expensive sensors, effectors, and computers; (2) new software technology, in the form of high-level programming tools and studies of how to structure the large knowledge bases that are essential for any intelligent system; and (3) prototypes of simple robot systems that can at least begin to perform truly practical tasks. For example:

At Stanford the hand-eye system that used to stack toy blocks can now assemble a real water pump.

At SRI a computer-controlled Unimate industrial manipulator arm with touch and force sensors can feel its way as it packs assembled pumps into a case.

At MIT programs are under development to enable a computer to inspect and repair circuit boards for use in computers, TV sets, and other electronic equipment.

Applications

As computers become less expensive and more widely available, society is becoming more dependent upon them to perform conventional bookkeeping functions. More important, however, is that as computers become more intelligent they can take on valuable new roles in the service of society. In education, computers constitute a rich new medium for a student's creative expression and experimentation. They can be used to demonstrate laws of physics on a dynamic display screen, to illustrate mathematical principles through the design of algorithms, and to carry on tutorial conversations. In psychology, computer models of mental behavior provide knowledge of how the mind works. In medicine, computers can model physiological and biochemical processes, and both store and deduce large numbers of facts about diseases, drugs, and treatments. In industry, computers can help both in the front office, scheduling activities and monitoring progress, and on the factory floor, directing automatic inspection, materials handling, and assembly systems. Such activities can both increase productivity and improve the quality of the goods produced. In mathematics and science, computers are beginning to function as intelligent assistants to professional scientists, performing such jobs as solving and simplifying symbolic equations, analyzing chemical compounds, and verifying the correctness of simple computer programs. As novel sources of information, amusement, or artistic experiences, the potential for us to benefit from thinking computers is limited only by our imaginations.

Primer on ARTIFICIAL INTELLIGENCE

by Lewis E. Garrett
Del Mar College, Corpus Christi, Texas

I. INTRODUCTION

Man has been building tools that help him in solving problems of his environment for thousands of years. The earliest tools were crude knives and axes. But today simple single purpose tools will not solve mankind's myriad complex problems. Many necessary tasks can no longer be accomplished by human intellect alone. So, with the advent of the electronic digital computer, attempts to construct a new problem-solving tool were begun.

Purpose of the Primer

The purpose of this primer is to describe artificial intelligence and to show the various approaches for attaining it. Information is presented on the necessary elements of artificial intelligence, the types of research that have been conducted, and the 'state-of-the-art' of artificial intelligence research. It is the writer's intent to inform interested persons on the many facets of artificially intelligent processes.

Scope

Readers of this primer are not required to be computer experts. The only requirement is an interest in 'thinking' computers. No attempt is made to describe in-depth computer programming techniques. The reader should realize that a digital computer basically has only two outstanding abilities: the ability to perform arithmetic computations and the ability to compare two quantities and determine which is the largest. Both of these operations are performed exceptionally quickly. The discussion of artificial intelligence will center on its parts as a function of the whole and will not give the reader the understanding necessary for writing a computer program that exhibits artificially intelligent behavior.

History

The term 'artificial intelligence' has, since its inception, come to mean the mechanization of thought processes. It can be classified into at least four distinct areas: game playing, language translation, problem solving, and pattern recognition. Work in these areas could not really begin until the advent of the general purpose digital computer in the early 1950's.

Each area of artificial intelligence has had dramatic early success followed by unexpected difficulties. These early successes prompted men like Herbert Simon to make enormous predictions about the future of artificial intelligence. Mr. Simon said in 1957: "... within ten years a digital computer will be the world's chess champion ...." This promise has not been realized.

Early successes were realized by Newell, Shaw, and Simon at Carnegie Institute of Technology. They concentrated on the simulation of human thought processes (artificial intelligence) with emphasis in the area of games and problem solving. One program known as the Logic Theorist, in 1957, was able to provide proofs of 38 out of 52 theorems from *Principia Mathematica*.

Another area of early success was in the mechanical translation of languages. In 1954 Anthony Oettinger devised the first mechanical dictionary for the translation of English into Russian. During the ten years following the development of this mechanical dictionary, about $20 million was spent on mechanical translation research by various governmental agencies.

Definitions

Artificial intelligence is best defined in the words of Marvin Minsky as: "... the science of making machines do things that would require intelligence if done by men." This definition reflects the core of this primer.

II. INTELLIGENCE

"Intelligence surely can exist only within very intricate structures." This statement by Marvin Minsky sums up rather neatly the problem of artificial intelligence. So it is that a study of artificial intelligence cannot be undertaken without a companion look at what constitutes 'natural intelligence'. This will lead to a better understanding of what must compose 'synthetic intelligence,' and how the two compare to each other.

Natural Intelligence

Some people think that humans have two qualitatively different resources for information processing. The first is a unit for controlling the functions of all the other units (a central processing unit). The operations of this central processing unit are carried out one at a time, that is, serially. The second resource is a network of interconnected elements which act in parallel. This means that the elements are available simultaneously for contribution to the information process. Another theory has man's central processing unit also performing operations in parallel. Although agreement is not unanimous on whether or not man's brain performs operations in parallel or in series, the nature of the second resource of the brain (network) is, for the most part, agreed upon.

Mechanics of the Human Brain. The human brain is composed of approximately 10^{10} neurons (nerve fibers). These nerve fibers come together to form junction points or what is known as synapses (Fig. 1). Neural physiologists are far from completely understanding how neurons operate and are interconnected but, many believe, the basic functions are performed by an 'on or off' process.

This 'on or off' function is an electrical process performed at the synapses. The nerve fibers conduct electrical impulses to the synapses. This results in the synapses being either 'excited' or 'inhibited.' Excitation occurs when the sum of several input pulses exceeds the 'threshold' voltage for that synapse. A synapse is inhibited when the electrical impulses are not sufficient to exceed the 'threshold' (Fig. 2). This stimulation or inhibition effect can also be thought of as the 'on or off' effect of the neurons on the synapses.

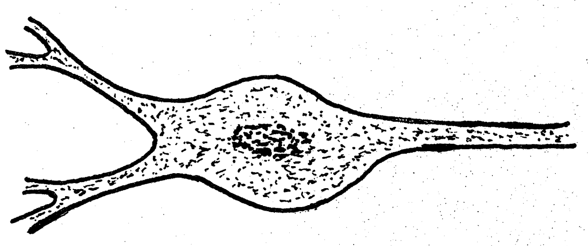


Fig. 1
SYNAPSE

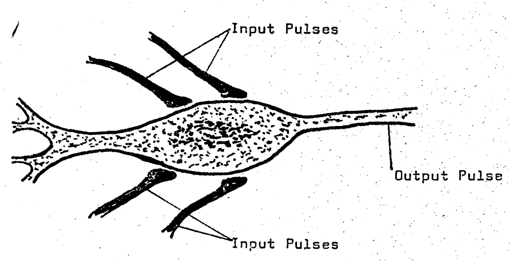


Fig. 2
SYNAPSE WITH EXCITATION

The trillions of connections between nerve fibers explain the human brain's ability to think. This is especially true since the human mind can use tens of thousands of nerve fiber chains simultaneously in performing an intelligent act. Although the individual actions which occur in this 'neural net' can be visualized and understood, the multiplicity of effect cannot be. If this process were understood, the question of how to create artificial intelligence would also be understood.

Human Behavior. Newell, Shaw, and Simon are convinced that free human behavior is based on a complex but determinate set of laws. Since the human memory capacity is estimated to be at least a million billion bits of information the truth of their conviction cannot be readily proved.

Many people think that human intelligence has evolved through a lengthy process of mutation and natural selection. Others think that this intelligence is based on natural neural networks alone. Whichever theory is used, natural intelligence appears to be essentially a trial and error process.

The words of Donald N. Streeter serve quite well to sum human behavior: "Man is inventive and flexible. He perceives, abstracts, and associates quite well, drawing on broad experience to make decisions and check reasonableness. However, he is forgetful of detail, inaccurate, and subject to boredom and fatigue."

Synthetic Intelligence

Any process which can be formalized so that it can be represented as a series of instructions can, in principle, be reproduced by a computer. This idea is the basis for the artificial intelligence researcher's belief in an eventual solution. Marvin Minsky states: "... that at least some mentalist description of thought processes can be turned into specifications for the design of machines or, what is the same thing, the design of programs."

Mechanics of the Digital Computer. The electronic digital computer is basically a block diagram consisting of three functions: input, process, and output. These functions are performed by various pieces of hardware: the central processing unit, the arithmetic logical unit, storage (memory), and the input-output devices (Fig. 3).

The central processing unit controls all the other elements of the computer. The storage of the computer holds instructions awaiting execution. The arithmetic logical unit performs comparisons and does arithmetic. And the input-output devices receive and issue information. All of these units, working in tandem, form the physical make-up of the electronic digital computer.

The actual processing within a computer is binary in nature, that is, it is controlled by the 'on or off' condition of discrete bits of information. This is accomplished by using electronic memory devices (core, semiconductor, bubble, charge couple memory, etc.) having either an 'on' or 'off' state.

Work done by a computer is controlled by its 'software. Software is a set of instructions for events to take place within a machine. These instructions are placed in storage where they are performed serially, that is, one at a time. Computers perform these operations in nanoseconds (10^{-9} sec.).

Synthetic Behavior. "Research scientists in Artificial Intelligence try to get machines to exhibit behavior that we call intelligent behavior when we observe it in human beings." In Dr. James R. Slagle's words, we can see that the search for artificial intelligence is really an attempt to duplicate human behavior. Thus it is that there are three schools of thought concerned with finding the answer. The three approaches are: artificial evolution, artificial networks, and heuristic programming.

Artificial evolution is an approach whereby computer simulated systems are made to evolve by mutation and selection. The main disadvantage of this approach is that it is practical only if artificial evolution can be made to proceed enormously faster than natural evolution.

Artificial networks are a large number of simple elements and their interconnections. Its main advantage is that, using this approach, the system is adaptive to new situations and can learn from experience. The disadvantage is that there is little prospect of making an artificial network as large as the network in the human brain.

The heuristic programming approach uses heuristics in attempts to solve problems. A heuristic is a 'rule-of-thumb' used to solve a particular problem. This technique, combined with the computer's computational power, has had a margin of success. But success is limited by the programmer's ability to conceive 'rules-of-thumb' for application to the problem.

Again, Donald N. Streeter's words are most apt for summing up: "Computer systems, compared to humans, are fast, accurate, and consistent in recalling and processing information, but are inflexible, requiring detailed pre-programming for all situations to be dealt with."

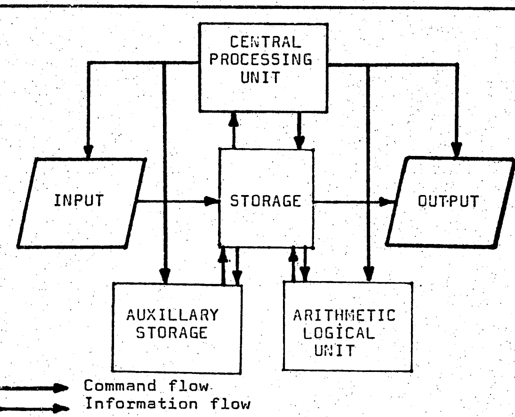


Fig. 3
COMPUTER DEVICE RELATIONS

Comparisons

The human nervous system is intricately more complex than that of the computer but is more sluggish in handling messages (Fig. 4). The reasons for this are found in the speeds with which electrical impulses are transmitted. Human nerve pulses last about one thousandth of a second whereas typical computer pulses last only a few billionths of a second. Hence, the human brain can process only 50 billion bits of information within a conscious lifetime, while the computer can process this same number within a couple of hours.

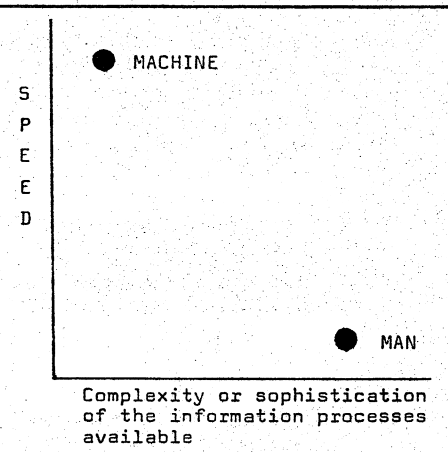


Fig. 4
COMPARISON OF MAN AND MACHINE

III. ELEMENTS NECESSARY FOR ARTIFICIAL INTELLIGENCE

A machine must have facilities for representing and analyzing its own goals and resources. There are three basic elements necessary to achieve true artificial intelligence: memory, pattern recognition, and learning.

Memory

It is assumed that the long-term storage of information and data in the brain is necessary to learning. Memory is, in actuality, a problem of recognition. This is true because facts are rarely at hand in the form they are needed. Man's pattern recognition of data is largely due to his fabulous memory system and its ability to classify information. If 225,000,000,000 computers (IBM 370/135 or equivalent) were connected together, they still would not achieve the memory capacity of the human brain.

Pattern Recognition

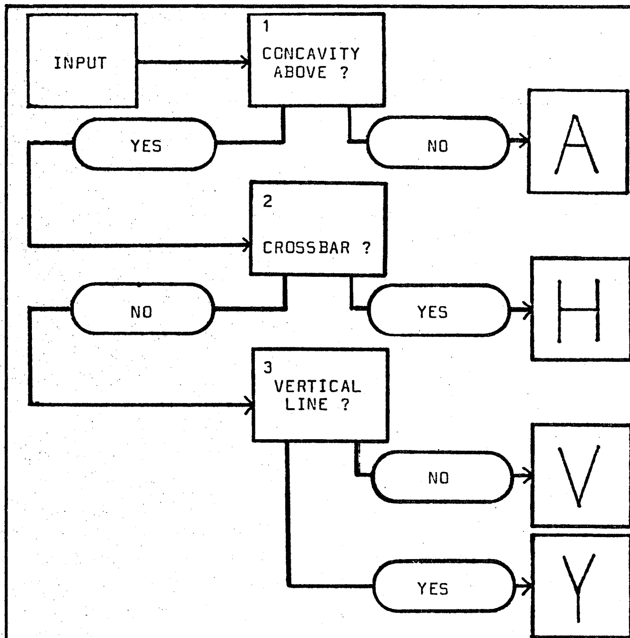
Many of the problems in artificial intelligence are in pattern recognition such as identifying printed letters (Fig. 5).

Pattern recognition techniques are necessary in order to cut down on the possibilities to be considered in solving a problem. Unless this is done, the search for a solution becomes exponential in growth, and soon outstrips the limits of the computer.

At present, pattern recognition programs do not even approach the flexibility of human pattern recognition abilities. Until they do, true artificial intelligence will not be possible.

Learning

Over 2300 years ago the Greek philosopher Aristotle studied the process of associative learning. For centuries man has been fascinated by this learning process. The two most important families of contemporary learning theory are: Stimulus-Response theorist and Cognitive field (or Gestalt) theorist.



Sequential-processing program for distinguishing four letters, A, H, V and Y, employs three test features: presence or absence of a concavity above, a crossbar, and a vertical line. The tests are applied in order, with each outcome determining the next step.

Fig. 5
PATTERN RECOGNITION

Stimulus-Response. Under Stimulus-Response, behavior is seen as a transaction between the stimuli that impinge an organism, and the resulting response. The Stimulus-Response theorist sees learning as a permanent relation between stimulus and response.

In the early 1900's American psychologist E.L. Thorndike formulated the Law of Effect—when a person repeatedly does something successfully, the neural pathways become reinforced; when a person repeatedly fails to do something successfully, the neural pathways become inhibited. Ivan Pavlov's famous experiments with the salivation of a dog when a bell rings illustrates this theory quite well.

Current computer programs, for the most part, follow the Stimulus-Response theory since the same input usually engenders the same output.

Gestalt. Gestalt is a German word which means a configuration has characteristics more broadly based than those of its parts. Gestalt psychology originated in Germany in the early part of the 20th century with four psychologists: Max Wertheimer, Wolfgang Kohler, Kurt Koffka, and Kurt Lewin. Gestalt theorists see learning as goal-oriented with the learner being creatively bent.

Researchers in artificial intelligence using Gestalt theories as their guide generally analyze techniques human subjects employ and then incorporate them into a program. Generally this type of research is called Cognitive Simulation.

Cognitive Simulation (Gestalt) in artificial intelligence research was marked by early success. Unfortunately, the successes diminished quite rapidly until researchers became disenchanted with this approach.

Feedback in the Learning Process. Learning is necessarily a goal-seeking process and feedback is inherent in it. In practice, feedback is the process of regulating a procedure or system by returning information gained from its outputs to its inputs (Fig. 9). In order for a system to obtain feedback information, it must be able to develop associative patterns from which it can determine how to use the feedback information (Fig. 10). In other words, a system must be told something for it to learn something.

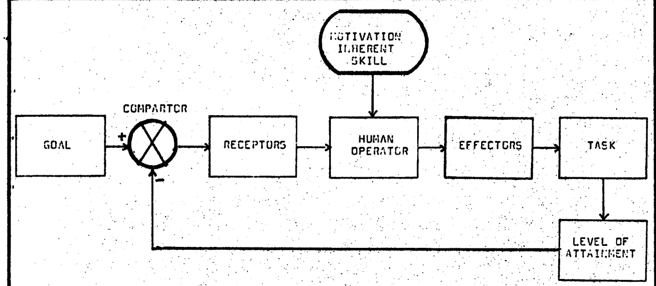


Fig. 6
FEEDBACK NATURE OF

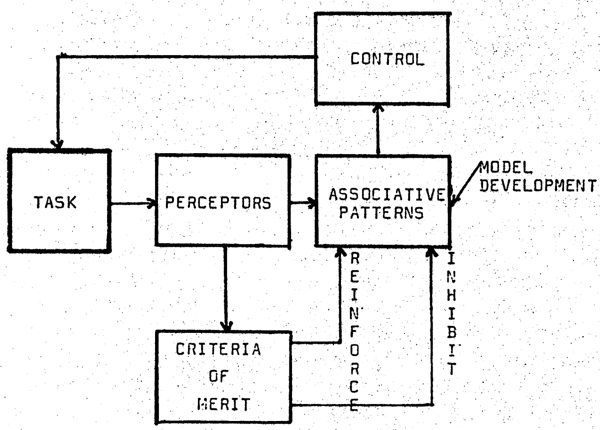


Fig.7
DEVELOPMENT OF ASSOCIATIVE PATTERNS (MODELS) IN LEARNING A TASK

IV. TYPES OF RESEARCH IN ARTIFICIAL INTELLIGENCE

There are many subjects that the artificial intelligence researcher has used as a vehicle toward understanding this complex subject. In addition to other areas, language translation and games are two of the broadest.

Language Translation

It was hoped in the early days of research in language translation that programs could be written that would act as automatic translators and interpreters of natural languages. These programs would perform the same type of functions as interpreters at the United Nations. It was also hoped that programs could be written for retrieving natural language text from a computer.

It seemed likely that encoding a relatively few basic words in a language translation program could be used as a 'bootstrap' toward the program's associating of new words (Fig. 8). But, after much work using this idea, researchers conclude that this is too difficult for our present stages of knowledge. Machine translation of typed scientific texts, let alone spoken language, is beyond our reach for the present.

Some researchers suggest that we do not understand our own language process well enough to replicate it with the computer.

Games

Programming computers to play games has several purposes. First, games resemble many real problems. Second, in games, the problem is well defined, thus easier to work with. The third reason is that doing so may lend solutions for the solving of real problems. And finally, it's

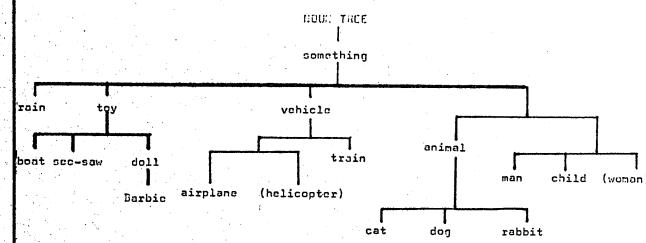


Fig. 8
A WORD ASSOCIATION MODEL

fun. Young animals have played games for eons in order to prepare for the real business of living. Game-playing on computers has a similar purpose for the computer scientist.

An important part of many game-playing programs is a procedure for searching a 'tree' of logical possibilities. Indeed, this is essential for any problem solving to take place. The problem of trees is that the total number of alternatives available to us are far too many to be exhaustively searched. In order to guarantee a perfect first move in the game of checkers, for example, the program would have to decide its move on the basis of about 10^{40} possibilities. In the game of chess, this figure would be about 10^{120} possibilities. Therefore, it is easy to see that a method must be used to 'prune' this gametree, and thus cut down on the number of alternatives to be considered.

One of the greatest achievements in the field of artificial intelligence has been a computer program to play draughts written by Dr. Arthur Samuel in 1967. This program takes about one minute to make its move and consistently beats Dr. Samuels, who is a good checker player himself. Although the World's Champion Checker Player has beaten the program four out of four games, the program beat the champion of Connecticut once.

Even though the game of checkers has yielded somewhat to researchers, the game of chess has not. Computer programs to play chess are still not proficient at the game, despite predictions that a computer would be the world's champion chess player by 1967.

Other Research

John G. Chubbock, A. George Carlton, and others at the Applied Physics Laboratory of Johns Hopkins University have developed a device they call The Beast. The Beast is a battery-operated cylinder on wheels which roams the halls and plugs itself into an electrical outlet whenever its batteries become depleted. It has been known to 'survive' for as long as 40 hours without running down its batteries. Research in this area could be extremely useful in learning how to construct robots to be used in hostile environments, or to automatically chauffeur automobiles. Another use could be in an unmanned space ship going to Mars. Immediate computer response to unexpected problems would be essential since communications between Earth and Mars would take a couple of minutes.

Another area of artificial intelligence research has been with computer programs called General Problem Solvers. Variants of this type of program attempt to simulate human behavior. Tests between a General Problem Solver and a college student have shown amazing similarities between thought processes of the student and the General Problem Solver. Although this area has been abandoned as a viable approach to artificial intelligence, it has served to broaden the researcher's outlook as to the problems of mechanizing thought processes.

V. STATE-OF-THE-ART

Research in the field of artificial intelligence has yielded much fruit since its infancy only twenty-five years ago. But the fruit it has borne is nothing compared to what it must bear in the future if it is to realize the predictions of its adherents.

Accomplishments

Artificial intelligence is a regenerative science; that is, accomplishments of today rapidly build those of tomorrow. This can probably be said about any field of endeavor, but nowhere is it more true than here. The reason being that, as we progress, the computer becomes stronger and more capable of aiding the researcher in his quest.

Research in artificial intelligence has given various byproducts such as assembly programs, debugging programs, test editing programs, and even a good mechanical arm developed at the Massachusetts Institute of Technology. There have been programs written that compose music and some that find chemical structures. In addition, there have been extremely complex programs written which handle such diverse tasks as traffic control, aiding architectural design, aiding electronic circuit design, monitoring patients in a hospital, and simulating chemical reactions.

By definition, artificial intelligence must have the ability to adapt to its environment and react to totally unforeseen circumstances. Although much has been accomplished, almost all computer applications fail to qualify as artificial intelligence under this definition.

Goals

The basic problem in artificial intelligence, as yet unresolved, is that all alternatives must be made explicit. Ways must be found for programs to determine useful information out of collections of seemingly random bits of information.

Computer programs must have shortcuts, similar in nature to those used by the human brain, built into them. In addition, they must be able to deal with facts about objects, relations between objects, and facts about facts. Also, all behaviors must be representable by the program, and it must be able to have or evolve concepts of partial success.

Furthermore, creative purpose is absent in the computer. In the words of Donald G. Fink: "At best, today's computers can only assist man in creative work." If artificial intelligence is to become an actuality, then one of the highest goals of the researcher must be to find ways to give machines creative purpose.

Computers are not constrained by a sociological environment; their only limitations are imposed by the programs given to them. Therefore, we must be sure that the 'top goal' given to a highly intelligent machine be the welfare of humanity, and not some private goal of the machine. As David Kendall said in the preface of a 1967 book of collected papers on machine intelligence: "Thus a study of 'Machine Intelligence' leads after yet one more remove to the ancient imperative, 'know thyself' and the universal problem of coexistence."

Predictions

There have been many predictions made in regard to the progress of artificial intelligence research. Among these is the 1965 prediction by Simon that: "... machines will be capable, within twenty years, of doing any work that a man can do." On the other side of the coin is the statement by P. E. Greenwood: "From the brief summary of the state of the art of artificial intelligence, one would conclude that little progress has been made since about 1960 and the prospects for the near future are not bright." Whichever side one decides to take, there are ample adherents to each.

Marvin Minsky sums it up nicely:

Once we have devised programs with a genuine capacity for self-improvement a rapid evolutionary process will begin. As the machine improves both itself and its model of itself, we shall begin to see all the phenomena associated with the terms 'consciousness,' 'intuition,' and 'intelligence' itself. It is hard to say how close we are to this threshold, but once it is crossed the world will not be the same.

VI. CONCLUSION

Artificial intelligence is a young science. Only time and the persistence of artificial intelligence researchers will reveal the full measure of its potential.

The late A.M. Turing laid down a test for adaptive intelligence. The test, which has come to be known widely as the Turing Test, is very simple. It consists of a man (the examiner) trying to discern whether or not the responses to questions he has proposed are being answered by a computer or another man. The test is conducted in such a way that the questioner can not physically determine which respondent has produced the response to his question; he must make the determination solely from the answer itself. So far, no one has been able to write a program that can consistently fool the examiner in the Turing Test. Until this is done, true artificial intelligence will remain within the realm of science fiction.

BIBLIOGRAPHY

Arnold, Robert, Harold C. Hill and Aylmer V. Nichols. *Modern Data Processing.* New York, New York: Wiley, 1969.

Bobrow, Daniel G. "Natural Language Input for a Computer Problem Solving System." *In Semantic Information Processing*, edited by Marvin Minsky. Cambridge, Massachusetts: M.I.T. Press, 1968, 146-226.

Cannara, Alexander B. "Toward a Human Computer Language." *Creative Computing, 1* (September-October, 1975), 38-40.

Chu, Kenneth C. "Applications of Artificial Intelligence to Chemistry." *Analytical Chemistry, 46* (August, 1974), 1181-1187.

Collins, N.L., and Donald Michie (Editors). *Machine Intelligence 1.* New York, New York: American Elsevier, 1967.

Dale, Ella, and Donald Michie (Editors). *Machine Intelligence 2.* New York, New York: American Elsevier, 1968.

Dreyfus, Hubert L. *What Computers Can't Do—A Critique of Artificial Reason.* New York, New York: Harper and Row, 1972.

Feigenbaum, Edward A. and Julian Feldman (Editors). *Computers and Thought.* New York, New York: McGraw-Hill, 1963.

Fink, Donald G. *Computers and the Human Mind.* Garden City, New York: Doubleday, 1966.

Good, I.J. "A Five Year Plan for Automatic Chess." In *Machine Intelligence 2*, edited by Ella Dale and Donald Michie. New York, New York: American Elsevier, 1968, 89-118.

Hamming, Richard W. *Computers and Society.* New York, New York: McGraw-Hill, 1972.

Kiss, G.R. "Networks as Models of Word Storage." In *Machine Intelligence 1*, edited by N.L. Collins and Donald Michie. New York, New York: American Elsevier, 1967, 155-167.

McCarthy, John. "Programs with Common Sense." In *Semantic Information Processing*, edited by Marvin Minsky. Cambridge, Massachusetts: M.I.T. Press, 1968, 403-418.

Minsky, Marvin (Editor). *Semantic Information Processing.* Cambridge, Massachusetts: M.I.T. Press, 1968.

Newell, Allen and H.A. Simon. "GPS, a Program That Simulates Human Behavior." In *Computers and Thought*, edited by Edward A. Feigenbaum and Julian Feldman. New York, New York: McGraw-Hill, 1963, 279-293.

Porter, Arthur. *Cybernetics Simplified.* New York, New York: Barnes and Noble, 1969.

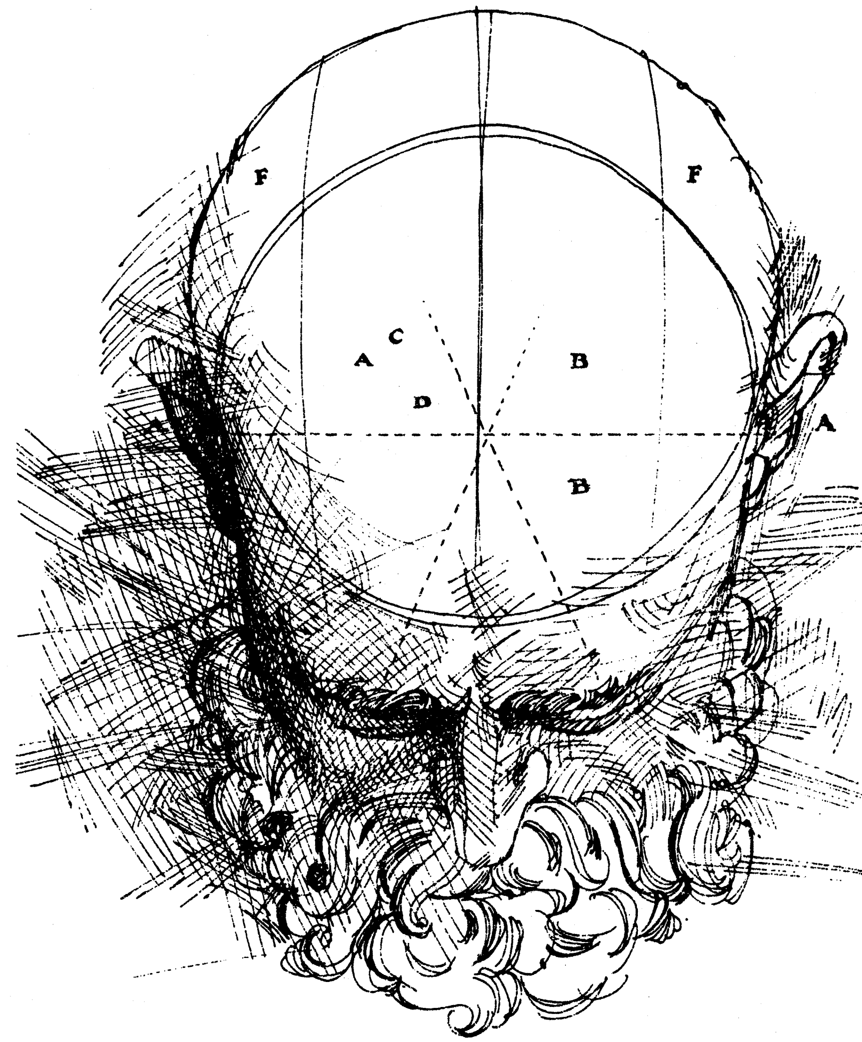
Quillan, Ross M. "Semantic Memory." In *Semantic Information Processing*, edited by Marvin Minsky. Cambridge, Massachusetts: M.I.T. Press, 1968, 227-270.

Slagel, James R. *Artificial Intelligence—The Heuristic Programming Approach.* New York, New York: McGraw-Hill, 1971.

Streeter, Donald N. *The Scientific Process and the Computer.* New York, New York: Wiley, 1974.

Wiest, Jerome D. "Heuristic Programs for Decision Making." *Harvard Business Review, 44* (September-October, 1966), 129-142.

Can Computers Think?



by David H. Ahl

"A Bird is an instrument working according to mathematical law, which instrument it is within the capacity of man to reproduce with all its movements."

Leonardo da Vinci (1452-1519)

Intelligent Machines and Today's Digital Computer
A common attitude toward today's computers is that such machines are strictly arithmetic devices. While it is true that machines were first built to carry out repetitive arithmetic operations, they are capable of other, nonnumeric tasks. The essence of the computer is the manipulation of symbols—it is only a historical accident that the first application involved numeric symbols. This incorrect notion of the computer as a strictly numeric device results in the inability of many to conceive of the computer as a device exhibiting intelligent behavior, since this would require that the process be reduced to a numerical one. The reaction of many people to statements about intelligent behavior by machines seems to indicate that they take such statements to imply complete functional equivalence between the machine and the human brain. Since this complete functional equivalence does not exist, such people believe they have thereby debunked intelligent machines. Their conclusion is incorrect because this equivalence was never implied. Intelligent behavior on the part of a machine no more implies complete functional equivalence between machine and brain than flying by an airplane implies complete functional equivalence between plane and bird.

The concept of comparing the behavior of men and machines in an n-dimensional continuum recognizes differences as well as similarities. For example, a common argument against machine intelligence is that the brain is a living thing—the machine is not. In our continuum we simply recognize the dimension of living and note that machines and men occupy different positions on this dimension.

Is It Possible for Computers to Think?
No—if one defines thinking as an activity peculiarly and exclusively human. Any such behavior in machines, therefore, would have to be called thinking-like behavior.

No—if one postulates that there is something in the essence of thinking which is inscrutable, mysterious, mystical.

Yes—if one admits that the question is to be answered by experiment and observation, comparing the behavior of the computer with that behavior of human beings to which the term "thinking" is generally applied.

The two negative views are unscientifically dogmatic. The positive, or empirical, view is supported by scientists who point out that there exists a continuum of intelligent behavior and that the question of how far we can push machines out along that continuum is to be answered by research, not dogma. One might add a further qualification: to assert that thinking machines are possible is not necessarily to assert that thinking machines with human capabilities already exist (or even that they will exist in the near future).

What, then, is the goal of artificial intelligence research? It seems to be this: to construct computer programs which exhibit behavior that we call "intelligent behavior" when we observe it in human beings. Because this research area is still in the formative stage, many different research paths are being explored and a wide diversity of results have been produced.

But Doesn't a Computer Do Exactly What It Is Told To Do and No More?

Commenting on this familiar question, a well-known researcher in the field had this to say:

This statement—that computers can do only what they are programmed to do—is intuitively obvious, indubitably true, and supports none of the implications that are commonly drawn from it.

A human being can think, learn, and create because the program his biological endowment gives him, together with the changes in that program produced by interaction with his environment after birth, enables him to think, learn, and create. If a computer thinks, learns, and creates, it will be by virtue of a program that endows it with these capacities. Clearly this will not be a program—any more than the human's is—that calls for highly stereotyped and repetitive behavior independent of the stimuli coming from the environment and the task to be completed. It will be a program that makes the system's behavior highly conditional on the task environment—on the task goals and on the clues extracted from the environment that indicate whether progress is being made toward those goals. It will be a program that analyzes, by some means, its own performance, diagnoses its failures, and makes changes which enhance its future effectiveness.

Similarly, it is wrong to conclude that a computer can exhibit behavior no more intelligent than its human programmer and that this astute gentleman can accurately predict the behavior of his program. These conclusions ignore the enormous complexity of information processing possible in problem-solving and learning machines. They presume that, because the programmer can write down (as programs) general prescriptions for adaptive behavior in such mechanisms, he can comprehend the remote consequences of these mechanisms after the execution of millions of information processing operations and the interaction of these mechanisms with a task environment. And, more importantly, they presume that he can perform the same complex information processing operations equally well with the device within his skull.

Conclusion

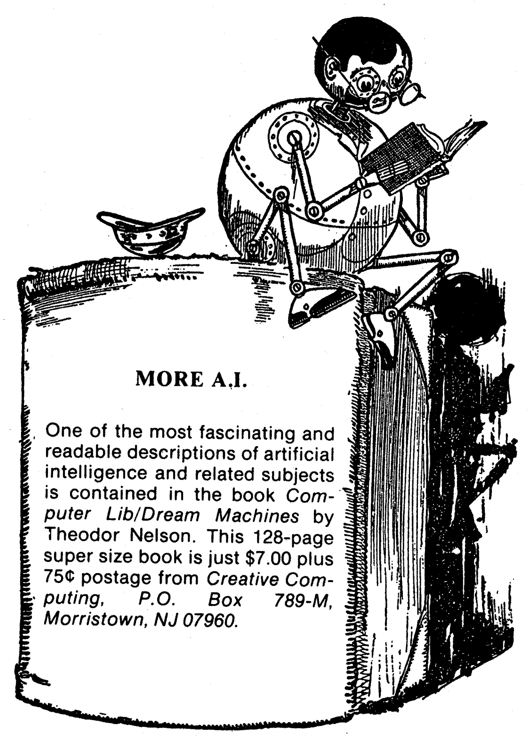
Thinking of the quotation of da Vinci's at the start of this article, one might paraphrase it as follows:

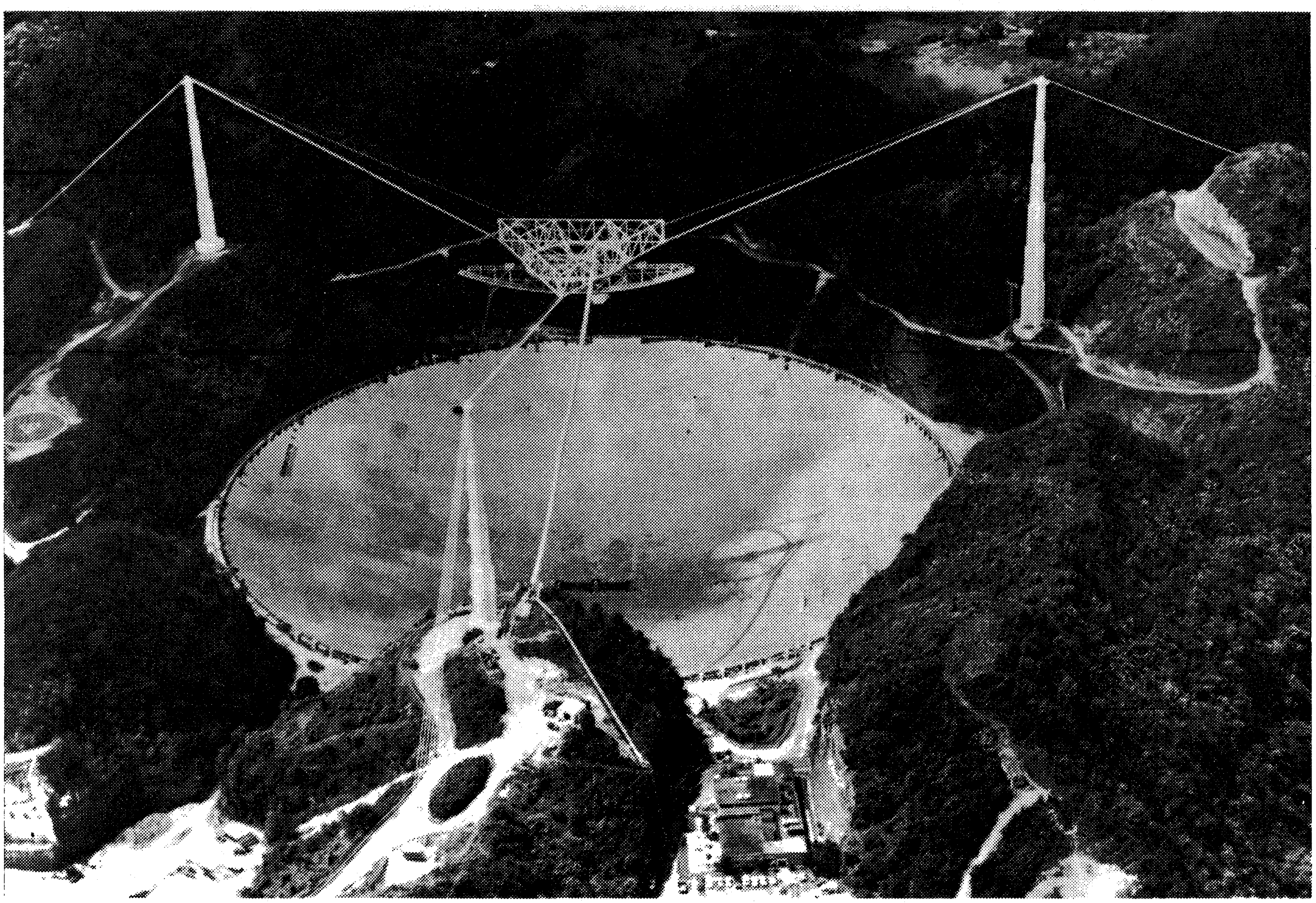
When men understand the natural laws which govern the flight of a bird, man will be able to build a flying machine.

While it is true that man wasted a good deal of time and effort trying to build a flying machine that flapped its wings like a bird, the important point is that it was the understanding of the law of aerodynamic lift (even though the understanding was quite imperfect at first) over an airfoil which enabled men to build flying machines. A bird isn't sustained in the air by the hand of God—natural laws govern its flight. If man gained an understanding of the processes of aerodynamics, may he not also obtain an understanding of the information processes of the human brain?

And then, once these processes are understood, there is no reason why man won't be able to duplicate in a computer the powerful facilities of association, recognition, and indeed, thinking of the human brain.

This article was adapted from the introduction to Computers and Thought, *edited by E. A. Feigenbaum and Julian Feldman, and an article from the same volume entitled "Attitudes Toward Intelligent Machines" by Paul Armer.*





An Ear On The Universe

by John Lees
University of Missouri—Rolla

Nestled in a natural limestone sinkhole in the jungle covered mountains of northern Puerto Rico is the largest radio telescope on Earth. Its reflector is a spherical bowl 1,000 feet in diameter — a surface area of 20 acres. The telescope is steered by repositioning the 600 tons of receiving and transmitting equipment, which is supported 50 stories in the air on cables anchored by three massive reinforced concrete towers. Each of the towers is guyed to ground anchors with five 3.25 inch steel bridge cables. When the Arecibo reflector is used as a 2380 MHz S-band radar transmitter its effective power output is 100 trillion watts. When used as a receiver for the radar echoes its sensitivity is one-100 million trillionth of a watt; a span of 34 orders of magnitude.

The Arecibo Observatory in Puerto Rico is part of the National Astronomy and Ionosphere Center operated by Cornell University. Constructed in 1963 at a cost of $9 million, the telescope's reflector originally consisted of a bowl of wire mesh supported by cables slung from the edge of the reflector bowl. It was not thought that the receiver-transmitter platform would be stable enough in normal winds and temperature changes to make it worthwhile to have an extremely accurate reflector bowl, since the aerial platform was expected to sway as much as one and one-half inches.

When in 1966 Puerto Rico lay in the path of Hurricane Inez these original expectations proved to be far too conservative. In the sixty-two mile-per-hour winds of the hurricane the platform was observed to sway less than half an inch. This meant that, under normal weather conditions, the platform could be expected to sway less than three-tenths of an inch. Because of these findings it was decided to upgrade the accuracy of the reflector curvature to match the stability of the receiver-transmitter platform.

With the backing of the National Science Foundation and the National Aeronautics and Space Administration this task was begun. An additional $8.8 million was spent to improve the accuracy of the reflector, improve the receiver-transmitter platform, and increase the power of the radar transmitter. The reflector surface now consists of 38,778 specially designed aluminum panels, each of which has on it a white square used as a target for the laser surveying system. This laser system permits the entire surface of the reflector to be surveyed to an accuracy of better than one millimeter. The panels are individually adjustable to keep the reflector as spherical as possible. The reconstruction was completed in November of 1974.

The History Of Radio Astronomy

Radio waves were experimentally demonstrated in 1888 by Heinrich Hertz and a few unsuccessful attempts were made shortly thereafter to detect radio waves from the Sun. Early receiving equipment was of low sensitivity and the effect of the upper atmosphere on radio waves was poorly understood or even completely unsuspected. Some progress was made in the 1930s, mostly as a byproduct of communications research or through the efforts of a few dedicated hobbyists, but the real progress did not begin until World War II brought the use of radar and military communications systems.

During the war several things combined to open the way for radio astronomy. A great deal of basic research was being performed by the scientists working on wartime radar installations. It was at this time that it was discovered that sunspots could interfere with radar operation. In England, problems were encountered with cosmic background noise and with upper atmosphere noise when systems were being developed for detecting the V2 rocket. Research in these vital areas was of course encouraged. The armed services also supported a good deal of spin-off research in such areas as the radar detection of meteorites and the reception of radar echoes from the Moon.

The universe outside of our little solar system abounds with mysteries and discoveries which stretch the mind's ability to comprehend.

The advances made during the war in electronics, radar, radio and all branches of technology had a vast impact on the resumption of scientific research in 1945. Many researchers had received new insights, formed new associations, or uncovered entirely new areas in which to do research. Sophisticated war surplus equipment was easy to come by and the armed services were quite willing to cooperate with the scientists who had been so much help during the war. Technology had been given its biggest push; it was not about to slow down.

Radio astronomical research blossomed in England and Australia and slowly spread to other countries. Jodrell Bank in England began with war surplus radar equipment. Research in Australia was the direct outgrowth of a group formed during the war. The U.S. Naval Research Laboratory initiated research in short wavelength radio astronomy, completing in 1950 the first large radio telescope designed specifically to operate at centimeter wavelengths. By the end of the 1940s most countries had established radio astronomy research groups. In the early 1950s several vital discoveries, such as detection of the 21 cm hydrogen line, laid a firm foundation for the expansion in research which turned into a boom with the coming of the space age. There are now at least as many major radio astronomical observatories as there are optical observatories. Computer techniques in pattern recognition and information theory have been of great help in radio astronomy and have made possible some of the newer synthesized radio telescope arrays.

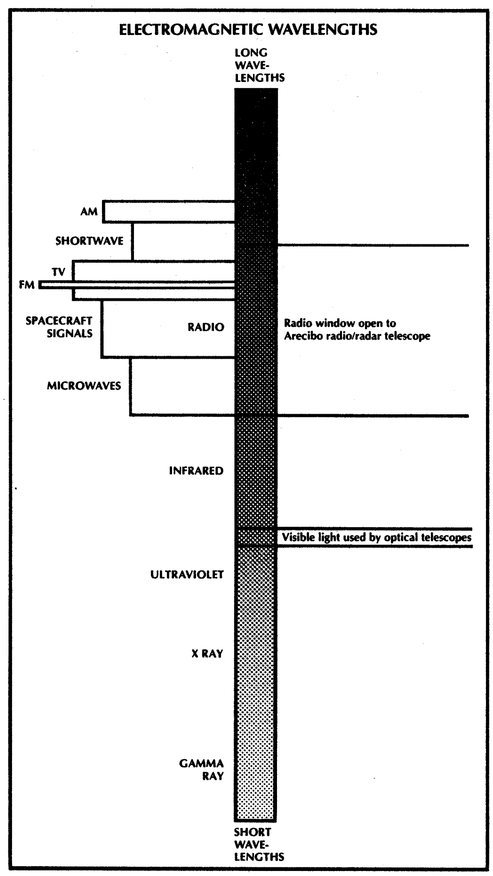
Discoveries In Radio Astronomy

A radio telescope differs from an optical telescope in the portion of the electromagnetic spectrum to which it is sensitive (see figure). Optical telescopes are used around the narrow band of frequencies to which the human eye is sensitive. Radio telescopes, however, are sensitive to a much broader range of frequencies. A radio telescope can be used day or night and is not bothered by an overcast sky or smog. Radio telescopes can also see into parts of the Universe which are blocked to optical telescopes by interstellar dust clouds. Some radio telescopes, such as the one at Arecibo, are also powerful radar transmitters and can bounce radio waves off satellites, asteroids, nearby planets, the Sun and particles in the Earth's atmosphere.

To a radio telescope, the sky looks totally different than it does to the human eye on a dark night. The visual brightness of a star in the night sky has little to do with its radio brightness. Familiar stars no longer look the same. The constellations are gone. The planets, themselves not strong emitters of radio waves, are almost invisible. The Milky Way is much brighter at radio frequencies and there are a multitude of new objects to be observed in the sky. But cosmic radio sources are still very faint. It is said that celestial radio signals reaching Earth are so faint that all the energy collected in the forty-year history of radio astronomy is about equal to that released when a few snowflakes fall on the ground.

Since it began operation, the Arecibo Observatory has yielded an astonishing amount of information about our solar system, our planet and our universe. The Arecibo telescope has heard more that 3,000 separate radio sources, only about 100 of which have been identified optically. Many of them will never be optically identified from Earth; they are simply too weak in the optical portion of the spectrum. Some of the radio waves received at Arecibo have been travelling nearly ten billion years on their way from the edge of space to Earth.

Radio maps of the Moon produced before the lunar landing missions predicted that the Moon's surface would be covered with a thick layer of dust. The close similarity between radar and optical maps of the Moon indicate that radio reflectivity and light reflectivity are probably close enough to allow accurate mapping of the planets. Extensive mapping of the surfaces of Venus, Mercury and



Celestial radio signals reaching Earth are so faint that all the energy collected in the forty-year history of radio astronomy is about equal to that released when a few snow-flakes fall on the ground.

Mars is planned, along with looks at the asteroid belt, the four major satellites of Jupiter and the rings of Saturn. It is also possible to look "under the surface" with radar, showing the terrain which lies below a layer of dust.

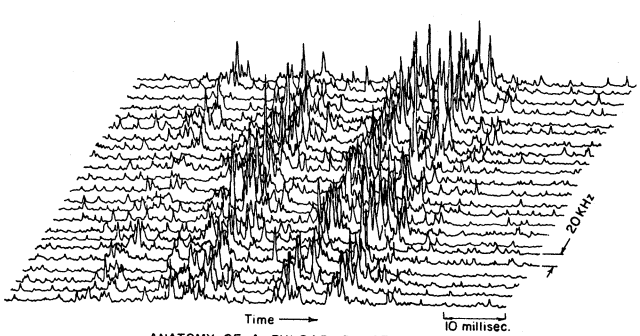
Venus, Earth's sister planet with the eternal cloud cover, has been a source of mystery and the target of sometimes wild conjecture for many years. In 1964, radar signals from Arecibo were used to accurately determine Venus' period of rotation and confirmed the theory that Venus, alone of all the planets, spins on its axis in a clockwise, not counter-clockwise, direction. It was later also determined that Venus exhibits the phenomenon of "Earth lock." Each time Venus swings by Earth, it turns the same face toward Earth. In 1968, Arecibo produced its first radar map of Venus.

Mercury, the closest planet to the Sun, completes an orbit every eighty-eight days. Until 1965 it was thought that Mercury always kept the same face turned toward the Sun. Radar studies by astronomers at Arecibo showed that Mercury did, after all, rotate slowly on its axis, turning alternate faces to the Sun at each close approach. Beginning in 1970, the Arecibo radar has been used to map portions of Mercury, showing the planet's surface to be rougher than that of Venus but not quite as rough as the Moon.

The universe outside of our little solar system abounds with mysteries and discoveries which stretch the mind's ability to comprehend. In 1967 a British survey of twinkling radio sources turned up radio pulses of startling regularity coming from a certain direction in the sky. The Cambridge, England team which made the discovery did not release the news of the discoverey immediately because of debate over whether the pulses might be signals from intelligent beings elsewhere in the Galaxy. After several weeks this possibility was largely discounted and the theory advanced that the pulses were coming from a rapidly spinning neutron star.

Conclusive evidence against the possibility of pulsar signals being intelligent interstellar communication came from the Arecibo discovery that the pulse interval was increasing by some 36 billionths of a second a day. This was confirmation of the theory that pulsars are neutron stars, since such a rotating star should be gradually slowing down.

Neutron stars, theoretically predicted in 1933, are the remnants of giant stars which have collapsed to a radius of about 10 km while retaining approximately the mass of our sun. This results in a density of some 10^{14} g/m^3, the density of an atomic nucleus. (For an exercise in the mind's inability to comprehend large numbers, try to imagine one cubic inch of matter with a weight in excess of ten billion tons! Of course such a density can only exist under the terrific pressures found within a collapsed star.) In some way which is not yet completely understood, the intense magnetic field of the rotating neutron star generates beams of coherent radio waves and light which appear as pulses to an observer on Earth.



ANATOMY OF A PULSAR PULSE

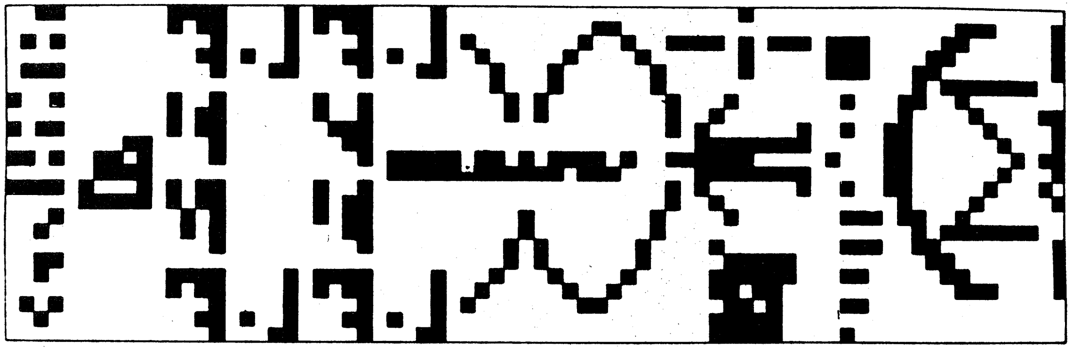
Perhaps the most famous pulsar is the one in the Crab Nebula, which was observed and recorded by the Chinese as an explosion in the year 1054 A.D. A nebula is a still-glowing cloud of interstellar gas and dust, the remnant of a supernova, or stellar explosion. The Crab Nebula pulsar also emits light pulses and has been optically identified as the exploded supernova.

The Arecibo Observatory continues to do research on pulsars and is also participating in the search for black holes. Without a doubt one of the oddest celestial bodies, a black hole is the super-dense remnant of a giant star which has collapsed in such a way that it almost no longer exists. Past a certain limit the gravitational field of a black hole does not permit any interaction with the rest of the universe. A beam of light, or anything else, will "fall into" a black hole and never come out. Black holes are unescapably predicted by the general theory of relativity, but their existence has not yet been observationally verified. Of course one must use indirect methods to "observe" a black hole, such as noting the apparent influence of a large mass on a stellar system when no such mass is observable with an optical or radio telescope. Interestingly enough, something like a black hole was predicted in 1795 by Pierre-Simon Laplace.

Of all the things which may exist outside the bounds of our planet Earth, surely the most wondrous of these is life itself.

In addition to pulsars and black holes, there are a multitude of other interesting objects Out There — several other types of radio stars, several types of radio galaxies and, the most distant known objects in the Universe, the quasars, or quasi-stellar radio sources. Receding from us at more than half the speed of light, quasars are whole galaxies in which a very small part (only light-weeks in diameter) releases tremendous amounts of energy equivalent to the total annihilation of millions of stars. Quasars emit enormous quantities of radio energy which, traveling at the speed of light, have taken as long as ten billion years to reach Earth.

Although the 1,000 foot dish of the Arecibo radio telescope is physically the largest on Earth and there are now several radio astronomical observatories using multiple antennas with their signals combined by computer in such a way as to synthesize antennas of more than a kilometer in diameter, there are very distant or very small objects (such as quasars) which can not be adequately measured with a single radio telescope. To make such measurements a technique known as long-baseline interferometry is used. This involves combining the signals from two or more radio telescopes, often on opposite sides of the Earth, and using computers to process the signals to yield data not obtainable with a single radio telescope. Pioneer work in this area was done in 1966 by a team which included a Cornell professor and made use of the Arecibo Observatory.



This is Man on Earth Speaking to Space, the message beamed from Arecibo Observatory to M-13.

Life In The Universe

The Universe is a rather large place, large enough to contain all the wonders imaginable and quite a few which we have not approached in even our wildest dreams. Of all the things which may exist outside the bounds of our planet Earth, surely the most wondrous of these is life itself. The search for extraterrestrial life is an exciting and important part of radio astronomy.

Scattered through interstellar space, between the stars and dust clouds, are isolated molecules of materials such as hydrogen, formaldehyde and methyl alcohol — some of the basic ingredients of life on Earth. The Arecibo radio telescope can be used to gather data for analyzing and quantifying these molecules, as well as to search for other freely floating chemicals. These chemicals may very well be the seeds from which life on Earth evolved. Important confirmation of this theory may come when the *Viking* landers conduct the first rigorous search for life on Mars in the summer of 1976.

If life has evolved on the planet Earth, in our solar system, why may it not have evolved elsewhere in the Universe? There are some 200 billion stars in our Milky Way Galaxy alone. It is now fairly certain that a number of stars, at least in our part of the Galaxy, have planets of about the mass of Jupiter. Present methods can not detect less massive planets in orbit around other stars, but it is generally accepted that stars with planetary systems are not exceedingly rare. Surely, on some of the other planets in our galaxy, the correct conditions have obtained for life of some sort to begin its slow way along the evolutionary process. Among 200 billion stars, odds of even a million to one begin to look rather plausible.

The most exciting possibility of all is that there may not simply be life elsewhere in the Universe, but that it may be intelligent life. There are billions of stars in our galaxy and there are billions of galaxies in our universe. It is not so difficult to believe that intelligent beings inhabit more than one planet in this vast universe. Some of these beings have probably reached the same level of understanding of natural phenomena as have human beings; some are more, and some are less, advanced. With all of them we must feel the most basic kinship and a yearning to know for certain that we are not alone in the face of the vastness of the Universe.

M-13 Response Received

Less than 10 days after a formal announcement of life on earth was beamed toward a far-off cluster of stars known as M-13 from the Arecibo Observatory in Puerto Rico, an "answer," purported to be from outer space, was received.

Cornell professor and director of the National Astronomy and Ionosphere Center (NAIC) Frank D. Drake who, with his staff, initiated and composed the "life on earth" message, received this answer by telegram Nov. 25:

"Message received. Help is on the way. —M-13." It came through on the NAIC telex machine in Ithaca via the International Telephone and Telegraph system. The true identity of the sender has not been confirmed, but Drake suspects that it may have come from practical jokers on the observatory staff at Arecibo.

Listening For Intelligence

It is indeed tantalizing to think that, right now, like an inaudible whisper, radio messages from light years away are falling into the valley of the Arecibo reflector bowl — messages that could be heard if their direction and frequency were known. When the upgraded Arecibo radio telescope was dedicated on November 16, 1974, a message was sent commemorating the occasion. Our first intentional attempt at radio communication with extraterrestrial life is now travelling at the speed of light through the Milky Way toward a globular cluster of some 300,000 stars known as M-13. It will take about 25,000 years to reach its destination. Any reply will take as long to return to Earth. Although the message was beamed from Arecibo for only three minutes it is entirely possible that one day a reply will be received. (see figure)

In the meantime, it is within Earth's technological ability to decide the question of whether there are other beings Out There trying to communicate with each other and with us. No more stupendous moment in the history of the Earth can be imagined than the first intellectual interchange with an intelligence other than our own. The Arecibo radio telescope is now the premier instrument in the world for such an undertaking. Given careful planning and an adequate observing program, there is a genuine probability that this most important of frontiers will be crossed for the first time.

Thanks to Cornell University and the National Astronomy and Ionosphere Center for permission to use illustrations and to make extensive use of published material on the Arecibo Observatory.

Annotated Bibliography

Hey, J. S. *The Evolution Of Radio Astronomy.* New York: Science History Publications, 1973. A semi-technical account of the history of radio astronomy, its instruments and its accomplishments.

Misner, C. W., K. S. Thorne and J. A. Wheeler. *GRAVITATION.* San Francisco: W. H. Freeman and Co., 1973. A massive book on an utterly fascinating topic. Takes at least calculus through simple partial-differential equations to really understand, but it is a mind stretching book to browse. Just reading the chapter headings and subheadings is a trip in itself.

Sagan, Carl. *The Cosmic Connection.* New York: Dell Publishing Co., 1975. An unrestrained intellectual romp through the concept of extraterrestrial life.

Sagan, Carl and Frank Drake. "The Search For Extraterrestrial Intelligence." *Scientific American.* Vo. 232, No. 5, pp. 80-89, May, 1975. A current appraisal of the subject. Contains the Arecibo message and its translation.

Strom, Richard G., et al. "Giant Radio Galaxies." *Scientific American.* Vol. 233, No. 2, pp. 26-35, August, 1975. Research into the largest known objects in the Universe.

Sullivan, Walter. *We Are Not Alone.* New York: Signet Books, 1966. One of the first popular books on the subject of extraterrestrial life. Includes an account of Project Ozma and Frank Drake's first proposed interstellar message.

COMMUNICATION ACROSS THE UNIVERSE

by Martin Harwit
Space Sciences Department
Cornell University

In the past few years a number of messages have gone out from Earth in the hope that an advanced civilization might find them. We also have searched cosmic sources of radiation for signs of extraterrestrial intelligence but have not yet found any.

We are faced with two questions:
What are the best ways for us to transmit messages?
What are the best strategies to employ in searching for messages?

The answers are not simple.

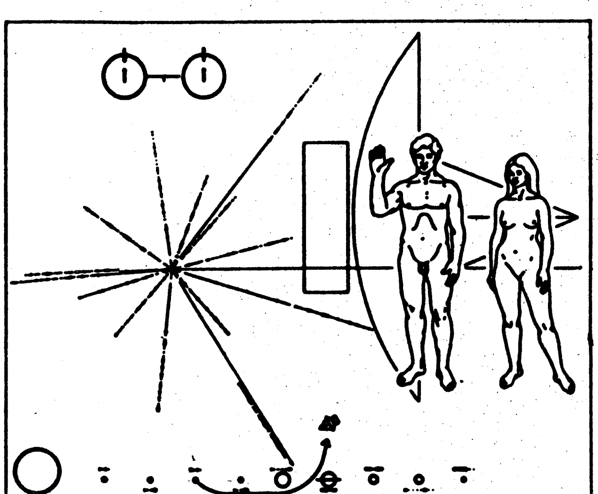
In writing an ordinary postcard, we have to consider three different factors. We have to decide to who to write, what to write and how to convey the card. These same decisions — not necessarily in the same order — are also involved in cosmic communication.

First, we must decide on the best channel of communication. If we plan to transmit, should we deliver our message by interstellar rocket, should we employ a modulated laser beam flashing coded signals, or is there some method far superior to either of these?

Once we decide how the message is to be conveyed, we must decide on the language to be used. Surely it won't be English, or Chinese, or even Fortran. The intended recipients won't know English or any other terrestrial tongues. How can we establish a common language understandable to all?

Lastly we have to decide on an address: If we knew how to choose a civilization that is alive and active today, would it still be around tomorrow? In fact, what is the life expectancy of a technically advanced civilization? Judging by recent Earth history, the step from the development of elementary radio transmission techniques to an ultimate nuclear explosive extinction of Man may last just one short century. If that is common, we had best make our message quite short and to the point. And we should forego asking for a recognition sign; we may not be here to receive it.

* * * * * * * * * *



Let us first look at the optimum channel for communication. Basically this is dictated by the structure and contents of the universe we inhabit. If the universe were different, our options also would differ. Consider, for example, the messages carried by the spacecraft Pioneer 10 and 11. Each of these vehicles carries a plaque that describes the location of our planet in the Milky Way galaxy, and describes the inhabitants of Earth.

What are the chances that these two plaques will ever be discovered? I think the odds are very low. There is not so much doubt that a sophisticated civilization would have the ability to detect the spacecraft. Rather there is uncertainty about the ability to recognize them:

We know that roughly once a year the solar system ejects a comet nucleus about a mile in diameter. For each of

In the past billion years alone, a billion mile-sized comet nuclei and countless smaller chunks have left our solar system. In this confusion of debris, how is any civilization going to pick out two tiny Pioneer spacecraft as having especial significance?

these large chunks of matter there are probably thousands, and perhaps millions, of smaller fragments that also find their way out of the solar system. In the past billion years alone, a billion mile-sized comet nuclei and countless smaller chunks have left our solar system. In this confusion of debris, how is any civilization going to pick out two tiny Pioneer spacecraft as having especial significance? Certainly, success will require a formidable effort. The Pioneer 10 and 11 probes are therefore basically addressed only to the most sophisticated and ambitious civilizations. That happens to be the best we can hope to do right now, but as our technology develops we should be able to do far better.

Similar questions of confusion arise when we think of communicating by radio or by means of visible lasers. The prime constraint is seldom the sensitivity of the detector. Instead, we are mainly limited by the profusion of signals emitted from natural astronomical sources. The detection of a message from another civilization requires means of discriminating against all this noise.

If we inhabited a quieter universe, in which there were far fewer bright astronomical sources, this problem would be less severe. On the other hand, perhaps there would also be correspondingly fewer intelligent civilizations in the universe.

The optimum means for transmitting signals also depends on the urgency of a message. Spacecraft are rather slow. At best they can approach the speed of light if we are willing to budget for the high expense of accelerating these craft. Even if we do, the destruction rate through collisions

One other carrier of information might exist in the form of the faster than light tachyons.

with naturally occurring interstellar debris increases dramatically with increasing speed, so it is not clear how well high speed spacecraft could survive prolonged interstellar voyages.

Electromagnetic signals, or signals employing gravitational waves and neutrinos all travel at the speed of light. At that speed a message can reach the nearest stars in a matter of years, the center of our Galaxy in thirty thousand years, and the nearest galaxies in some millions of years. Two way conversations would have to be correspondingly slow. Just now electromagnetic waves may look like the best bet because we do not yet know how to receive or transmit gravitational waves, and the transmission and reception of neutrinos is discouragingly inefficient. But all that may change.

One other carrier of information might exist in the form of the faster than light tachyons. Tachyons are hypothetical particles which may very well not exist at all. We just don't yet know. However, if they do exist, they would be priority choice for urgent messages. They could be transmitted at such high speeds that two-way conversation might be practical in some instances. Tachyons, if they can be generated, would take advantage of a loophole in the laws of relativity. Einstein's special theory states that no particles can be accelerated up to or beyond the speed of light. However, it does not state that particles can't be generated directly at these high velocities. Just as two colliding photons can give rise to a pair of particles such as an electron and a positron or a proton and an antiproton, both travelling at velocities well below the speed of light, so also we can imagine two photons colliding and giving rise to a pair of tachyons. We do not yet know how to detect these entities, but as long as we remain uncertain about their existence, we will be ignorant of what might well be the most promising channel for interstellar and intergalactic messages.

* * * * * * * * * *

We now turn to the language to be used. Two ideas have been widely discussed in this context. The first involves pictograms — simple pictures that might be easily understood. The second concerns artificial languages that are self explanatory.

This second approach is particularly interesting. The most extensively developed language of this kind is *Lincos* developed by the mathematician Hans Freduenthal at the University of Utrecht. In essence, he starts out with a list of symbols representing one dot, two dots, three dots, . = *1*, . . = 2, . . . = 3, and so on. He then lists examples of addition. Essentially

$$1 + 1 = 2$$
$$1 + 2 = 3$$
$$2 + 1 = 3$$
$$1 + 1 + 1 = 3$$

and so on.

This defines not only mathematical relations, but also the concept of equality that is a common part of our social thinking. Symbols like $<$ and $>$ are similarly defined by lists of numerical examples and eventually work their way into such concepts as greatness, prominence, and so on.

In principle such a language can go on to physics, for instance, by listing the relative masses of all atoms and nuclei known to be stable. Every advanced civilization will know the values of these masses which bear a nearly — but not quite — integer relationship.

And once the elements are defined a further listing can show elementary chemical reactions and structures including those that are fundamental to our existence — metabolic processes and genetic structure. Beyond this stage a message can become encyclopedic. For, it is really the fundamental structure of the language that is most difficult to establish. Once that is accomplished, more sophisticated concepts can readily be added in any quantity.

The problem of constructing intelligible pictograms is somewhat different. Such pictures consist of light and dark squares arrayed very much like a television picture. There are mn elements — where m and n both are prime numbers. With this choice of prime numbers there are only two ways that a rectangular array — picture — can be obtained: a picture with n rows and m columns, or one with m rows and n columns. If m = n the picture is square and unique except for a left-to-right, up-down ambiguity.

Proponents of pictograms often do include some elementary arithmetic and chemical concepts in their pictures, but they rapidly go on to show more sophisticated ideas, a picture of man or a sketch of the solar system.

Whether such pictures are intuitively obvious is not

The first reception of extraterrestrial messages no doubt will involve great difficulties and major technical advances. However, once this initial barrier is overcome a whole new social era could begin.

really clear. Our own intuition is so strongly conditioned that such messages are likely to be quite fundamentally prejudiced to the extent of being incomprehensible to an alien civilization.

Ultimately the pictograms should be subject to exactly the same need for logical development as any other language, and I imagine a properly designed interstellar message is likely to contain both a preliminary developmental chapter that defines the language, as well as a message that may be partly in the form of words, and partly illustrated by pictograms.

* * * * * * * * * *

The final question to be considered here is whom to address. Or whom to expect to transmit.

If we believe that life can only exist on planets, then we might first transmit messages to the nearest known planetary systems or search for messages emanating from these nearest neighbors. Such messages might consist of individual symbols (letters, numbers) each transmitted for a period lasting anywhere from seconds to hours.

On the other hand if, as Frank Drake has pointed out, some forms of life could exist on a neutron star where temperatures are far higher than on Earth, metabolic rates could be speeded up a million-fold. Such a civilization might then transmit messages at a rate of a million symbols a second. The problems of transmitting and receiving at these speeds would not necessarily be more difficult for us, but we would have to be aware of the great range of possibilities. Not only would we have to worry about the language to use, the contents to be transmitted and the means of transmitting, we even would have uncertainties about how "fast to talk."

The first reception of extraterrestrial messages no doubt will involve great difficulties and major technical advances. However, once this initial barrier is overcome a whole new social era could begin.

Isaac Asimov

The Cosmic Subway Line


ILLUSTRATION BY HELMUT WIMMER

Artist's conception of a "black hole" in space.

The most exciting phenomenon in astronomy these days is the black hole—an apparent final graveyard of matter, thanks to its gravitational field. There are only four kinds of forces known to exist in the universe, and gravity is by far the weakest of the four—but wait.

Two of the forces are very short-distance phenomena that involve only subatomic particles and aren't felt outside atomic nuclei, ordinarily. A third one, electromagnetism, is long-distance, but expresses itself as an attraction under some conditions and a repulsion under others. The two tend to cancel each other, so that electromagnetism never manages to display really great intensity.

Gravity is different; it shows itself as a long-distance phenomenon, and only as an attraction. The more matter you pile together in one place, the greater its gravitational field becomes. If you start with a certain amount of matter and squeeze it together more and more tightly, the stronger its gravitational field becomes. Either way (or in combination), a gravitational field can be made greater than any other force can possibly be.

As gravitation becomes extreme, all matter within its influence breaks down. Atoms and even subatomic particles squeeze down to nothing. Anything that falls into a sufficiently intense gravitational field can never come out at the point it entered, so that the field acts as a "hole." Even light can't emerge, so it is a "black hole."

A black hole can form when a large star explodes and collapses. Astronomers think that an object they call "Cyg X-1" is a large black hole in our own galaxy. It may be that there are black holes of all sizes distributed all over the universe. Even "mini-black holes," perhaps of no more than pinpoint size, may have formed in the great explosion that produced our universe in the first place.

It would seem that all matter will eventually fall into one black hole or another, until only black holes are left, and that this would represent the final end of the universe. Such an end is many billions, perhaps trillions, of years in the future, however, and meanwhile black holes could, conceivably, be put to use.

Objects spiraling into black holes gain vast energies of motion from the black holes' gravitational field, and some of this energy is converted into intense radiation. An advanced civilization (we ourselves someday, perhaps) may set up outposts near a black hole—but not too near, of course—in order to tap this overflow of energy.

We might even imagine methods devised to force wandering objects closer to the black hole, close enough to push them into the ultimate inward spiral from which they will never return, and, in the course of which, floods of usable energy would be emitted, absorbed and stored. The black hole would thus be treated as a huge furnace for which any sort of matter whatever would serve as fuel.

But what happens to matter that enters a black hole? Some astronomers think it isn't really lost forever but is extruded like toothpaste into another part of the universe. At the point of emergence, it would expand and blaze with energy as a "white hole." Perhaps the mysterious quasars, far-distant objects that gleam with the light of a hundred galaxies at once, are white holes.

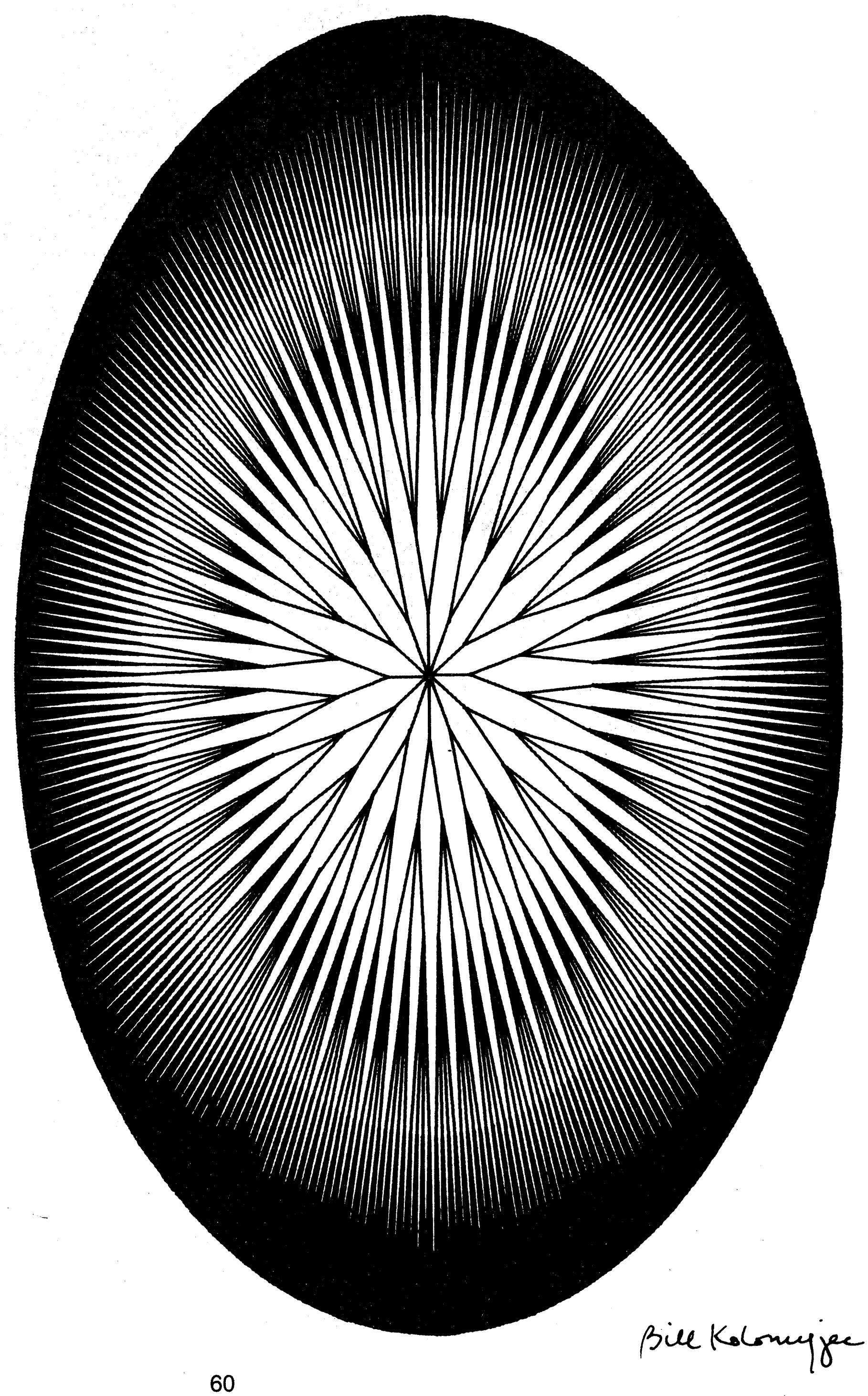
Under the extreme conditions of the black hole, matter may travel from one place to another very-far-distant place in very little time, transcending the speed limitations of the ordinary universe.

The Cornell astronomer, Carl Sagan, wonders whether the day might not come when mankind would learn enough about black holes to devise methods for surviving the conditions within them. Perhaps special gravity-resistant ships, using scientific principles undreamed of today, could carry men and goods through black holes to that distant terminus at the other end.

It may be that there are many black holes (Sagan estimates perhaps as many as a billion in each galaxy, and that the average distance between them is forty light-years, just a hop and a jump on the cosmic scale) and that each represents a different route, going from some particular place to some other particular place. Little by little, mankind might be able to map out the routes of these cosmic subway lines and work out schemes for traveling from any point in the universe to any other point by some appropriate combination of black holes.

Or perhaps some other, more advanced, intelligence (or groups of intelligences) in the universe, has already succeeded in doing this. Perhaps a Cosmic Empire exists, with prosperous industrial planets located not too far from some black hole terminus. It may be, then, that we won't have to map the universe at all, but that, when the time comes and we are sufficiently advanced, we may simply join the Cosmic Empire and become full members of the universe at once.

If it is true that black holes represent not the death of matter, but its death-and-resurrection, the universe would last forever. And with it, the various intelligences, including mankind's, might last forever as well.



WHAT IS COMPUTER LITERACY?

by David Moursund
University of Oregon

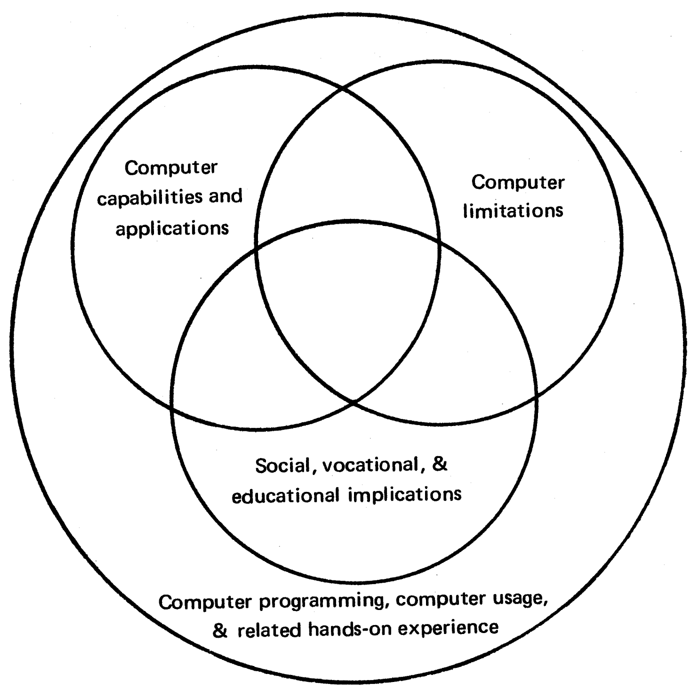
The concept of "computer literacy" is receiving much mention today. Over a period of time, we have developed a definition.

> Computer literacy refers to a knowledge of the non-technical and low-technical aspects of the capabilities and limitations of computers, and of the social, vocational, and educational implications of computers.

While such a definition can provide a focus for thought and discussion, it still does not pinpoint what is meant by computer literacy. Among other things it does not provide a measure of computer literacy nor a method for improving a person's level of computer literacy.

Most of you are familiar with the question "What is IQ?" and the answer "IQ is what is measured by an IQ test." It seems to me that we are at a similar stage of development for CL (computer literacy). Lately, many course outlines for computer concepts or computer literacy courses at the college level have been developed at Oregon and elsewhere. These courses are designed to raise a person's level of CL, and a knowledge of the content of such courses constitutes a certain level of computer literacy.

The University of Oregon's computer concepts course is a no-prerequisite, low level, introductory computer science course. Its major goal is to raise a student's level of computer literacy. Over a period of six years the course has evolved to the current point, where its content is approximately 1/3 computer programming and 2/3 non-programming materials. A Venn diagram of the course content is given below.



In the diagram the computer programming, computer usage, and hands-on experience provides a foundation upon which the non-programming aspects of the course are built. Each of these four areas strongly overlaps the other three, and each supports the other three. A well balanced course needs aspects of each of these four areas.

It seems difficult to develop a course that is coherent and well integrated, and still preserves a reasonable balance among the four major areas. Probably the computer programming and related computer usage and hands-on experience is the major source of trouble. Most computer programming texts are designed to teach computer programming. That is, their major goal is to move a student rapidly along the computer programmer path. Most such books contain little information on the capabilities, limitations, or implications of computers. The material is not organized in a manner to make it fit in well with non-programming, computer literacy materials.

To overcome this difficulty in the UO's course, I have written a 150 page book, *BASIC Programming for Computer Literacy.* This book is currently being used in the course, and seems to be a satisfactory text. It is available for $4.00 (which includes postage and handling) from the Computer Science Department, University of Oregon, Eugene, Oregon 97403.

The non-programming content of a CL course can range over a wide variety of topics, and will depend to a certain extent upon the interests and knowledge of the instructor. One cannot tell if a person is computer literate on the basis of a single true-false or multiple choice question. That is, CL refers to a broad, integrated knowledge of low level computer science. Such knowledge must include many facts and how these facts interrelate. But it is difficult to isolate a single fact that is indispensable, or fundamental.

On the non-programming content of the course, I use an objective-type final exam. In fall 1974 this exam consisted of 150 questions. An item analysis was run on these questions to determine which were the more difficult and which best differentiated the students who scored high on the test from those who scored lower on the test. Thirty of the better questions (harder, and good differentiators) have been selected and appear at the end of this article. A student making an A or high B on the exam probably answered at least 3/4 of these questions correctly.

The answers in most cases are not obvious. The 30 question test was administered to students on the first day of the winter term 1975 course. The class average was 14.75. Random guessing by all students would have produced a class average of about 12.

Taken individually, the merits of any single question are certainly subject to debate. One can easily argue that the question is not relevant to his concept of what constitutes computer literacy. Taken as a whole, however, such a group of questions provides a reasonably broad measure of many parts of the non-programming content of a computer literacy course. Try the test yourself. Try it on your students. Individual questions can provide a good basis for class discussion or individual student reading/study projects.

COMPUTER LITERACY QUIZ

1. All computers understand the language BASIC because, as its name implies, it is the most fundamental of computer languages.
2. For any problem within its capability, a computer can always solve it more quickly and cheaper than can be done manually.
3. Example of random access storage devices include:
   1. core and disk
   2. magnetic tape and punch cards
   3. disk and magnetic tape
   4. paper tape and punch cards
   5. all of the above
4. M.I.C.R. stands for magnetic ink character recognition, and is used on bank checks in the United States.
5. Which of the following does not manufacture and sell computers?
   1. Control Data Corporation
   2. IBM
   3. Digital Equipment Corp.
   4. Honeywell
   5. American Telephone and Telegraph Company
6. A typical CAI drill and practice program:
   1. works only when one is teaching elementary arithmetic
   2. aks the student questions and checks his answers
   3. forces all the students to work the same set of problems
   4. allows three incorrect responses before going on to the next problem
   5. none of the above
7. Although learning a machine language is difficult, once one has mastered it, he can write programs that will be understood by any machine.
8. The best computer programs for playing chess and checkers are based upon having the computer memorize tens of thousands of board positions (i.e. rote memory).
9. It is now possible to manufacture a single large-scale integrated circuit, called a chip, which contains all of the circuitry for a CPU.
10. The concept and use of punched cards was developed:
    1. before 1900
    2. about 1920
    3. about 1940
    4. about 1960
11. PLATO is an educational computer system which uses a gas plasma display terminal.
12. In the early days of computers, all programming was done:
    1. in FORTRAN
    2. in BASIC
    3. in machine language
    4. in UNIVAC
13. The Turing "Imitation Game":
    1. has a computer imitate a business environment to train executives in decision-making.
    2. has a person imitate a computer to find program errors.
    3. has a computer simulate a complex situation providing a detailed study of alternative effects.
    4. has a computer pretend to be human, demonstrating artificial intelligence.
14. The science of control and feedback theory is called cybernetics, and Norbert Weiner contributed a lot to this area.
15. One threat to privacy comes from the willingness of most people to provide information about themselves voluntarily.
16. Which of the following is a characteristic of a problem which is well-suited to solution by the computer?
    1. Problem solution involves value judgments
    2. All necessary decisions are quantifiable
    3. The problem is ill-defined
    4. The solution to the problem is needed only one time
17. The largest user of computers in the U. S. Government is:
    1. The Internal Revenue Service
    2. The Census Bureau
    3. The military
    4. Congress
    5. None of these.
18. When one is buying a computer system, he might purchase hardware and software from two different companies.
19. Magnetic tape is an effective medium in operations requiring frequent access to data on a random basis.
20. Very large computer programs are apt to contain undetected errors even after the programs have been used for several years.
21. NCIC is a method whereby checks printed with a special ink can be machine read.
22. By 1950 about 1000 electronic digital computers had been manufactured and placed into service.
23. The fastest core memories have retrieval times of about one millisecond.
24. Using an 8 bit code (such as on magnetic tape), how many different characters can be represented?
    1. 8
    2. 16
    3. 32
    4. 256
    5. 512
25. A computer's memory can think about and solve a problem much in the same way as a person's brain works on a problem.
26. A major problem with computerized data banks is guarding against erroneous data getting into the system.
27. A disadvantage of punched card machines is that the speed of processing is limited by the movement of mechanical parts and devices.
28. Which of the following is not an example of the administrative application of computers in education?
    1. Payroll
    2. Scheduling
    3. Student records
    4. Computer-assisted instruction
29. Why do computer scientists write computer programs to play games?
    1. Computer scientists have lots of fun doing this.
    2. To communicate the ability of the computer.
    3. To study the nature of problem solving.
    4. All of the above.
    5. None of the above.
30. Computer costs (measured in terms of computations per dollar) have leveled off in the last five years.

Answers on page 239.

A FABLE

Once upon a time, long before they invented the ball point pen (even before they invented the pencil), teachers used to teach by lecturing to the students who memorized every word their teacher said. Memorizing was the only way to learn.

One day a bright young man came up to his teacher and said:

"Sir, I have invented a pencil."

"What is a pencil?" asked the Teacher.

"It is a device to assist you in teaching and assist us in learning," replied the Student.

"What do I do with it? If I eat it will it help me memorize better? If my students eat it will it help them memorize better?"

"No," said the student. "If you use it and we use it we won't have to memorize at all."

"What kind of teaching would that be, without lecturing and without memorizing? How will I know if my students are learning if they don't memorize and recite for me?"

"That's easy," replied the student. "You will ask them to write what they have learned."

"Write?" queried the Teacher.

"Oh, that's something they will have to learn to do before they can use the pencil."

"And how long does learning to write take?"

"Perhaps a year."

"You mean to say that I will have to wait a whole year before students can use your new method for learning? Then they will come to class and write down what I say. What is the difference between that and memorizing right now, without learning to write?"

"Well, for one thing, the slow students will be able to keep up by reading the notes."

"Reading?"

"Another skill they have to acquire."

"Notes?"

"That you will prepare."

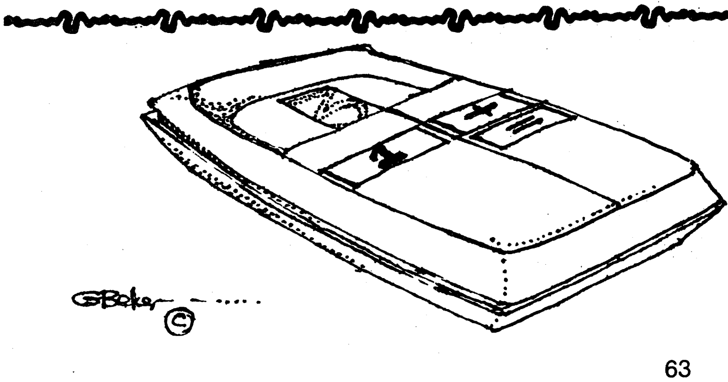
"You mean I have to do more work with your new method of learning? I like memorizing better. I still see no advantage."

"But sir, the advantages will present themselves once the system is in operation, because there will be so many things students and teachers will be able to do that they could not do before."

"You have to prove it to me before I will make any radical change like you suggest. Anyhow, haven't we always learned by memorizing?"

MORAL: Newton's 2nd Law that a body at rest stays at rest unless acted upon by an outside force is not just a law of physics.

This article is reprinted from the May, 1972, issue of Computer Mediamation. *In the words of the newsletter's editor, Dr. Sam Spero, "*Computer Mediamation *is an occasional publication of the Educational Media Center of Cuyahoga Community College, Cleveland, Ohio. Its purpose is to motivate faculty to examine the potential of computers in instruction. The newsletter explores all the ways that a computer can be used as a medium ("... means, agency or instrumentality ...") for implementing any or all facets of the instruction process."*



The "Averageman"

Allows one to endlessly enter "1+1"; The answer is always "3". Answer appears in easy-to-read LED display after a few seconds of calculating time.

UL Approved, with Case, Recharger and instruction manual.

LET US FIRST MAKE IT
or
AND NOW I SAW, THOUGH TOO LATE
or
ROBINSON CRUSOE: A BOOK FOR ALL COMPUTING SEASONS

by Robert Taylor
Teachers College, Columbia University

Upon re-reading *Robinson Crusoe* recently, I was struck by the numerous parallels between the activities of Crusoe on his island and the work of today's analysts and programmers. Crusoe laid plans; established timetables; implemented solutions, modified his implementations, and occasionally even abandoned them; provided backup for various system components; and so forth. One might even argue that he trained a junior programmer or analyst named Friday.

THIS I DID WITHOUT FIRE, BY MERE MALLET AND CHISEL

The book is so rich in readable case studies appropriate for today's analyst or programmer to ponder that any computing professional would do well to read (or re-read) it at his earliest convenience. Why should what many regard as the first novel in the English language be so appropriate to today's analysts and programmers? Because Defoe was a good story-teller and because human problem solving hasn't changed as much as we sometimes like to imagine. Consider just two examples from Defoe's text: (1) Robinson Crusoe's unsuccessful attempt to escape from his island by building his own dugout canoe and (2) his successful attempt to guard against the catastrophic loss of his goat herd. The first illustrates how need causes the desperate to overlook fatal design flaws and attempt impossible implementations. The second illustrates the design and implementation of a backup system.

Attempting to implement a catastrophically flawed system

The first illustration concerns Crusoe's attempt to build a dugout canoe with which he can escape from the island. Note that the seductive attractiveness of a portion of the system (building the canoe) clouds his critical insight into the major flaw of the overall system (launching the canoe once it's built).

> At length, I began to think whether it was not possible to make myself a canoe, or periagua, such as the natives of these climates make, even without tools, or, as I might say, without hands, of the trunk of a great tree. This I not only thought possible, but easy, and pleased myself extremely with the idea of making it, and with my having much more convenience for it than any of the Negroes or Indians; but not at all considering the particular inconveniences which I lay under more than the Indians did, viz., the want of hands to move it into the water when it was made, a difficulty much harder for me to surmount than all the consequences of want of tools could be to them: for what could it avail me, if, after I had chosen my tree, and with much trouble cut it down, and might be able with my tools to hew and dub the outside into the proper shape of a boat, and burn or cut the inside to make it hollow, so as to make a boat of it – if, after all this, I must leave it just where I found it, and was not able to launch it into the water?
>
> One would imagine, if I had had the least reflection upon my mind of my circumstances while I was making this boat, I should have immediately thought how I was to get it into the sea: but my thoughts were so intent upon my voyage in it, that I never once considered how I should get it off the land; and it was really, in its own nature, more easy for me to guide it over forty-five miles of sea, than the forty-five fathoms of land, where it lay, to set it afloat in the water.
>
> I went to work upon this boat the most like a fool that ever man did, who had any of his senses awake. I pleased

myself with the design, without determining whether I was able to undertake it; not but that the difficulty of launching my boat came often into my head; but I put a stop on my own inquiries into it, by this foolish answer: Let us first make it; I warrant I will find some way or other to get it along when it is done.

This was a most preposterous method; but the eagerness of my fancy prevailed, and to work I went. I felled a cedar tree, and I question much whether Solomon ever had such a one for the building of the Temple at Jerusalem; it was five feet ten inches diameter at the lower part next the stump, and four feet eleven inches diameter at the end of twenty-two feet, where it lessened and then parted into branches. It was not without infinite labour that I felled this tree; I was twenty days hacking and hewing at the bottom, and fourteen more getting the branches and limbs, and the vast spreading head of it, cut off; after this, it cost me a month to shape it and dub it to a proportion, and to something like the bottom of a boat, that it might swim upright as it ought to do. It cost me near three months more to clear the inside, and work it out so as to make an exact boat of it: this I did, indeed, without fire, by mere mallet and chisel, and by the dint of hard labour, till I had brought it to be a very handsome periagua, and big enough to have carried six-and-twenty men, and consequently big enough to have carried me and all my cargo.

When I had gone through this work, I was extremely delighted with it. The boat was really much bigger than ever I saw a canoe or a periagua that was made of one tree, in my life. Many a weary stroke it had cost, you may be sure; and there remained nothing but to get it into the water; which, had I accomplished, I make no question but I should have begun the maddest voyage, and the most unlikely to be performed, that ever was undertaken.

But all my devices to get it into the water failed me; though they cost me inexpressible labour too. It lay about one hundred yards from the water, and not more; but the first inconvenience was, it was up hill towards the creek. Well, to take away this discouragement, I resolved to dig into the surface of the earth and so make a declivity; this I began, and it cost me a prodigious deal of pains; but who grudges pains that have their deliverance in view? When this was worked through, and this difficulty managed, I was still much the same, for I could no more stir the canoe than I could the other boat. Then I measured the distance of ground, and resolved to cut a dock, or canal, to bring the water up to the canoe, seeing I could not bring the canoe down to the water. Well, I began this work; and when I began to enter upon it, and calculate how deep it was to be dug, how broad, how the stuff was to be thrown out, I found by the number of hands I had, having none but my own, that it must have been ten or twelve years before I could have gone through with it; for the shore lay so high, that at the upper end it must have been at least twenty feet deep; this attempt, though with great reluctancy, I was at length obliged to give over also.

This grieved me heartily; and now I saw, though too late, the folly of beginning a work before we count the cost, and before we judge rightly of our own strength to go through with it.

This passage gives a very graphic picture of how the seduction takes place. His decision to "first make it" and then to worry about launching the boat later characterizes all too well the misplaced optimism we all seem to succumb to from time to time in designing and implementing systems.

Backing up major system components

The second illustration concerns Crusoe's design and implementation of a backup supply of goats. Earlier in the narrative, Crusoe went to considerable pain to capture and domesticate some wild goats and they have since become a principal component in his diet.

While this was doing, I was not altogether careless of my other affairs: for I had a great concern upon me for my little herd of goats; they were not only a ready supply to me on every occasion, and began to be sufficient for me, without the expense of powder and shot, but also without the fatigue of hunting after the wild ones; and I was loath to lose the advantage of them, and to have them all to nurse up over again.

For this purpose, after long consideration, I could think of but two ways to preserve them: one was, to find another convenient place to dig a cave under ground, and to drive them into it every night; and the other was, to enclose two or three little bits of land, remote from one another, and as much concealed as I could, where I might keep about half a dozen young goats in each place; so that if any disaster happened to the flock in general, I might be able to raise them again with little trouble and time; and this, though it would require a great deal of time and labour, I thought was the most rational design.

Accordingly, I spent some time to find out the most retired parts of the island; and I pitched upon one, which was as private, indeed, as my heart could wish for: it was a little damp piece of ground, in the middle of the hollow and thick woods, where, as is observed, I almost lost myself once before, endeavouring to come back that way from the eastern part of the island. Here I found a clear piece of land, near three acres, so surrounded with woods, that it was almost an enclosure by nature; at least, it did not want near so much labour to make it so as the other pieces of ground I had worked so hard at.

I immediately went to work with this piece of ground, and in less than a month's time I had so fenced it round, that my flock, or herd, call it which you please, who were not so wild now as at first they might be supposed to be, were well enough secured in it; so, without any further delay, I removed ten young she-goats and two he-goats to this piece; and when they were there, I continued to perfect the fence, till I had made it as secure as the other, which, however, I did at more leisure, and it took me up more time by a great deal.

That Robinson Crusoe should provide backup for such an essential component of his survival as his goat herd seems so obvious as to need no comment. However, how often do professional programmers and analysts fail utterly to provide backup for equally essential components of their own systems?

Further reflections

Cases like the two just presented abound in the book. Mistakes and successes are even replicated, just as they too often are in the contemporary systems world. The insight which these cases afford is particularly appealing because the adventure does not purport to deal with systems or computing at all! Moreover, though the cases largely lack the complex human interface problems common to systems involving many people (Robinson Crusoe lives and labors alone on his island for most of the narrative), this simplification allows Defoe to strip problems to their essentials. Crusoe's work is done primarily for himself. *He is his own end user.* As such, he must, in the most literal of all senses, "live with" the system he creates. Defoe thus presents us with the entire systems environment in a very compact form.

What else lies within the narrative of Crusoe's life on the island? Get a copy of *Robinson Crusoe,* read it, and find out. Be prepared to see your systems work in a refreshing new light. And should you feel too critical of Crusoe's solutions to his many problems, notice how long he survived. His solutions were viable enough to keep him going for 27 years, 2 months, and 19 days. Can any of us top that?

Some Thoughts

John R. Lees, Jr.
Associate Editor *Creative Computing*

I have been thinking quite a lot lately about *People's Computer Company* and *Creative Computing*, and what the existence of such publications means; about the explosion in computer, minicomputers, microprocessors and the "hobby" computer thing, and what such an explosion means; and about community communications and free schools and deschooling society and social change and the certain knowledge that world-wide disaster is imminent, and I am wondering if anyone has the faintest idea as to what is going on.

We appear to be rushing head-on in this country (only in this country?) into something which I have started thinking of as the distributed-computer society. Two computers in every garage . . . and in every washing machine, oven, radio, watch, telephone, doorbell, in short, a computer as an integral part of every technological device. The $10 microprocessor is here and getting cheaper every day. Already everything is electronic, soon, everything will have a microprocessor snuggled somewhere within it. This worries me.

To help you see why it worries me, let me rephrase a sentence from the above paragraph: Already our *tools* are all electronic, soon, our *tools* will all have microprocessors as integral parts. This is an important point because a microprocessor is an *inherently incomprehensible device;* a device which cannot be understood out of the context of an extremely complicated, elite technology. How does a microprocessor work? No, not what it does, but how it actually does it? Is that not really important? We are beginning to build things using tools which we do not really understand. Do we understand what we build with those tools?

We are touting the computer as the educational device to end all educational devices, but we frequently stress the point, "Don't worry about how it works, that isn't important," often adding, "I don't really understand it myself." I can't help but think of what Ivan Illich says in *Deschooling Society* about the radio, how mass production techniques changed it from an educational source of parts and electronics knowledge into a disposable throwaway. With computer it's worse; in many cases we don't even understand the software. In many cases we are prohibited from understanding the software because it is critical or proprietary or necessary to "system security" or simply written in a half-assed way which makes it impossible to figure out.

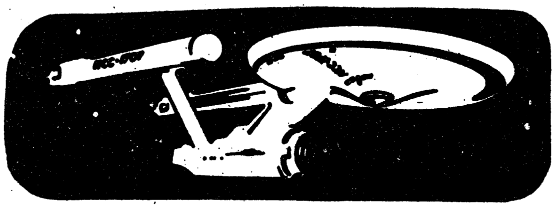
Some of us are thinking of basing various social revolutions around the "inexpensive" computer. Try comparing the price of an Altair against the world median income! And how secure is any revolution based on a black box, the production, the understanding, of which is not in the hands of the revolutionaries? (Seen any basement diffusion furnaces lately? Ion-Implantation in the bathroom?) Whose revolution is it, anyway? The people's? Or the fraction of a fraction of a percent of the population who at least partially understands the technology involved? Think of the *eliteness* of even the readership of *People's Computer Company* and *Creative Computing!* Are we serious, or are we just playing with our fascinating new toy?

I'll assume that we think we are serious, because computers are transforming world technology. It's hard to comprehend how omnipresent the influence of the computer has become. Environmental and social impact statements are not required on new technologies that bloom overnight and captivate all thinking in the wink of an eye. And that's the way it happened!

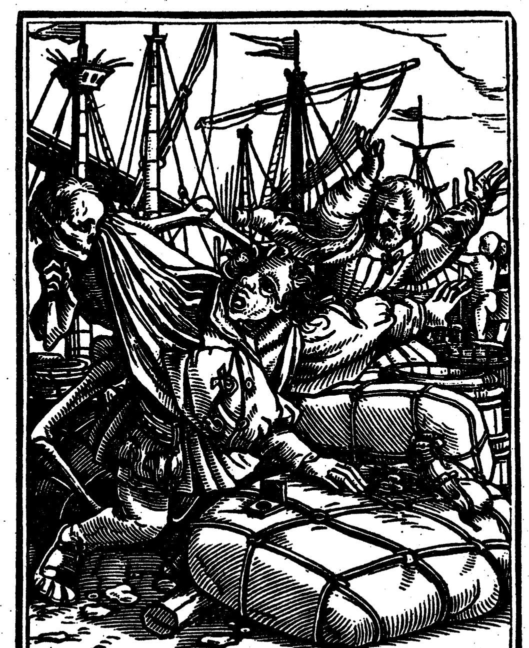
I entered college in 1971 and the hottest thing going was the Heathkit electronic calculator that would do four functions for only $130.00. It was fantastic! By the time

This article also appeared in *People's Computer Company*. (P.O. Box 310, Menlo Park, CA 94025) Mar/Apr 1976.

I graduated the Altair was in production and the 8080 had 78 instructions for $130.00. That is starkly unbelievable! Except that it happened and it's still happening. I entered college from one culture and graduated into another.one. I'll have my master's degree in one year and what will it be like then? Star Trek on every TV in the nation, probably. Is that the idea?



The idea is for some people to make a lot of money and expand their industry and keep the GNP growing. That is the gut force behind the computer explosion, that and the fact that the computer is the advertiser's dream come true. "Here is our universal do-all. Take a close look at it. We guarantee that you can think of something to use it for! If you can't, well, sorry, it looks like your business is obsolete." The self-expanding product; the product which grabs you by the throat and says, "Thou shalt do it my way, or your investment is worthless. And by the way, I'd do it a lot better with another 32K."



To return to my main point, I guess what really bothers me is that we are beginning to base so much of our everyday world on technologies which are not intuitively understandable. We no longer feel that it is necessary to understand our tools. I believe that if we do not understand our tools, then we do not control our tools, our tools control us. The people who do understand our tools control us.

If I am the end user of a computer statistical package, but I am not a programmer, then if someone changes the package I must change. If I am a programmer, but don't understand hardware, then if someone changes the machine I program for, I must change. Even if I understand all facets of the computer I use, from software to hardware, I am still in trouble, for if someone changes the design of an integrated circuit device such as a microprocessor or a memory chip, there's not a damn thing I can do about it except change to suit Them.

All users of advanced technologies are subservient to the elite who understand and control those technologies.

Even the elite represented by *PCC* and *Creative* are not very elite. How do *People's Consumer Company* and *Creative Consuming* grab you? (Down, Dragon!, Down!) I'm not too taken with those names, but even though I am fairly knowledgeable about computers I realize that I am basically helpless. I am still only a user of someone else's technology. If things continue on in the same way they are going right now, I am not sure that I see the situation getting any better.

Aha! The way is clear for the usual Basic Question: Must Things Go On This Way? No, I'm not denouncing computers or technology or capitalism or anything else. Perhaps there was only one way to reach this point in history, it makes little difference, we are here. The distributed-computer society is upon us. We know that computers are, if nothing else, great toys and we have hopes that they can be much more.

But must computers remain black boxes? Must computer technology, itself, remain of no educational value? Must control of the use of computers for social change remain, ultimately, with others than those who are trying to bring about change? Must the public forever fall farther and farther behind in understanding the devices with which it is manipulated?

Okay, I am but an egg, and all that, and I don't have many answers, so I'm asking: Can we have an understandable computer technology? Is the way we are doing things now the only way to do them? Can we transform computers into tools which most people can understand and use? Can we have computers for people? Can we use computers to bring about useful social change? Can we reconcile personal computers in this country with the fact that much of the world population will starve to death by the end of this century of technological progress? Are we really doing something useful in terms of the future of this planet, or are we really just playing games?

Those are some pretty brutal questions, and to some degree I have been playing the devil's advocate, but I really want to find some answers. So now that I've raised the points, and I'll admit that some of the things I've said could use some expansion and clarification, let's have some discussion.

INFORMATION ANYONE?

Illustration by George Beker

by Bill Griffith
Boston College

You are nobody unless you have your share of information. Money is not power, knowledge is not power — information is power.

With the CIA collecting information on private citizens (Why don't they stick to overthrowing foreign governments?), commercial credit companies recording the contents of your trashcans and your seven year old using words like "software" and "hardware" and "PL/I", is it any wonder you wonder? "It all looks too big and complicated.", you say. "What can I do about it anyway, I'm not an expert?" Well, it can get pretty scary if you think about the ramifications of all this in our computer-infested society. It's all very serious. But it seems to be serious in a way that caused people to do nothing about it. They can't even laugh about it. Now scary is O.K. if you use the fear towards productive defense and laughter is O.K. if it puts things into a workable perspective.

I work in a computing center and some times it gets pretty hectic. One night, after a particularly grueling day, I passed out at about 7 P.M. and I dreamt that I couldn't speak unless I talked in JCL. Now, JCL, for those of you who have the fortune not to know, is short for Job Control Language, which is kind of a pseudo-language used to get your work through IBM's larger computers. We've all had nightmares — tigers chasing you, falling down endless holes — the usual nightly ramblings of our collective unconscious. This one however, freaked me out a little more than the time I was about to be crushed to death for playing with a OUIJA board at the blessing of the fleet in Gloucester. No one could understand me except systems programmers and the computers themselves. I couldn't order dinner, I was banned from the singles bars (my wife was happy about that — serves me right), and I was bitten by our pet gerbil. Fortunately, I woke up before I starved to death or whatever. The ramifications of this nocturnal psychic spasm began to revolve in my head and I was led to an appendix in *1984* which discussed the language of the future, NEWSPEAK.

> *The purpose of NEWSPEAK was not only to provide a medium of expression for the world view and mental habits proper to the devotees of Ingsoc, but to make all other modes of thought impossible.*

It takes a very special turn of mind to be able to communicate with a computer. You have to use a language which is unambiguous and any "sentence" you construct will, at least in the context, be univalent—i.e.—it means one thing and only one thing. Now I admit it can be a real kick (and useful) for linguistic puzzle maniacs both to construct and to try and break such languages. However, if you continue to think in these strictured terms, it is rather unlikely that you will say anything that will ultimately benefit humankind. After a hard day at the coding pad (the blocked and lined paper programmers use to record their musings), I once went to a cocktail party in Cambridge and I was unable to enter into the current discussion about the 19th century symbolists because I couldn't say anything relevant in FORTRAN. That was the day I chose to go into management. I figured that this might save me from the shrink or at least if I went to the shrink I could still talk to him/her. At the time I didn't realize that programmers don't need shrinks anyway and that managers do because they can still think about what's bothering them enough to have it bother them.

I often ask myself what in the name of heaven ("heaven" is a legitimate variable name in most computer languages) we are doing with these monolithic monsters. Pretty soon we won't even be able to use words like "freedom," "liberation," etc., because they won't mean anything anyway. You've got to have a counter-vocabulary to produce creative change; but the language is fast reducing, at least in the career circles I run in. In contrast, the social environment (at least, mine) doesn't seem able to respond at all. And these people, I like to think, are intelligent, caring individuals but when they ask me what I do I find myself speechless or mumbling. When you think about the so-called lack of religion in our "post-modern" world, you quickly find substitutes abounding in the techno-scientific fields. You can tell where pseudo-religion is because that is where mystery is. Any religion worth its salt leads to knowledge, not mystery. So the new mystery-religion is upon us in the form of technology and the new high priests are the systems analysts and programmers. Every cult has its symbols and computerese is no exception. It's impossible to fight, if you stand in awe and wonder. I find awe and wonder best reserved for sunsets and rainbows and only a way of avoiding responsibility when it comes to technology. As in Newspeak, some programmers sit up nights trying to say more, more efficiently and with less "words" and giving themselves gold stars when they succeed in doing it. I admit to all the "wonder"ful things computers have done — they have eased out bureaucratic pains, made our lives more

One night, after a particularly grueling day, I passed out at about 7 P.M. and I dreamt that I couldn't speak unless I talked in JCL.

efficient and carefree, simplified government and helped catch income tax cheaters and who am I to argue with "rationality." Anyone who stands in opposition to "technique" certainly has a suspect mental balance. Who can stay the tide of technological sophistication? Now that's a word — sophistication. It means neat, cool and in the know — sexy even (we'll get to that in a later article for you neo-Freudians out there.) But one of the meanings which we overlook is *complex.* Whenever a computer salesman uses the word sophisticated, I get hives. And do we remember the sophists? They didn't say anything but how wonderful it sounded and who would volunteer ignorance of what they were talking about? Somehow, I don't fear any organized takeover of the language a la *1984* but I sure as hell worry about habitual lack of responsibility — we don't need to be taken by force, why waste the effort — we'll take care of ourselves ourselves. So, next time somebody says a word you don't get — ask them to explain it until you understand. You didn't get into a responsible position by not asking "stupid" questions, so why stop now? Get out that copy of Webster's and cherish it — hug it to your breast. It may be more important than the Bible in the decade to come. Have you seen *Fahrenheit 451?*

There is another side to this technological revolution and that is information. Even the computer is falling into the background of information. It is no longer computer processing or data processing but information processing. Everybody has to have information—managers need information, congress needs information, hospitals need information. You are nobody unless you have your share of information. Money is not power, knowledge is not power—information is power. Some people don't seem to want to know anything, but they sure as shootin' want to know about everything. And if you give them a T.V. computer terminal they will certainly use it even if they don't need it. Now in order to fulfill this addiction to information, we need information. I remember someone quoting J. Edgar as saying "We have dossiers on 95% of the American public and we must close this gap." Why do people collect information—the IRS, the CIA, the FBI, commercial credit companies? Only to have it—I know they give other reasons but "only to have it" is the truth. They have to supply the habits of the "information seekers." If you are a manager and you have no MIS (Management Information System) where is your credibility? What do you want to know—it may not matter, you'll figure out something... later. Right now, collect that information. Meanwhile... all the information has been collected. It reminds me of a thought Marshall McLuhan put forth in *Understanding Media*—you invent T.V. and what a marvel it is; what a wonderful communications media. Then you put the control of the media into the hands of the networks and they don't worry so much about all the wonderful things we could communicate but how to fill up the time slots. Now this wouldn't be so bad if people didn't have the mistaken idea that all we are is what somebody knows about us. I don't like to see tighter social control via data banks but I feel that if we are only our social selves (what is known about us) then we are in big trouble. In fact, believing it is the best way to reinforce it. It breeds fear, lack of self-expression, secrecy and in the age of communication we find ourselves communicating less and less. I would like to shout from the housetops, "INFORMATION IS NOT TRUTH." You see, when we moved from data to information a very subtle transition took place. Data is not just data but reality (where did I hear that?) when it is information. It is now in-FORMED—it has form. Somehow, we are lead to believe that the evaluation has already taken place when the information arrives on our desk. IT HAS NOT. What about the content? Who cares? We should. Someone once told me (or did I think it up all by myself?) that management information is anything that a manager reads that was printed by a computer. Oh well! The problem is that information has a rather unbiased tone about it when at the same time it is heavily biased. Oh for the days of the Buddha when "all you were was the result of what you had thought" rather than today when all you are is what somebody knows about you. So in between your Peter Drucker and your IBM manuals (they are second in publication only to the U.S. government—that's one for the time capsule) see if you can't sandwich a copy of some O. Henry or some Mark Twain or the Bhagavad Gita. It'll do ya good.

In closing, I would say that I am not a cynic—far from it, but one who wants to take the responsibility for my small area of the universe as all should. I do it badly sometimes and sometimes well—but I do. The computer has no intrinsic moral bias but when it imbeds itself in a culture as it has today—it does. "Back to nature" won't help. You'd better have a book on systems in one hand and a philosophy book in the other or it is all over. The only way around it is through it.

Opinion

THE GOVERNMENT DINOSAUR

Charles Winn
530 McAlway Road
Charlotte, N. C. 28211

Over the years we have witnessed the growth and development of what could best be described as the governmental dinosaur — a large unwieldy critter, who though well-meaning, thinks slowly, moves awkwardly, reacts tardily, and is beset with many ailments and infirmities. Like its animal form it may well ultimately become so ineffective and weighed down as to collapse into death and extinction. This governmental monstrosity has assumed duties and responsibilities out of all proportion to its intrinsic talents and abilities. It was designed or put together piecemeal to serve in slower less demanding times and has had new and challenging obligations thrust upon it.

Human comprehension and reaction at the governmental and legislative levels is incapable of coping with the multitude or complexity of rapidly changing facts, figures, and demands. New discoveries, advanced technologies, and other newly revealed knowledge moves at such a rapid pace that it defies human assimilation and coordination by those who require these benefits and insights most. Government, like the great retarded beast, stumbles and staggers its way along — trampling many underfoot, wavering from a proper course, over or under compensating, and becoming more lost and confused. The mere size of the organism renders it ineffectual. The multitudinous bureaus, departments, offices, agencies, and divisions grow like cancerous cells in wild profusion. When they do function or respond it is not always in conjunction with other parts or in their best interests. Communication and cooperation between these segments is a nightmare activity.

Other investigators and writers have examined and ably described the wild inconsistencies and glaring deficiencies of this tremendous organization. We are all familiar with some of the errors, shortcomings, and the ineffectiveness of the government as it exists today. It is no one individual's fault that these conditions prevail, but does this ignorant beast in its present form warrant our trust and confidence? Does it even begin to efficiently serve the potential and actual needs of the American populace?

This beast is not peculiar to only the United States, but has its counterparts in other countries. In fact, there the colossus may be even more ponderous, awkward, and intractable. Our chief concern however, is with remedies, improvements, and innovations. This process shouldn't require revolution in anything but our thinking and our solutions.

Initially, we might try a massive undertaking — the creation of the largest, most efficient computer yet conceived or assembled — a computer capable of retaining, assimilating, and processing information relating to all phases of governmental concern or interest. The machine would require a sizeable complex of experts to insure its proper function. Initially, to insure the accuracy of data, statistics, and information gathered, there would have to be a virtually fool-proof systems organization to gather such data and a board of experts from various fields and disciplines, men above suspicion or reproach and of sterling character, to make final judgments on what material was to be programmed and to oversee those that physically entered information into the machine. The margin for error would most certainly be less than it is under present random circumstances.

The machine could be programmed to temporarily reject material that was inconsistent with or deviant from information already retained. This material, in turn, could be reviewed by the board to ascertain its accuracy before resubmission. Computer experts and designers could build the necessary safety procedures and safeguards into the machine at its inception to prevent tampering and misuse.

This computer might well be dubbed Uncle Sam and could be relied on for accuracy, rationale, and impartiality. The machine may not be infallible, but would be infinitely superior to the mass confusion and human error that exist today. It could provide the president and his cabinet, the Congress, and other essential government figures with up-to-date and comprehensive information on which decisions could be partially based. The extent to which this information would be acted upon could be determined by experience and performance. Uncle Sam would obviate much of the reliance placed upon outside lobbyists, who are almost always selfishly motivated, and self-appointed experts whose information is often, at best, questionable. Uncle Sam could offset and reduce much of the mediocrity, partiality, and outright chicanery that now exists in the government area.

If these machines can be relied upon to deliver expensive and complicated space vehicles to obscure destinations with unerring accuracy and exactitude, they could certainly be channeled to the task of eliminating much of the human weakness and shortcomings, the boondoggery, and all of the prolonged hassling that they generate in government affairs. The legislative and administrative systems have become a snarl of ineptitude. The idea of Uncle Sam is not inconsistent with the human element for humans are, in fact, computers of a sort that lack the total recall, instantaneous operation, tirelessness, and the uncanny accuracy of their machine counterparts. They work very well in harmony and in conjunction with each other — the one providing what the other lacks.

The slowness and frequently the absence of reaction of the government to social needs results in the creation of a sense of alienation and helplessness in the general populace. Many experts and observers feel that the citizen's inability to express himself and make himself heard is the greatest weakness and ill in our society. Millions regard themselves as ineffectual pawns incapable of bringing about change or improvement even at the lowest levels of government and administration — they crave recognition and a sense of participation.

This pressing need might well be also answered by computer technology and provide a step toward truly democratic procedures. Present knowledge is sufficient to devise an electronic voting system whereby the average voter could be consulted and heard on basic issues. A coded card, similar to a credit card, might be issued to a qualified voter. This, in turn, could identify the voter upon insertion into an apparatus incorporated into the telephone system, thereby permitting the voter to dial his vote by following designated procedures. This type of equipment could be in every home possessing telephone service and conveniently located booths could serve others who lack individual telephone service or who may be away from their homes.

Radio and television could be more fully employed to keep the public abreast of current issues and prepared to make intelligent decisions on matters relating to the public welfare. Computers could tally these votes and furnish them directly to legislative government thus bringing about a return to democratic procedures unknown since the days of the great Grecian cities. If anything this would be improvement upon any democratic procedure ever employed on a large scale basis. For the first time the public could express itself directly without all of the representative inconsistencies and blindness that we now experience.

Another improvement would be to gradually transfer much of the governmental function to segments of private industry. This would be in keeping with our professed faith in free enterprise and consistent with the fact that business is much more efficient and goal oriented than the present framework of governmental administration.

These duties and responsibilities could still be originated, supervised, advised, and checked upon by government personnel at the upper levels. The functions and activities themselves would be the assignment of private companies or independent agencies that are geared for efficiency and results. This arrangement would produce greater accomplishment of goals and at great savings to the taxpayer. There would be benefits to be reaped all the way around. One such benefit would be the elimination of duplication and the removal of personnel who are often incapable, indifferent, or lacking in motivation or dedication. The present efficient government employee would have nothing to fear for he would readily be absorbed into the new system with extended responsibilities and a much greater opportunity to prove himself.

There is nothing sanctified or holy about governmental form that precludes its being changed or altered to conform to new demands and needs, to make better responses. Many of the founders of our country expressed this attitude freely and made allowances for it in the Constitution. If we're going to retain the old dinosaur, let's at least give it some assistance and provide it with modern aids. It deserves our wholehearted support and is most certainly in our best interests.

Juvenile Information System Killed

Last April, an unusual alliance of computer professionals and civil liberties advocates managed to halt the implementation of a Juvenile Information System in the Santa Clara, California Probation Department. The victory was the result of a lot of people learning about other people's problems and beginning to care about them.

Dorothy Ellenburg, director of the Council for Community Action Planning, Inc. in San Jose brought to light the problem with JIS. The system, which was to centralize the files of juveniles in order to enable agencies all over the county to have access to them, raised questions among her group because it called for the inclusion of the names of juveniles who weren't officially "delinquent". With the help of Carol Guddal, a technical writer for Hewlett-Packard and more than 30 other volunteer computer professionals, the CCAP studied the abuses to which the JIS might be subject, and were able to convince the county board of supervisors to defeat its implementation. A central objection to JIS that the computer professionals brought to light was its lack of security. With terminals all over the county that would produce not just displays of data, but printouts too, there was little that could be done to control the confidentiality of information. CCAP argued that such easy access to information about children who weren't even criminals would be tantamount to labelling them as "asocial kids". The computer experts who studied the implications of the system agreed.

A byproduct of the joint effort was the establishment of a data confidentiality commission in the county. The group is studying the implications of computerization and information sharing of the more than three million individual records compiled by county government on its citizens. Carol Guddal stated very clearly what she and the other computer experts learned and intend to remember as a result of their involvement: "It's easy to become isolated from the use of these systems after they've been shipped out the door. . . . I'm not so sure that I'll be guilty of it anymore. I never realized how profoundly a computer system can influence a community. They are not toys."

SYSTEMATIC TRI-PHASE PROJECTION

(Prophetic Rhetoric)

by Linda S. Labelson

Our company has integrated a transitional logistical concept. The total policy options consist of flexibility and capability with a functional parallelled time-phase projection.

Total hardware mobility is to be monitored with programmed digital synchronized contingency.

Our objective is third-generation programming; responsive, balanced management, with a totalitarian compatibility.

Circumspectly, we purport an accelerated expediting of this highly technical innovation.

THE MAGIC OF ELECTRONIC FUNDS TRANSFER
OR
THE LITTLE GUY TAKES IT IN THE EAR (AGAIN)
OR
WHY FOREIGN SUBSCRIBERS MUST NOW SEND CASH

by David Ahl

It all started innocently enough one day when I received four French checks back from the bank (Morris County Savings) full of staples, folds, and mutilations. Accompanying them was a teller's slip with the cryptic notation, "not cashed due to change in foreign exchange regulations."

Okay, set them aside and remember to inquire about them next time I'm in the bank.

Next day, I receive another slip from the bank *charging* my account $31.84, again with a cryptic notation on the teller's slip, "charged to your account by Manufacturer's Hanover Trust Company." Strange, I thought, since I'd never gone there in my life.

I decided a visit to the bank was in order, even though I normally avoid it like the plague (just can't stand long lines and well-meaning-but-not-very-bright tellers).

Decided to see an officer. Did you know that the "officers" sitting in the open area where you open new accounts and redeem bonds and get your signature approved et al are really just pseudo officers? After two of them huddled for about 15 minutes, one left to check the account "on the computer," some more discussion and then this woman pseudo officer announced, "Manufacturer's Hanover charged your account for cashing four foreign checks."

"But they *didn't* cash them," said I. "Here they are."

"Well then they charged you for processing them."

"By processing, you mean returning them uncashed" I said, trying to keep emotion from creeping into my voice.

"I guess you'll have to take it up with Manufacturer's Hanover," she said, cleverly trying to pass me out the door.

"But they aren't my bank, you are!" I said, allowing my emotion somewhat more open rein. The discussion continued in an inconclusive but gradually loudening manner, when she hastily excused herself. She returned in 10 minutes with a large man. I had visions of being escorted out the door but he showed me into a big office in the back. If not a real officer, he was less pseudo than the ones in front.

He explained that First National City Bank and Manufacturers Hanover were Morris County's foreign correspondent banks. There was another link through Heritage-Iron Bank but I never did understand that one. Anyway, the correspondent banks had announced that they were instituting a wonderful new computerized Electronic Funds Transfer System and they were doing away with time-consuming, messy, paper transactions and doing everything by electronic wire transmission. The announcements added that the cost per transaction was only $7.96 (M-H) and $9.26 (FNCB). This compares to $1.50 per check in the "old-fashioned" paper way. The announcement also said that regretfully, personal accounts could not be handled at this point.

In short, what had happened to our account was this: we received 4 subscription checks from France (total value $38.00). They were sent for collection to M-H who charged $7.96 ea. or $31.84. They were not collected (personal checks) but we were still charged. The officer promised to "look into the situation."

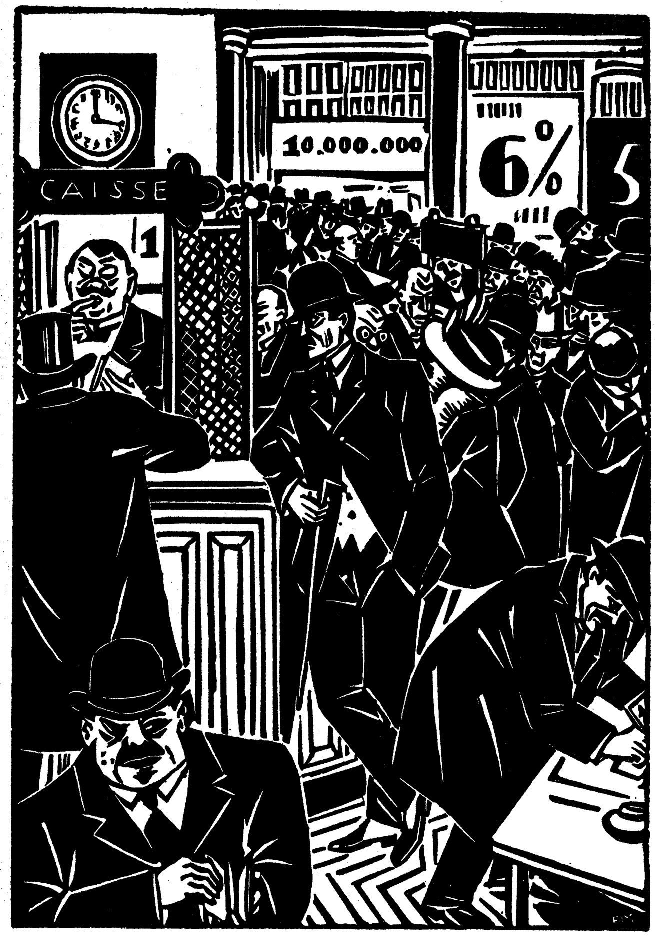
That was six weeks ago. A call yesterday indicated that Manufacturer's Hanover finally replied to the effect that they incurred the cost and it was up to us to pay (even though no service was performed).

I also inquired whether the use of cash had been outlawed yet and they allowed that it was still OK. Hence, instead of adding $8.00 or $9.00 to foreign subscriptions, we are suggesting that foreign subscribers pay for subscriptions *in U.S. currency*. Yes, I know that sending cash through the mail is "dangerous" but I either (1) have to open a foreign bank account in every country in which we do business or (2) add $8 or $9 to foreign subscriptions or (3) risk mail theft. Frankly, the last seems the most sensible alternative.

(We can still handle checks from Canada, but I suspect that won't last long).

The recent postal rate increase, has also forced a slight adjustment in foreign subscription rates. Also, no more foreign student subscriptions—one rate for all.

Foreign Subscription Rates:
1 Year $10.00 3 Years $27.00



Instructional Computing In Schools — How, When, What?

by David Ahl

The survey completed by the American Institutes for Research in 1975 projects that every secondary school in the USA will have computer access by 1984 (appropriate date!). But what kind of access? And, more important, will it really be used to advantage?

Personally, I am appalled at what I see in numerous schools that think of themselves as having a computer available for instruction. For example, one local school has an IBM 1130 and one computer science course. In the course, the students "write" (really copy) three canned FORTRAN programs which they then keypunch (Wow!) and give to an operator to run. They get results 3 days later. This is computer education??

I strongly believe that schools should align their education more closely with what will be found in the real world when their students get out. In other words, schools should be at the leading edge of the state of the art instead of lagging it by 10 to 20 years because of *imagined* cost constraints.

What this means in concrete terms is that schools should have one or more timesharing terminals to a powerful large or medium scale system with ability to manipulate very large problems. Secondly, they should have a terminal into an information network such as Lockheed Dialog or the NY Times Data Bank. Third, they should have a variety of mini- and microprocessor systems around for all sorts of things. Indeed for under $1000 a school could build a new MPU kit or two every year! Just think, if the same $5500 per year that the average school spends on *one* timesharing terminal were put toward kits, after 5 years, the school would *own* 25 working computer systems.

The advantage of building a computer kit are many-fold. It's a project that can be done jointly between the vo-tech classes and the math and science classes. Those students who want to get involved with the nitty-gritty of the hardware have an opportunity to do so; students who want to write operating systems, compilers and interpreters can do so; and when the system is finished you have a nice BASIC-speaking computer for everyone to use.

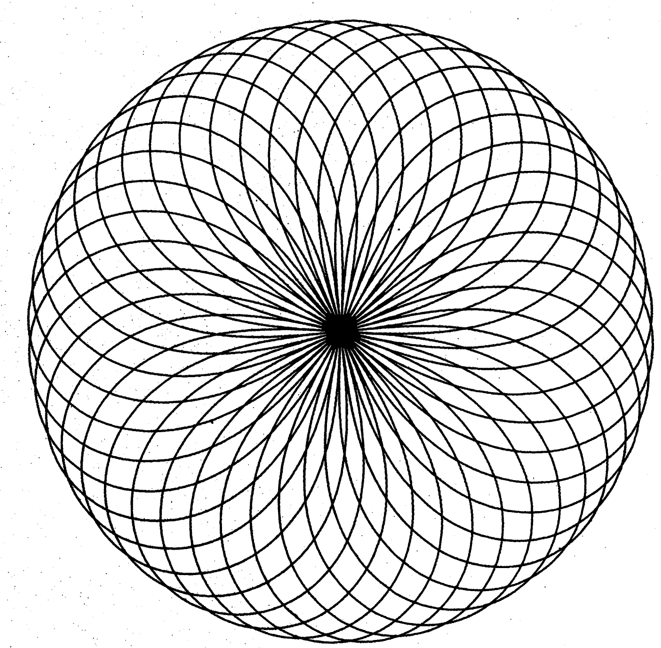
Schools should also attempt to assemble a wide variety of devices and peripherals such as A/D and D/A converters, music synthesizers, audio cassette recorders and interfaces, TV interfaces and character generators, plotters, etc.

Only by having a full range of computer access (both remote and local), hands-on hardware and systems, and variety of related equipment will students begin to get the education they'll need to utilize the tremendous computer power that will be theirs in the 1980's.

An Ideal?

Here's my personal idea of the ideal computer facilities for an average high school in 1976-77 — DHA.

1 hard copy and 1 CRT terminal to comprehensive timesharing system	DEC LA-36 and Hazeltine 1000 to On-Line Systems Timesharing.
1 printing CRT terminal to information network	Dataspeed 40 to Lockheed DIALOG.
1 terminal with animated graphics	General Turtle TT250
1 BASIC Language calculator with plotter or CRT graphics	HP 9830A with printer and plotter or Tektronix 4051
1 mini (kit) with audio cassette, terminal and TV driver	Altair 8800 (8K), ASR 33, Chromemco TV dazzler
1 mini (kit) with A/D, D/A, Music Synthesizer	SWTP 6800, Performer Music Synthesizer



"Rotated Infinity" by Tony (Core Dump) Martin

How much longer will a computer illiterate be considered educated? How long will he be employable and for what jobs? Is it enough to be merely a subject of computer administered instruction?

Should the computer teach the student, or vice-versa?

by ARTHUR W. LUEHRMANN
Dartmouth College
Hanover, New Hampshire

This sermon begins with a parable.

Once upon a time in the ancient past there was a nation in which writing and reading had not yet been invented. Society was as advanced as possible, considering that it had no mechanism for recording the letter of the law or of writing agreements, contracts, or debts. Nor was there a way of recording the heritage of information and knowledge that had to be passed on from generation to generation.

As a result, a great fraction of the total effort of the society was spent in oral transmission of information. Master teachers, who themselves had been taught by older master teachers, lectured before children and young people of the society. Training a master teacher was a long and expensive process, and so the society could not afford many. For reasons of economy the curriculum was quite rigid and lectures were on a fixed schedule. Teaching, obviously, was a labor-intensive industry based on skilled, expensive talent. Education, per force, was a luxury that could be afforded by the elite classes only.

Then, one day, writing and reading were invented. Not surprisingly, the first application of this new technology was to business and government. Money was printed; laws were encoded; treaties were signed. In response to these needs, a reading and writing industry grew up. Within a few years it was able to offer a broad range of reading and writing services to its customers. The customers found this to be a convenient arrangement, since hiring readers and writers from service vendors eliminated the need for each customer to invest in an expensive R&D effort of its own. The customers remained illiterate.

At first the situation was somewhat chaotic. Each vendor of reading and writing service tended to develop its own favorite language and its own technique for encoding information, leading to incompatibilities that impeded the spread of the new technology. After a winnowing-out period, however, the number of competing systems settled down to a few and major difficulties were handled by translators—though inevitably something seemed to be lost in the process.

Always looking for new markets, the vendors of reading and writing service began to examine the area of education. In view of its elitist role in the society it had been dismissed at first as too limited a market. A few, more imaginative people, however, argued that the application of reading and writing technology could turn education into a mass market. They proposed the following plan of attack. Reading and writing specialists and master teachers would work as a team. The master teachers would deliver their best, most carefully prepared lectures to the reading and writing experts, who would write them carefully *verbatim* into books. The books would then be copied many times, and each copy would be made available to a new type of educational functionary—the *reader*. His only job would be to assemble groups of students and to read aloud to them the recorded lectures of the master teachers. In view of the fact that training such a reader would be far less expensive than the education of a master teacher, the on-going cost of such a program would be far less than that of the conventional lecture method. The new method came to be called Writing Assisted Instruction, frequently abbreviated to WAI.

Needless to say, WAI had its opponents. Established master teachers expressed doubt whether a less skilled reader would be able to communicate subtleties of inflection, and they were certain that a mere reader could not process student responses with skill or intelligence. WAI proponents counter-charged that the master teachers were merely expressing their vested interest in the present educational establishment, and, indeed, that they ought to be fearful because the superiority of WAI would ultimately drive out the conventional practitioners. Even within the education establishment some younger members became WAI supporters on the grounds that the new method was a boon to education research. Until then, teaching had

Reprinted from Proceedings of Spring Joint Computer Conference, 1972.

been something of a black art, shrouded in the privacy of the classroom. To compare one teacher with another was impossible. But in the future, they said, the written record of the lectures of master teachers would make the teaching experience explicit and subject to analysis, comparison and improvement. It was high time, the young Turks exclaimed, that the teaching profession act with accountability to the public it served.

Unfortunately, such controversy remained for many years on a hypothetical plane. The number of actual WAI efforts was very small and their results were not striking. There was also a credibility problem. Many of the most outspoken advocates of WAI, especially in the legislature and in business and on local school boards, were themselves almost totally illiterate in the new reading and writing skills. How could they evaluate a new technology if they had not mastered it themselves?

Finally, government, business and some members of the education establishment decided to mount two or three large-scale demonstrations of WAI in order to show publicly the advantages of the new educational technology. For a period of several years curriculum experts collected information on a few key courses of lectures by assorted master teachers. The reading and writing experts wrote down the best series and read them aloud to the curriculum experts, who would criticize them and make improvements. The reading and writing experts would then incorporate the improvements in the next draft. Then came the field test. Readers began to read the drafts aloud to actual classes of students, and this led to further revision by the curriculum experts and rewriting by the reading and writing experts. At the end of a few more years a summative evaluation of the projects was undertaken by an independent, reputable educational testing organization, whose mission was to compare the cost and effectiveness of WAI with conventional education.

The parable is nearing its conclusion now. Actually it has two alternate endings, one happy and one sad. The sad ending, which follows now, is brief.

The educational testing organization reported that the projects were a complete vindication of Writing Assisted Instruction. It found that students taught by WAI performed even better on standardized tests than students taught by the average master teacher, that the students liked WAI better, and that the total cost of WAI was about a fourth that of conventional instruction. These pilot projects were imitated on a grand scale and education was revolutionized. Special institutes turned out vast numbers of readers and within ten years they were reading courses of lectures aloud to masses of people who could never have been educated before the new instructional technology arrived. The nation grew and prospered and thanked the day that the reading and writing industry was founded.

That is the sad ending. The happy ending is somewhat longer and more complicated. Here it is:

The educational testing organization found that WAI was neither measurably worse than conventional instruction, nor better. It found that costs were somewhat higher than anticipated, mainly because the market demand for people with reading and writing skills had driven their wages up near those of master teachers.

But this lukewarm finding was anticlimactic when it came, for the impact of reading and writing on education had taken a new turn during the intervening years. Here is how it happened.

At first a few master teachers had themselves found it necessary in pursuing their own research to spend the enormous effort required to master the skills of reading and writing. As they became more and more competent readers and writers, they began to see clearly the power of the written word within their own disciplines. Naturally enough the humanists were the first to apply this new intellectual tool to their fields of interest. Literature specialists collected stories, wrote them down, exchanged them with each other and began to develop literary criticism to a new height. Language specialists compiled lists of grammatical rules, which became writing manuals. Scientists were slower in becoming literate, with mathematicians leading the way, since they grasped the possibility of writing mathematical concepts in abstract notation. Nevertheless, for many years scientists continued to remain in verbal darkness.

While reading and writing had its primary impact on scholarly research, at the same time many master teachers across the land began to wonder whether it might not be beneficial to introduce elementary uses of reading and writing to students in their courses. A few language teachers began to show students how to write phrases and sentences, and the more venturesome teachers even asked students to write sentences of their own. Such experience, they claimed, greatly enhanced a student's understanding of syntax and rules of grammar. Even in subject areas far removed from language, to which reading and writing have a natural affinity, teachers began to report pedagogical gains due to having students carry out elementary reading and writing tasks as an adjunct to conventional instruction.

One obstacle to student use of reading and writing was the awkwardness of the main systems of notation, which had been developed mainly for research and business applications. The most popular such system was particularly difficult to format, since its characters all had to be positioned accurately in a fixed number of columns. Occasionally there were rumors that a group of teachers in a remote province near the northern frontier had developed a simpler writing system and all their students were using it daily. Such rumors were hard to verify; only a few people ever voyaged that far north, and, in any case, experts in the reading and writing industry seemed confident that anything that made the current system simpler would also take away its power and elegance. So most teachers adhered to it.

Within a few years teachers began to hold national meetings to tell one another how their students used reading and writing within their courses. Advocates of this type of use, which came to be called *adjunctive*, insisted that it be distinguished clearly from WAI. Writing Assisted Instruction, they charged, was nothing more than an improvement in the technology of delivering instruction. Adjunctive use of reading and writing by the student, on the other hand, represented a change in the intellectual content of instruction. They argued from the following philosophical premise:

> Reading and writing constitute a new and fundamental intellectual resource. To use that resource as a mere delivery system for instruction, but not to give a student instruction in how he might use the resource himself, was the chief failure of the WAI effort, they said. What a loss of opportunity, they exclaimed, if the skill of reading and writing were to be harnessed for the purpose of turning out masses of students who were unable to read and write!

WAI advocates responded that it was well and good that a few elitist schools teach their students the difficult skill of reading and writing; it was enough that WAI teach lesser skills to masses that might otherwise remain uneducated and unemployable.

How much longer, asked the WAI opponents in rebuttal, will an illiterate person be considered educated? How long will he be employable and for what jobs if elitist schools are turning out competent readers and writers by the hundreds?

The more visionary advocates of mass literacy told of foreseeing the day when students would spend more hours of the day reading and writing than listening to lectures. Small research libraries had indeed sprung up at some schools, but they were expensive operations limited to a few specialists who had to raise funds to pay for their use. Such people were particularly incredulous at the suggestion that every school ought to adopt as an educational goal the establishment of a significant library open freely to all students. School administrators were at first appalled at the idea that the library should not be on a pay-as-you-go basis but should be budgeted as part of the general institutional overhead costs.

But as time went on and even school administrators became competent and imaginative users of the skill of reading and writing, all schools gradually accepted as a mission the bringing of literacy to all students. Accreditation agencies examined the quality of libraries before approving schools. Books began to appear all over and finally even in people's homes. WAI did not die out altogether, but continued as a cost-effective alternative to the lecture. But as books reduced dependence on lectures, students made less use of both WAI and lectures and spent more time on their own reading and writing projects. The nation grew and prospered and wrote poems in praise of the day that reading and writing were discovered and made available to all people.

End of parable.

It is a perilous strategy, bordering on bad taste, to tell a joke and then for several pages explain why it was supposed to be funny. However, this allegorical tale has been told here not merely for entertainment but mainly for the moral lesson it carries. To compare reading and writing with computing might be dismissed as an amusing frivolity; but that would be wrong. Our fundamental philosophical premise here is that, like reading and writing,

> "[computing] constitutes a new and fundamental intellectual resource. To use that resource as a mere delivery system for instruction, but not to give a student instruction in how he might use the resource himself, has been the chief failure of the [C]AI effort. What a loss of opportunity if the skill of [computing] were to be harnessed for the purpose of turning out masses of students who were unable to [use computing]!"

As this example shows, it is a trivial editing task to go through the entire reading and writing fable and turn it into a story about computing and its uses in education. In fairness, the author admits that the story really *is* about computing and that reverse editing was done in the original telling so that it would seem to be about reading and writing. Yet, as a story about reading and writing it has considerable plausibility, doesn't it? The Writing Assisted Instruction program outlined in the story is not a totally absurd idea for putting reading and writing to use in education. One cannot argue against claims that committing lectures to writing would make education available to more people, would invite critical comparisons and a consequent improvement in subsequent revisions of written materials, and would be an asset to the study of the learning process itself. What does appear absurd, however, is the failure of these mythical WAI proponents to recognize that the best educational use of reading and writing is the teaching of reading and writing itself to everyone. Mass literacy is an educational mission about which few of us have doubts today.

Yet that consensus among us seems to vanish when one substitutes "computing" for "reading and writing" and "CAI" for "WAI". Mass computing literacy is not an agreed-upon educational goal. Today very few courses at any educational level show students how to use computing as an intellectual tool with applications to the subject matter being taught. Oh, there are a few isolated, subject-matter-free courses in computer programming; but their market is largely restricted to vocational-education students, at one end of the spectrum, and future computer professionals at the other. It is true that most schools consider it prestigious to have a large and powerful computer facility; but the fact of the matter is that such computers are usually the captives of research and administrative interests and operate on a pay-as-you-go basis. Ironically, it is in the most prestigious universities that students are least likely to be permitted to use those prestigious computers. It is a rare secondary school, college, or university that budgets and operates its computer facility in the same way that it budgets and operates its library. (There is a persistent rumor of an exceptional example in some remote province near the northern frontier, but so few people ever travel that way that the report is hard to verify.) In the main, literacy in computing simply is not an educational goal at many schools. Most educators seem to find bizarre the suggestion that accreditation agencies examine schools for the quality of their educational computing facilities, just as they now do with libraries.

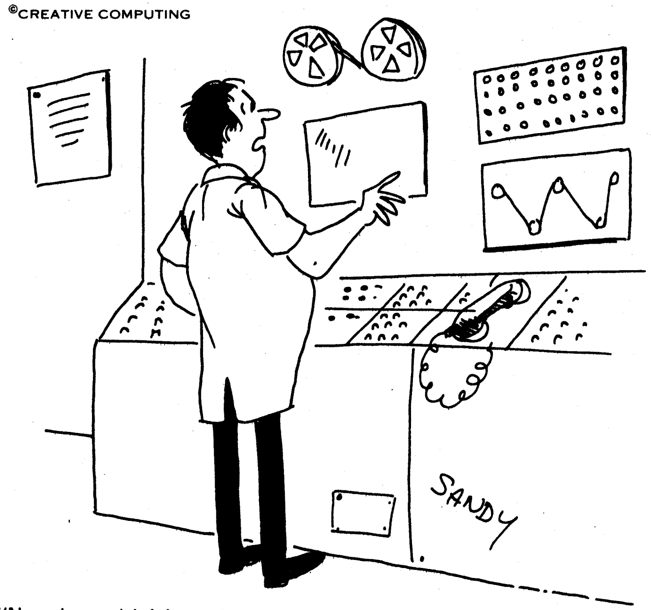
The distressing truth today is that educators, local school boards and federal policy-makers are far more receptive to the plans of CAI proponents for using the technology of computing as a cost-effective delivery system for instruction in math or remedial English than they are to making computing itself a part of education. This statement should not be taken as a blast against CAI. On the contrary, CAI advocates are to be commended for their desire to reduce the cost of instruction, to tailor it to the different learning styles of students, to develop systems that encourage closer examination of what is being taught and systems for improving instruction, and to hold teachers and schools accountable to their clientele. With enough developmental work on CAI, it is likely that students will perceive the computer as a very superior teacher. Above all, CAI promises to make education a less labor-intensive industry and so to enable masses of people to become better educated. This is certainly a goal worth working for.

But there is a higher goal. If the computer is so powerful a resource that it can be programmed to simulate the instructional process, shouldn't we be teaching our students mastery of this powerful intellectual tool? Is it enough that a student be the subject of computer administered instruction—the end-user of a new technology? Or should his education also include learning to use the computer (1) to get information in the social sciences from a large data-base inquiry system, or (2) to simulate an ecological system, or (3) to solve problems by using algorithms, or (4) to acquire laboratory data and analyze it, or (5) to represent textual information for editing and analysis, or (6) to represent musical information for analysis, or (7) to create and process graphical information? These uses of computers in education cause students to become masters of computing, not merely its subjects.

It will be countered that such an educational mission is well and good for a few elitist schools, where students are willing to learn the difficult skill of computing; but it is enough that CAI teach lesser skills to masses of students that might otherwise remain uneducated and unemployable.

In response we ask, how much longer will a computer illiterate be considered educated? How long will he be employable and for what jobs if elitist schools are turning out competent computer users by the thousands?

The true story about computing and education is at its midpoint. Like the reading and writing parable, it has a sad ending and a happy ending. Which one actually occurs will be determined by you—teachers, school administrators, computer professionals, and government policy-makers.



"Now hear this! *I* am the programmer. You are the *programee*!

THE ART OF EDUCATION: BLUEPRINT FOR A RENAISSANCE

Thomas A. Dwyer
Project Solo
University of Pittsburgh
Pittsburgh, PA 15260

Crises come and go in public education, and our present times are no exception. The statistics about student and teacher dropouts (not to mention violence and senseless destruction of property) are known well enough, but the deeper losses signalled by these symptoms have been obscured by other pressures in society today. We desperately need the insights of careful and calm thought if the opportunities made possible by years of hard work are not to escape our grasp.

A starting point is to recognize that the real tragedy to fear is the waste of those human qualities that are cultivatable only in the fleeting years of childhood, qualities without which the wisdom of mature years will never flourish. An educational system is important only if it is literally a cradle of wisdom; all other issues—personally satisfying careers, new knowledge, even the fate of civilization itself—flow from this premise.

But wisdom is an intangible and idealistic concept while educational systems are tangible and nitty-gritty realities. How can there ever be a lasting union of these two worlds? I believe that part of the secret to managing this challenge is to view education as an advanced art. Because it is the most demanding of arts, it admittedly needs the modern tools of science, business, and *especially* technology. In fact, it is only the balance of an artistic viewpoint that will extract real substance from these other advances.

The Magic of Art

Walking through an art museum produces an exhilaration that is hard to explain. The exhibits are static, and mostly from places and times foreign to the visitor. What kind of magic is it that can transcend these barriers and touch us so deeply? I think it is the awesome realization that we are seeing one common world interpreted in as many incredibly different ways as there are artists.

I propose that the way to bring a Renaissance to education is to view its global character as homomorphic to this dynamic "one-world, many-understandings" lesson from art. I believe that an educational system will become a cradle of wisdom when it learns to build genuine artistic diversity, based on common experience, controlled by discipline willingly embraced because there is real purpose.[1]

How to manage this challenge is of course a tough question. Remarkably, it is answered in part by again returning to the world of art. To see why, consider the following "thought" experiment.

The Magic of Technology

A phonograph is placed in a room with a small group of listeners. A well-made recording of a Verdi opera is placed on the turntable, and for an hour or so the listeners concentrate on everything they hear. We then ask the question: What have the listeners received?

The answer is that the amount, subtlety, and utility of information received by each person is *radically* dependent on the history of experiences (or "cultural background") each person brings to the listening room. A listener raised in Napoli within a community that lived and breathed opera, will actually "hear" things that are beyond comprehension for other listeners. Even more startling will be the depth of information absorbed by a listener who has been a creative worker in the field of opera, whether at the composing, performing, or production levels. At the other extreme, a listener who has had no part of the operatic culture, will literally be hearing mostly "noise." *Efforts aimed at perfecting either the record or record player will not substantially change this situation.*

Our imagined experiment reveals three fundamental approaches to bringing about learning, and makes clear the remarkable catalytic interactions they can have on one another. These approaches (or techniques) can be described in terms of the adjectives "transmittal", "experiential", and "creative".

No knowledge is really transmitted; it must all be created.

Transmittal techniques are those that attempt to pass on ideas, facts, skills, etc. from person to person. As our experiment suggests, this information can be both limited and cryptic, and heavily immersed in "noise." But when we add a human receiver to the system, there now arises the possibility of retrieving, reconstructing, and even creating content from the original noisy signal, provided the appropriate *experiential* and *creative* faculties of that human listener have been enabled.

We have been using this model at the Soloworks[2] lab to guide our use of technology as both an art and a craft. In particular, we have been investigating the potential of computer-based technology for providing invigorating sets of experiences that will sensitize students as expert "receivers" in a laboratory-based mathematics curriculum. We have also used this same technology to support students in creative work, so that they will not only get maximum benefit from the transmittal elements of the curriculum, but will themselves eventually contribute to the growing body of knowledge from which transmittal mode draws.

To extend these ideas from a laboratory level to the difficult arena of public education, more than "advanced technology" is needed. We must also work at defining "advanced goals" that stimulate new thinking and new dedication. We must develop models that clarify the role of technology within the "advanced art" needed to support such goals. Finally, we must address some very practical questions, and ask what "advanced crafts" are needed to implement these goals, what they will cost, and what alternate or redundant branches are needed in our plan to assure high reliability and success. Let me go into some detail.

Advanced Goals

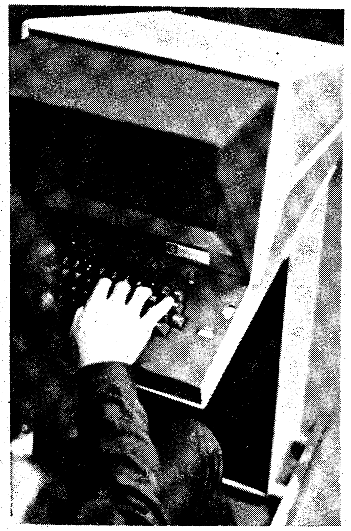
The question is sometimes asked "how do you decide which things the student is to invent (or re-invent), and which things are to be accepted on the say-so of teachers and books—after all, time is limited, and no one can re-invent all knowledge." One flaw in this rhetorical argument is the assumption that acquiring knowledge from others is time-efficient, and that therefore the main job of education is to accelerate and refine the transmittal process.

A more imaginative (and I believe a much more productive) position is that *no* knowledge is really transmitted; it must all be created. This view starts with the assumption that knowledge is a medley of many components and relations, and that the resultant "whole" is determined by schemata unique to each learner. It therefore argues that factual data given to a learner through "instruction" doesn't become knowledge until it has somehow or other been fitted to these schemata. This means that instruction makes sense only when it recognizes the existence of an internal representation/transformation system unique to each person. The example of how instructors of blind students learn to respect the internalizations of others shows how this theoretical view can translate into very practical pedagogical methods.[3]

The revolutionary goal that follows from this stance is to design a school where the students assume from the beginning that their task is to invent all knowledge. A good way to clarify what this goal means is to immediately address two obvious questions:

(1) Does this goal rule out teachers, books, lectures, television, films, or CAI? Definitely not. What it does is revolutionize the use of these "transmittal" elements, and give each of them revolutionary goals of their own. As the main goal suggests, there is a radical difference in the learning of two students attending the same lecture when one student views his task as ingestion followed by regurgitation,[4] while the other has the goal of appropriating (or rejecting) ideas for either present or future use in personal invention.[5]

(2) Does this goal imply abandonment of what is usually called a curriculum structure? No, but it does imply very different approach to the design and use of such a structure. Our successful use of a "top down" approach to curriculum writing at Soloworks illustrates one way in which the design process can take on very new dimensions.[6]



The most fascinating place to be is at the podium, not in the audience; this is a tremendously important point.

I believe that an important sub-goal of the educational setting being recommended would be to invent techniques that transmit certain major ideas almost instantaneously. For example, we once wrote (with great difficulty) a rather long, cleverly illustrated explanation of what time-sharing really meant to a computer user. Now all we have to do is let these users hear music played as one job on our time-sharing system, and they immediately know more about both the qualitative and quantitative aspects of time-sharing than we were ever able to describe in writing. By linking this demonstration to remembered ideas about the regularity and timing of music, we have made the new idea of time-sharing completely transparent.

A meta-goal that comes out of this sub-goal is to involve the students in creating similarly radical "teaching" techniques. On a more general level, this meta-goal translates into a concern for, and analysis of, learning itself, but by the students as well as the educators.

Education as an Advanced Art; Relation to Technology

In setting the advanced goals that have just been described, I have tried to combine imagination with insights that come out of laboratory experience. I have also tried to set goals that are theoretically realizable. Now I must address the question of what overall architecture is needed to support this realization.

The model proposed is not describable with formulas or flowcharts. Its structure is suggested much more by words like "culture" and "community," while its realization is dependent on the ideas of "orchestration" and "adaptability." Its inner workings spring from a concern for honesty, discipline, and responsibility, while the dominant character it seeks is one of enthusiasm, friendliness, and humor.

This list of descriptors is not as Utopian as it may seem. It suggests in fact the kind of total image we attach to the workings of serious art. For this reason I suggest that the process of developing and refining an architecture for education be viewed as an advanced art, rather than a science, business, or social service.

Let me further illustrate the power of viewing education as an advanced art by focusing on one of these descriptors ("orchestration"), and then applying it to the specific problem of selecting and using technology in education.

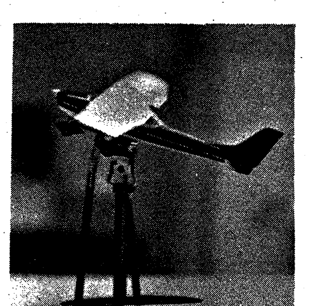
The diversity of views on how to use technology (computers in particular) for education can be very confusing. These views range from single-minded advocacy of computers as automated drill, practice, and "tutoring" systems, to the belief that the real payoff will come when every child has a personal computer to be used as a kind of "supertoy."

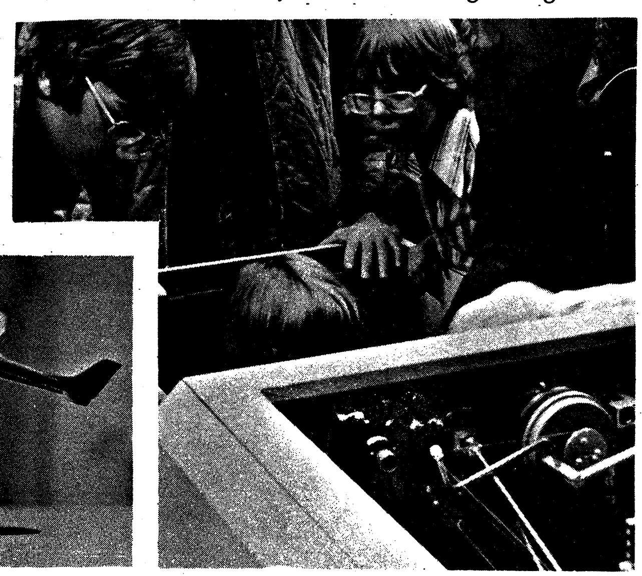
The idea of orchestration helps sort things out by reminding us of the advantages in a rich palette of tonal colors. It suggests that new textures can be created, and new dimensions explored, by exploiting differences, not sameness. (It also says that the most fascinating place to be is at the podium, not in the audience; this is a tremendously important point.)

To introduce a somewhat more abstract but (at least for mathematicians) more extensible terminology, we can say that the power of the orchestration concept flows from the idea of orthogonality of components. In a way this is an unexpected result; it says that global unity comes out of local diversity. It argues that new dimensions are possible precisely because our new technological tools do *not* all point in the same direction, and because they are *not* all hardware oriented. It is a result with implications as profound for education as was the discovery of the role of independent but rich basis elements in structuring extraordinarily imaginative spaces for mathematics.

This abstract idea translates nicely into practice. At one level, we have found it useful to think of three orthogonal *classes* of tools described by the words transmittal, experiential, and creative (recall our phonograph experiment). For example, a CAI lesson belongs in the first class, an interactive simulation in the second, and a debugging session or synthesis project in the third. It is also useful to distinguish orthogonalities within classes. For example, exploiting the contrasts between transmittal elements such as CAI sessions, books, films, and lectures is much more effective than trying to make them equivalent, pointing all in the same direction. There is much more payoff in building multi-dimensional systems from elements that contain different intrinsic perspectives.

Past educational systems have been denied such hyper-dimensionality. What the recent developments of computer technology (and computer science) now present to us is a large set of non-trivial orthogonal basis elements. We must of course continue to enrich this set. But we are also ready to begin work on another enterprise, namely the art of creating new and complex "forms" that generate imaginative educational systems from this growing basis. It is





If educational administrators run unimaginative educational shops, it's because they must spend most of their time responding to anti-imaginative pressures.

no exaggeration to assert that the rewards of such efforts will be structures with elegances considerably greater than those found in any of their parts.

Education as a Craft

We now come to the problem of translating artistic ideals into reliable practice in an imperfect world. We must now ask what policies, skills, techniques, materials, tools, and craftsmen are needed to successfully put together and maintain the kind of educational enterprise we have described.

Particular attention needs to be given to developing new ideas about educational administration, especially the "killer demands" of overhead. If educational administrators run unimaginative educational shops, it's because they must spend most of their time responding to anti-imaginative pressures. Very few of them are given the chance to break out of this pattern. It's definitely the squeaking wheel that seems to grab most of their attention. As a result, fresh ideas are seldom supported, and initiative soon dies of loneliness. The solution to this problem is not to ignore administration, but to design new administrative climates. Making distinctions between "creative", "logistical", and "fiscal" administration will be an important first step.

There is not space to discuss the other crafts needed except in summary form. However it is worth reporting that a number of us working with these ideas have found that many of the needed talents are best developed "in-house." The idea of using older students and alumni as part of a first-rate staff works very well in practice, especially when there are good teachers around who know how to energize young talent.

Summary

Our experience at Soloworks indicates that the learning phenomena we (and others doing similar work) have observed in settings of the type described in this paper are extensible, workable, and applicable. We believe that this experience can be applied to new educational structures that have been explicitly engineered from the ground up as advanced artistic enterprises.

Some of the elements that we see as essential to such an undertaking are the following:

1. A Set of Advanced Goals
2. An architecture based on the idea of education as a complex art made possible by new ideas, theories, strategies, and technologies.
3. A view of education that sees more power in the ideas of community and culture, than in the methodologies of business or science.
4. Craftsmen with complementary skills, including teachers who like to teach, all kinds of students, theoreticians, engineers, and imaginative administrators.
5. An extensive collection of the orthogonal materials needed to support an adaptive curriculum.
6. Advanced tools, especially those related to the general purpose computer, and post 1970 man-machine interfaces.
7. An administrative sub-structure that fosters initiative, controls unproductive overhead, and encourages continued experimentation.
8. A built in proof-of-performance mechanism which gives constant attention to good communication with others through use of imaginative media.
9. Most important of all, the recognition that good art is the product of singular devotion. A great deal of attention should be given to mechanisms that make it impossible for vested interests, or committee-type compromise and mediocrity to ever settle in.

While we have some specific ideas on the forms such structures might take, a true educational renaissance will be possible only when a multitude of "artistic" views are brought to bear. We have received many letters at Soloworks proving that there are lots of such good views, representing lots of good people, and we continue to invite this feedback. We'll try to synthesize as many of these views as possible in our 1976 final report on Soloworks.

[1] A complementary view is found in Mark Van Doren's book "LIBERAL EDUCATION" where he urges that the work of education "be done as artists do things, with skill and thorough care, and with a reverence not hostile to high spirits."

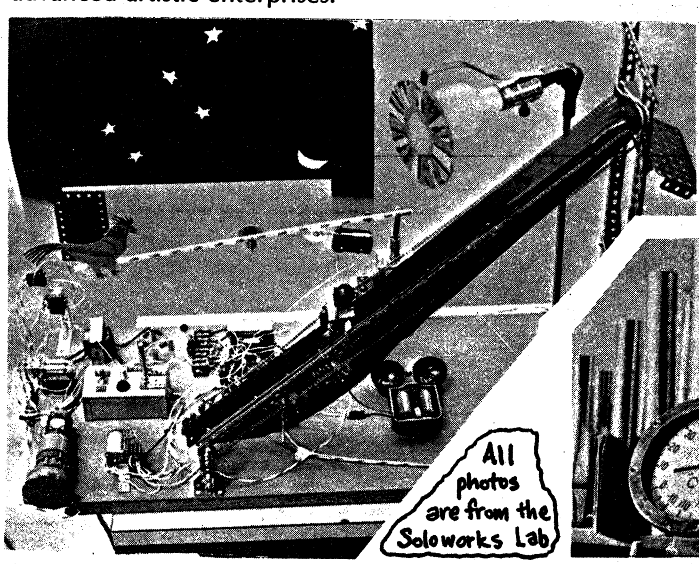
[2] Soloworks is the informal name of an NSF project entitled "A Computer-Based Laboratory for High School Mathematics."

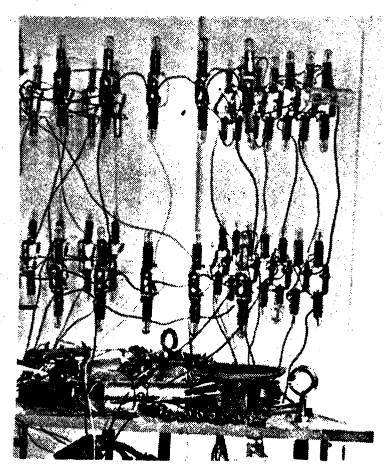
[3] For further detail, see page 142 of T. Dwyer "Heuristic Strategies for Using Computers to Enrich Education" (Soloworks Newsletter #26).

[4] That most students view this as their function is easily proved by listening to student conversations in the hall after an exam. "What did you give for #4?," "I don't think that's what he wanted," and "I'm glad that's over" are common and revealing reactions.

[5] To illustrate by example we cite the adult who still (35 years later) remembers *every single word* of a German translation of a silly song because it was created by a class plot to have each student interrupt the lesson every few minutes and ask the teacher for the German equivalent of successively needed words.

[6] For further detail, see "The Significance of Solo-Mode Learning for Curriculum Design", Soloworks Newsletter #33.





Computing At The University of Texas

Predictions that computers someday would replace teachers in the classroom have long been made, but replacing teachers has not been the goal of computer-based education at The University of Texas.

In the words of Dr. George Culp, coordinator of instructional computing for the UT Computation Center, computer-based instruction is directed toward "relieving the instructor of routine work and freeing him for more effective contact with students."

Instructional computing has been done for several years on UT's CDC 6600/6400 SYSTEM, one of the most powerful computers in the world. The Computation Center also has a NOVA 840 which performs some of the instructional programs, and the center recently has added a DEC-10 system which is especially well-suited for instructional computing.

The DEC-10 gives priority to classroom instructional users and its BASIC offers extensions and instructional features beyond those available on the CDC 6600/6400 system.

Instructional computing is basically interactive, meaning that the student actually 'talks' to the computer through a typewriter device, and the computer responds.

For example, if the computer asks a question, the student responds and then is told whether his answer is correct. Programs can be set up in a number of ways, but essentially interactive computing means that the user receives immediate response which gives students an idea about their progress, as well as reinforcing learning.

UT's extensive use of the computer in the classroom received a substantial boost from a four-year National Science Foundation-sponsored project known as Project C-BE (computer-based education).

The intent of Project C-BE was to apply computers to the teaching of science and engineering courses, but the success of the project's application to other areas is attested to by the fact that 27 courses at UT, ranging from chemistry to English, now are taught with the aid of computers.

"We would like to expand the use of the computer for instruction," says Dr. Culp, "and with the addition of the DEC-10 to our facility, we have the capability to do so."

He adds:

"One of the problems we have to overcome is the reluctance a lot of people have to using the computer, which they perceive to be impersonal."

In truth, the computer can be used to assist the instructor in achieving closer personal contact with students.

"We are aware of the problem, and do everything we can to overcome it, but our best spokesmen are the instructors on campus who are using interactive computing as part of their teaching."

Dr. J. J. Lagowski, professor of chemistry, and Dr. John Allan, associate professor of mechanical engineering, were co-directors of Project C-BE. Dr. Lagowski says he first began to consider the use of the computer as an instructional aid in 1965 when he faced the task of coordinating chemistry courses for more than 3,000 freshmen.

"A lot of information is lost in transmitting knowledge from professor to graduate student and then to student," Dr. Lagowski says. "My aim in developing computer instruction was to forestall this loss of information, not to replace the instructor."

With the aid of computers, teachers can be more flexible in their use of time while teaching large classes, he explains.

"Before, I was doing the work in the classroom that students could have been doing on their own and learning better than could be taught in a lecture," he explains. "I am now able to present the great schemes of chemistry and nature in my lectures. Students are receiving the benefits of what machines do best and what people do best."

"I am completely sold on the concept," he adds. "We have the data to prove that students do better than they did before the computer became part of our program."

Dr. Walt Reed, assistant professor of mechanical engineering, is using instructional computing to teach kinematics, which is the study of the motion of mechanical devices.

In the course of learning kinematics, beginning engineering students also learn the basics of computing which they will use throughout their careers.

Students in Dr. Reed's course use a television-like screen to draw machine designs with an electronic "light pen." They describe the motions they wish their design to make by using symbols.

"What appears on the screen is exactly what would occur if the student built the device in the workshop," says Dr. Reed, "but the computer can show the student more than he could see if he had actually built the machine."

The computer, Dr. Reed explains, is used to do things that otherwise could not be done in the course.

"One of our biggest problems in engineering education is that we lose a lot of good potential engineers because of the drudgery they face mastering the necessary basics," says Dr. Reed. "They are anxious to perform actual engineering tasks, and the computer makes this possible."

In the College of Business Administration, Dr. Joel Stutz, assistant professor of computer science, is teaching

Pat Caroom (left), manager of the DEC-10 system at The University of Texas at Austin, and Judith West, programmer, design software for the system dedicated to interactive instructional computing at UT. Elissa Vogel (at console) is a system operator.

beginning statistics which also introduces business students to basic computing.

"The goal in my course is to teach students enough about computing so that they will be able to communicate with specialists as they will be doing throughout their careers," Dr. Stutz says.

The Graduate School of Library Science is employing the computer to prepare students for their careers as professional librarians in a field which is becoming increasingly computerized.

"Our students are mostly inexperienced with the computer," says Dr. Ron Wyllys, assistant professor of library science, "but we are confident that the experience they gain here, both in computer basics and the actual use of computers as they are used in libraries, prepares them for their careers."

Computers are being used now in courses and subject areas that once were thought to be impossible to translate into terms that could be used by computers.

Dr. Susan Wittig, assistant professor of English, uses the computer to teach freshman English.

The computer cannot make subjective judgments about the quality of a student's work, but many freshmen are deficient in the basic skills of writing and communicating. In Dr. Wittig's classes the computer is used to teach and reinforce the basic mechanics of English grammar.

Dr. Culp says that the possibilities of instructional computing are just beginning to be realized.

"With the addition of the DEC-10 we have increased the services the Computation Center can offer to the faculty who wish to use computer-aided instruction," he points out.

"Among the features of the DEC-10 are the ability of the user to switch to a calculation mode during an instruction program," he explains. "That is important in many types of applications that we could not handle before."

Another advantage of the DEC-10 is its record-keeping capability. The instructor can find out immediately such information as how many students have completed a lesson or look at the performance record of an individual student. Another advantage is the upper case/lower case printer which has made classroom material more readable as well as providing a means to add emphasis on terminal output.

"Many of our users have requested the capabilities we now have," says Ms. Pat Caroom, manager of the DEC-10 and NOVA systems for the Computation Center, "and we are ready to help potential users develop any special programs they might need."

Dr. Culp points out the Computation Center is mainly a service facility, designed to help researchers and teachers use the computer to its full capacity.

Under the direction of Dr. Charles Warlick, the UT Computation Center is among the most diverse and modern academic centers in the country.

This spring the center is conducting a series of seminars to familiarize users and potential users with the capabilities of the DEC-10.

"We encourage everyone, whether they have any knowledge or prior experience with the computer, to call on our staff any time they have questions or ideas," Dr. Culp says, "and we will do our best to meet their specific needs for instructional computing use."

If the growth of computing in the recent past is any indication of future trends, instructional computing will have a large impact on higher education, and the UT Computation Center will play a large role in creating that future.

"The best way to have a good idea is to have lots of ideas."

Linus Pauling

COMPUTERS IN SECONDARY SCHOOLS—1975

A second study funded by the NSF on the use of computers in secondary education was completed in the fall of 1975 by the American Institutes for Research (The first study was done by the AIR in 1970). Principal investigators were William Bukoshi and Arthur Korotkin. The sample consisted of 25% of U.S. School Districts; about 65% of these responded.

Major conclusions of the study were as follows:

- Since 1970 computing in secondary education has steadily increased with 58.2 percent of the schools that responded to the survey indicating they are currently using a computer for administrative and/or instructional purposes (34.4 percent in 1970).
- The trend is toward more fully using the computer. Of schools using computers, only those using them for both administrative *and* instructional uses *increased* from 1970 to 1975 (26.2 percent to 37.5 percent). The percentage of schools using computers for only administrative *or* only instructional purposes dropped from 1970 to 1975 (62.5 percent to 54.1 percent for administration; 11.3 percent to 8.4 percent for instructional).
- Given the findings concerning the growth of secondary school computing for the last five years (1970-1975), and with the assumption that the current rate of adoption of computer technology in the schools (4.8 percent/year) will continue, it can be projected that within the next decade every secondary school in the country will have access to a computer system for some type of administrative and/or instructional application.
- Respondents indicated that using the computer as a "problem-solving tool" and as a subject area for "computer science" courses were the most frequently utilized instructional applications in secondary education.
- In schools using computers CAI has increased from 8.4 percent in 1970 to 13.8 percent in 1975.
- The predominant instructional use of computers in 1975 is still for mathematics.
- With regard to administration the most frequent uses of the computer are for student accounting and resource management.
- The BASIC language has become the predominant computer language for instructional computing.
- Schools using computers tended to be larger than non-user schools (median number of students 1000 versus 400). The size of the user schools, however, is smaller than in 1970, when the median number of students was 1347.
- The current survey indicates that over 90 percent of the funding for educational computing at the secondary school level comes from local and state sources.
- Despite the growth in computing activities, there was virtually no change since 1970 in the relative amount of the operating budget spent for instructional computing ($0.18 per $100 of school expenditures in 1975 versus $0.17 in 1970).

COMPYOUTER FAIR

Wes Thomas*

Hello - Scott.

"Who's that?"

It's - me.

"Me who?"

I - am - a - Wang - 2200 - computer - interfaced - to - a - Votrax - audio - synthesizer.

I - want - to - show - you - how - I - can - help - kids - learn. . .

"I'm scared. Will it hurt?"

Do - not - be - afraid. - Let's - first - play - a - game.

"Oh, goody, I love games."

I - am - thinking - of - a - number - from - one - to - one - hundred. - Try - to - guess - the - number - and - I - will - tell - you - if - your - guess - is - low - or - high. - You - have - only - six - guesses - until - you - lose. - What - is - your - guess,-please.

"Ah, one hundred and two."

You - are - not - thinking. -Your - guess - is - very - high. -Try - again...

Scott is a dummy. But so what? He was the star of the show. I had wandered in from the exhibits at the New York Coliseum—part of the 1976 National Computer Conference—attracted by a sign that said COMPYOUTER FAIR. And here I was in the middle of the top 58 entries out of 350 submitted by students in the U.S., Canada, and U.S. schools in Europe. And I was talking to Michael Taylor, a 9th grade Lexington, Mass. student-ventriloquist-programmer who had created a "talking computer."

"The program was written in BASIC on the Wang 2200 computer interfaced with the Votrax speech synthesizer," Mike was explaining. "To make the Votrax talk, I used hex codes, which the Votrax understands as the different phonemes (sounds). It puts the phonemes together to form words, so the vocabulary is really unlimited."

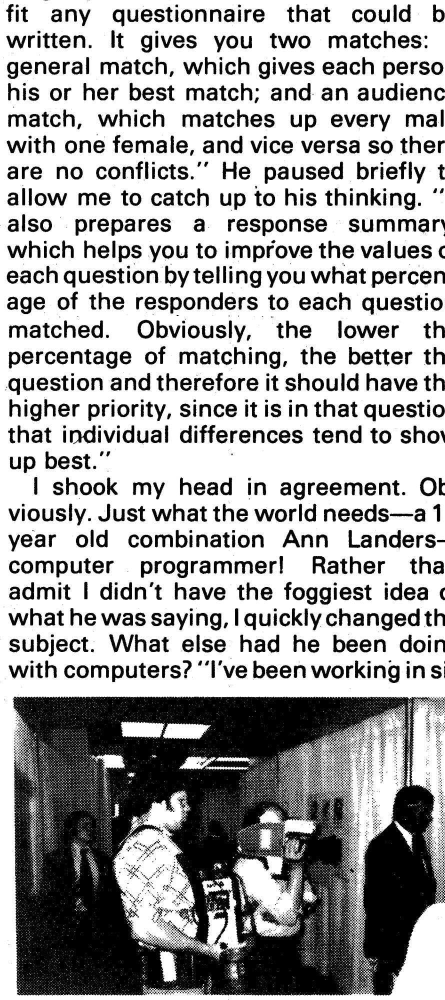
"One of the best things it could be used for is education, because it's a teacher that will never give up and its very persistent. It's also good for hospitals. Let's say, for example, someone is sick on the road. The doctor can call the computer at a hospital. The computer will give the person's history, and what medicine should not be given to him, and what should be..."

I thanked Mike and slipped out of the crowd pressing around me and headed toward a bank of teletype terminals. Seated behind one was Donald Abrams of New York City. He looked like he was

Michael Taylor, a 9th grader from Lexington, MA poses with his ventriloquist dummy next to his project, a voice synthesis unit connected to a Wang 2200.

about 8 years old, but turned out to be 13. I asked him what he was doing. "This is the Questionnaire Independent Dating System," he came back with, sounding vaguely like the Votrax. I guess he had been hearing it all day. "It is a dating system that does not rely on any one single questionnaire, but can be suited to fit any questionnaire that could be written. It gives you two matches: a general match, which gives each person his or her best match; and an audience match, which matches up every male with one female, and vice versa so there are no conflicts." He paused briefly to allow me to catch up to his thinking. "It also prepares a response summary, which helps you to improve the values of each question by telling you what percentage of the responders to each question matched. Obviously, the lower the percentage of matching, the better the question and therefore it should have the higher priority, since it is in that question that individual differences tend to show up best."

I shook my head in agreement. Obviously. Just what the world needs—a 13 year old combination Ann Landers—computer programmer! Rather than admit I didn't have the foggiest idea of what he was saying, I quickly changed the subject. What else had he been doing with computers? "I've been working in six

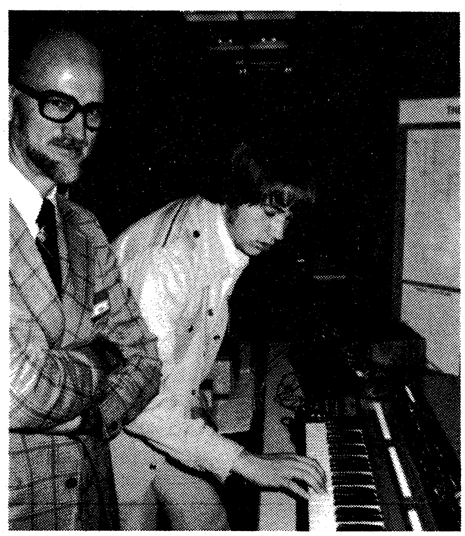


A CBS-TV crew videotaped Mike Taylor for a spot on the 6 pm news.

languages. I intend to modify the system at the school that I go to to make it a bit more like a higher-level computer." What system? "Our IBM 1130. I intend to modify the Disk Monitor System." Sorry I asked.

Or the Grand Prize winner, Walter Freitag, Jr., of Dresher, Pennsylvania, age 15, who developed a computer perdiction model for the spread of fire. Walter is a self-taught programmer (no computer classes are offered in his school). His father, a chemist, told him his project was impossible, but Walter did it anyway, using the Univac 1108 computer where his father works. The model, written in BASIC, uses a series of three-dimensional matrices to represent the spread of fire in a structure which includes the temperature at which the material ignites, time to "burn out," etc.

Walter believes the model can be developed to be useful for fighting fires and planning new buildings. Walter, who was a previous winner of 5 other science fairs, won an Altair 8800. He thought he might use it to further develop his model, or possibly lend to his school, or both.



William Blum, a hs senior from Huntington, NY demonstrates his digitally controlled electronic music synthesizer to Daniel McCracken, noted author and a Computer Fair judge.

I turned to Andrew Shooman, age 11, absorbed in modifying his own computer or something. A giant drawing over the terminal read "Computer Astronomy Almanac." What was that? "It consists of two programs—a planet program and a moon program." He sounded like Votrax, too. What was this, casting for "The Bionic Boy?" "The planet program gives the position of the planets for any day in

*Wes Thomas is editor of *Communications Tomorrow,* an occasional publication of the World Future Society; an interviewer for WBAI-FM in New York; a contributor to a variety of other interesting publications; and also toils regularly for Reeves Teletape Corp.

Wes Thomas interviewing Student Computer Fair participants.

1976 and 1977. And the moon program gives the phases of the moon for any date in the 20th century." The TTY was clunking out a series of numbers. "This tells the position of the planets. For instance, this month, Mars is in the Constellation Leo. If you look in the sky tonight, you will see Mars in the Constellation Leo. That makes it easier for astronomers to find." Astronomers that would like some help can contact Andrew at Glen Cove South School in Long Island.

Around the corner, Matthew Korn of Forest Hills, New York was explaining his stock analysis system to two businessmen avidly taking notes. Matthew told me his system throws new light on the stock market and can be seriously used. He would charge "5% of your earnings." Matthew, age 17, is looking for a summer job "at full executive salary, but I'm willing to go down a bit. I'm starting in Yale this September. That should be good for an extra $20,000." Prospective employers can contact his secretary for an interview at Bronx High School. He'll try to fit you in.

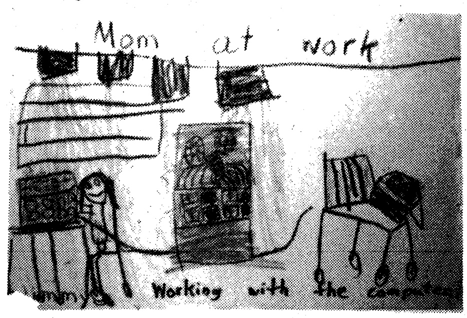
In the next exhibit, Abraham Lederman was playing his advanced monopoly game with a computer terminal, and elsewhere Glenn Sage of Portland, Oregon was toying with a computer game that prepares the player for an imminent

Leslie Heller, a 6th grader from Poughkeepsie, NY choreographed and performed a ballet based on APL symbols.

stock market crash by buying and selling stock for maximum profit before the crash. These three could be dangerous if they ever got together.

Wisely deciding not to make that suggestion, I joined a crowd gathering around a piano, where Stephen Basili, Grade 5 was playing "Computer Boogie," using different musical passages to represent different computer components ("... the card reader sounds like this..."). Maybe Stephen could set IBM documentation to music—maybe that would help...

Later, I listened to performances by several other programmer-musicians, including Glenn Poole of Springfield, Virginia, who used "probabilistic mathematical techniques" on a computer

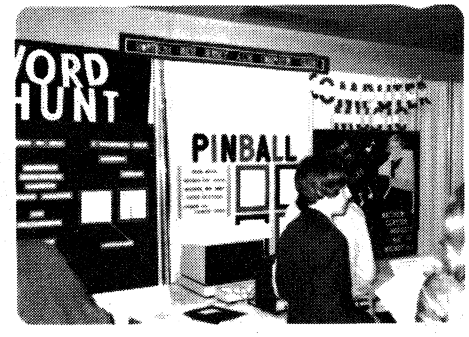


Entry in the poster category at the National Student Computer Fair.

to compose music; David Shmoys of Huntington Station, New York, who uses a computer to automatically transform Telemann flute compositions; Bruce Horn, Palo Alto, California, who uses the SMALLTALK language to plot musical notes on a CRT screen, and the New York Chapter/ACM award winner, William Blum, who has developed what may be the world's most advanced analog-digital music synthesizer (more on that in a future issue of *Creative Computing*).

Many, if not most of the fair projects were dreamed up and executed by the students completely on their own, like Alan Sung of Douglaston, New York, who wrote a program to organize Regents' exams, grade the results, and produce statistical analyses, replacing a team of 10 teachers.

Space doesn't permit a description of the many other exciting exhibits, such as computer pinball, football, and poker games, or BATTLESHIP, or "A Natural Language Problem Solver Employing Modified Deterministic Finite State Automata" (I carefully bypassed that one—I had learned my lesson with Abrams), or Robert Bedichek's homemade minicomputer breadboarded over an entire length of the Coliseum (well, almost), or Lane Molpus (great name, that), who designed his own computer from scratch, or SWARMS (a computer model of attacking bees from South America, no kidding), or the many imaginative stories and drawings—and even a ballet—but *Creative Computing*

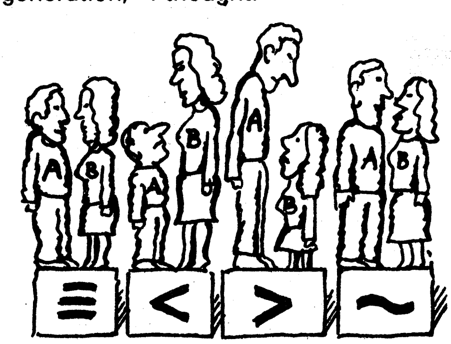


Three computer fair entries from the Northern NJ Student ACM Chapter.

(where else?) will be carrying more on the fair in future issues.

What was this fair all about? I asked organizer Sema Marks: "We wanted to get away from the computer science fair. This wasn't going to be 'let's just build another compiler.' Computers are for everyone, and computer power is soon going to be in the hands of all the people. One of the interesting questions is how is this computer power going to be used, how can we start thinking about it? These kids regard computers as a free and easy-to-use resource. They are totally intolerant of hard-to-use systems. They are interested in how computers can be fun and how to do things better. We wanted to influence the schools in a very subtle way—to say there's more to computers than FORTRAN programming. We wanted to get the English and Social Studies teachers involved—to encourage students to think about computers, draw pictures about them, write essays, incorporate them into their own way of thinking. I think we succeeded." Well, so did the thousands of enthusiastic people attending the fair. It was the largest, the best organized, and most attractive student computer fair so far. The kids even wrote their own proceedings, published in a giant binder. As Sema modestly told me: "This fair set a new standard."

As I walked out, slightly numbed by it all, Scott was still trying to outguess the talking computer ("...too low—try again..."), Bedichek was adjusting the data rate of his I/O board and trying to tap into the experiment next to him, and Korn, Lederman, and Sage were in a huddle, probably plotting the takeover of IBM. "It's going to be an interesting generation," I thought.



The Madness Known as Programming Contests

by John Lees
University of Missouri-Rolla

Programming contests are a rather new form of competitive sport. I don't know when the first programming contest was held, but it can not have been too long ago; the necessary technology has been in existence no more than fifteen years. Such contests are probably a phenomenon of this decade, since the primary ingredient — crazy students of computer science — has been available in quantity only for the past few years.

Just what is a programming contest? That is still open to definition at this time. I am going to describe my experiences as a co-chairman of the second Annual North Central Regional Programming Contest which was held at the University of Missouri-Rolla on April 17, 1976. The preceding year I was a contestant in the first North Central Regional, so I have now seen programming contests from both sides. If pressed on the question of whether I prefer being a contestant or being a co-chairman, I will reply, "Next time I think I'll just watch."

The contests held at Rolla drew from twenty to twenty-five participants from the ACM's (Association for Computing Machinery) North Central Region, an area which includes about 400 Colleges, Universities and Junior Colleges, all of which were invited to participate. Each institution could send one team consisting of up to four student team members and a team sponsor to the contest. Teams had to pay their own transportation and lodging, but there were no fees associated with the contest itself. Facilities and time were donated by the UMR Computer Center, students, faculty and staff, with financial assistance from the ACM North Central Region. The UMR student chapter of the ACM was responsible for organizing and running the second annual contest.

The exact form of a contest is dictated by the facilities available and by the need for a standard language. One of the most difficult requirements of a programming contest is that everyone must be programming in the same language, everyone must be aware of the language standard used, and the language must be popular enough that the contestants don't have to learn it on the fly. Taking all three points into consideration there was no choice but to adopt ANSI FORTRAN IV as the contest language. No one in their right mind would think of a contest using COBOL, and there are no other widely used programming languages. (Ed. note: BASIC?)

The facilities at UMR are punchcard oriented, so teams were distributed around the building in such a way that each team had a keypunch, a table and blackboard space. To ensure essentially instant turnaround, we detached ourselves from the University Network and ran WATFIV, a fast, in-core student FORTRAN compiler, on our own 360/50. Contestants could read in their decks on a cardreader in the hall; another cardreader in the machine room was used to run decks which had been handed in for judging. Output was printed on an 1100 line-per-minute printer and handed back immediately after the run had been logged.

Following an explanation of the contest rules and time to look over the facilities and locate their assigned rooms, the four contest problems were handed out and the contest began at 10:30 a.m., closing at 4:00 p.m. The teams could work the entire five and a half hours (most did), but program distribution was closed during a one and one-half hour lunch period: decks could be read in during that period, but no output was handed back and decks could not be handed in for judging.

Each team had to solve the same four problems. They could approach the task in any way they wished; all that counted was getting a program written in ANSI FORTRAN that would give the correct output when run with the official data. Most teams seemed to assign one member to each problem and work more or less independently, although it is not at all clear that this is the best of all possible strategies. As many runs as needed to debug a program (at 10 points penalty per run) could be made through the hall cardreader with any test data dreamed up by the contestants allowed. Once a team felt that they had a problem correctly programmed, such that it would work to specification with any possible input data, the deck was handed in at program distribution to be run with official data and judged. Submitting a deck for judging incurred a 15 point penalty plus a real-time penalty in that ten minutes or so were required for the deck to be run with official data and the source and output judged.

If the judges found an incorrect answer or an ANSI violation, the source listing was marked as such and handed back to the team for them to correct the problem and try again. The *output* of judged runs was *not* handed back, i.e., the contestants could not find out what the official data was. When a run was judged to be totally correct, the time of the run was put on the scoreboard and that team could forget about that problem.

The problems were far from easy. This year only one team, University of Nebraska-Lincoln, completed all four problems to finish in first place. One team completed three problems and quite a few teams did not complete any of the four problems. The problems ranged from playing BINGO to a loading dock problem involving moving crates through a hall and around a corner (the program had to determine if the dimensions of a crate allowed it to be moved without tipping it on a corner). The winning team made seven judged and twelve non-judged runs. One team gave up before the contest was over and ran a job which printed the message "WE GIVE UP. WE ARE SLASHING OUR WRISTS IN THE RESTROOM." 2000 times. We felt that this was in rather poor taste and canceled the job.

We feel that the First and Second Annual North Central Regional Programming Contests have been successful. The participants have all seemed to have fun and everything has run more smoothly than we have hoped for. (Note: 100 dozen homemade cookies are too much for eighteen teams and assorted contest personnel to eat in one day. The same comment applies to 60 gallons of soda pop. 30 dozen donuts is about right.) One thing that bothers me is that programming contests seem to encourage anything but "good programming." The only thing that counts in a contest is results, however gained. Perhaps someone can figure out how to have a contest in which programming technique and style are taken into account.

What do you need to hold your own programming contest? First of all you have to decide if you have the physical facilities. Acquiring the usage of an entire building, a good size computer system and twenty-five *working* keypunches for an entire day is not a trivial task. Of course it should be possible to use a different setup for running a contest. I'd like to see one run interactively in BASIC, but the problems of familiarizing teams with BASIC and with the interactive system, of there being no standard for BASIC and of scoring and allowing protected access to official data sets must first be surmounted.

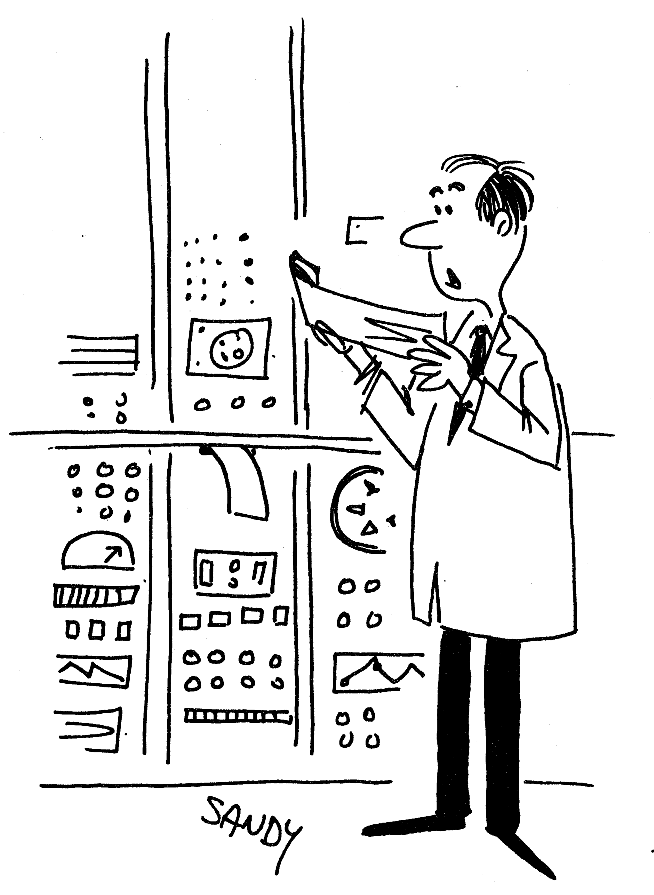
Once one has the facilities required to host a programming contest, all that is needed to pull one off is ten or fifteen people willing to enthusiastically write letters, type letters, duplicate letters, stuff envelopes, formulate rules, make plans, bake cookies, move keypunches, tear paper, answer questions, clean the place up, etc., etc., etc. The number of picky little details which must be taken care of during the four or five months it takes to plan and carry out a contest are staggering.* Having a few hundred dollars for postage and trophies and food doesn't hurt at all, either. To anyone crazy enough to host or to participate in a programming contest, I can only wish the best of luck!

---

*A packet of information on how the UMR contests were set up may be had from:

Dr. John Metzner
Computer Science Department
University of Missouri-Rolla
Rolla, Missouri 65401

Please include several stamps to help defray postage costs.



"To err is human. To really foul up a computer takes a Man."

DARTMOUTH TO DESIGN, DEVELOP COMPUTER-BASED VISUAL RETRIEVAL SYSTEM UNDER EXXON GRANT

HANOVER, N.H.—The Exxon Education Foundation has granted Dartmouth College $87,500 for the design and development of a computer-based system for retrieving visual information, such as reproductions of works of art, photographs of minerals, star maps and anatomical illustrations, according to Prof. Arthur W. Luehrmann, assistant director of the Office of Academic Computing (OAC) and assistant director of the Office of Instructional Services and Educational Research (OISER).

Professor Luehrmann described the system as one in which pictorial information will be reproduced as color images on microfiche cards. As many as fifty thousand different images will then be stored in a microfiche projector. The projector will be attached to an ordinary computer terminal on the Dartmouth Time Sharing System and will receive its commands from the computer. Five microfiche terminals will be acquired under the grant and will be used initially by the art and earth sciences departments and the Hopkins Center Art Galleries. In future the system will be available to other departments. Data bases will be developed for information retrieval using Project Find.

In describing the potential uses of the system, Professor Luehrmann cited several representative possibilities, including student use in art history courses. For example, the student might ask the computer to search through the entire art collection and retrieve all reproductions of a time period or style, including or excluding specific artists. The computer would then select all objects fulfilling the definitions of the student, and would project them in an order selected by the student onto the screen of the microfiche projector.

Another user of the system might be a geology professor who would write a tutorial program with photographic illustrations to teach certain concepts to students in a mineralogy course. The system could also be used by the staff of the Hopkins Center to plan art exhibits, by computer-assisted browsing through pictorial reproductions of the art collection, and making selections and deletions for a specific theme.

Initial project work and planning were begun with the aid of a $9,400 award to OISER in October, 1975, by the President's Venture Fund, established at Dartmouth by the Ford Foundation to encourage new educational projects.

Much of the work on the OISER/OAC project, which will include computer programming, photography and preparation of the microfiche images, will be conducted by students. Steven R. Johnson of Seneca Falls, N.Y., a bachelor of engineering candidate at the Thayer School of Engineering, has designed the computer interface for the projector. His work is supervised by Prof. Irving Thomas. Dr. Jan van der Marck, director of galleries and collections, will supervise cataloging activities for the Hopkins Center. Profs. Robert McGrath and Richard Birnie will oversee activities related to the art and earth sciences departments. Professor Luehrmann has responsibility for overall direction of the project.

"On one occasion Aristotle was asked how much educated men were superior to those uneducated: 'As much,' said he, 'as the living are to the dead.' "

Diogenes Laertius

Calculators in the Classroom

Pro: Calculators make tedious math fun, fast and accurate, educators and students agree. When used for creative problem solving, motivation in students seems spontaneous.

Con: Mechanization of fundamental classroom skills may leave kids unable to do simple math on paper. The cost for electricity or batteries may make operating the device daily too expensive.

by Deedee Pendleton

Conrad, a Washington, D.C., second grader, is 1 billion, 296 million seconds old right . . . now. Or, if you prefer, seven years, three months and five days. If you ask him how he knows, he'll tell you he figured it out on his calculator. If that sounds a little unsettling, relax. Conrad is getting a first-hand lesson in using his father's $40 electronic hand calculator at school. And although some parents are complaining the basics of education are being undermined by machines, the kids seem to love it.

Pocket math, as it's called, has been assaulted on all sides, but both the manufacturers of electronic hand calculators and progressive educators are anxious to see one in every classroom, if not one at every desk. Some first graders are already doing basic addition with calculators the minute their teachers feel they understand the principles, and high-school and college students are buying calculators as if they were radios.

Some calculators cost as little as $20, or about the same as some textbooks, and instructors say they could become required equipment in advanced math classes. The pocket-sized units are already replacing textbooks in elementary schools, and teachers are hoping that what once seemed to children a tedious labor may, through the calculator, become fun.

Opponents of calculators say that kids won't know how to count if their calculator batteries ever go dead, just as TV-oriented students no longer seem to know the basics of grammar and spelling. The device, critics contend, will make pencil-and-paper math obsolete.

But instructors who are using them take the opposite stand. They say that calculators stretch the student's interest, allow for more relevant kinds of problems (how far is it to the moon?) and increase motivation. Because of their speed and accuracy, calculators lend themselves to complicated problems previously avoided by grade-school teachers.

"One of the important uses of hand calculators is to enable children to solve more interesting problems, and to work out large divisions which would otherwise discourage them," says George Springer, an Indiana University mathematics instructor. Thus, oversimplified problem solving becomes unnecessary.

Teaching the basics before letting the child experiment with the calculator, many arithmetic teachers say, is essential for the machine's best use. "The hand-held calculator can be a very valuable tool, but only to an operator who understands the basic ideas, concepts and meanings behind the instantaneously generated answers it provides," says Frank S. Hawthorne of the New York State Education Department. Unlike the abacus, a calculator provides little or no help in learning computational skills.

Calculators will help children adjust more readily to a technological world, Springer says, and will make it easier for them to understand decimals, on which the metric system is based.

There is some opposition to the calculator in the classroom, admits Douglas Lapp, a Fairfax County (Va.) science curriculum specialist, but he says it isn't always valid. "Americans are particularly prone to think technology will offer easy solutions to everything, when in fact it simply solves existing problems, but does nothing automatically.

"The fundamental problem in math education still is that kids too often don't know the meaning of mathematical education and won't learn any more than they did by rote memorization," Lapp says. "We need to first give them concrete specific solutions as physical models for multiplication, which they can later transfer to concepts." Jill Horlick, an elementary-school math specialist, agrees. "The calculator doesn't think for you; it doesn't have a brain." She says that once her students can understand the theory of multiplication, they can adapt their knowledge to their imaginations. "Kids *normally* think about the universe; they love to manipulate large numbers because it makes them feel important. Why stop [the child] from thinking beyond those numbers just because he doesn't have the tools yet?"

New mathematical principles adapted for the calculator classroom are inevitable, Horlick maintains. More emphasis is placed on estimation, or on learning to judge which of the answers the calculator gives is reasonable. In addition, decimal placement becomes much more understandable, she says, because the calculator is able to provide answers of 6 to 12 digits, far beyond a young child's ability to calculate on paper. Children too often become bogged down in the complexity of a problem on paper, "and lose sight of the problem they are trying to solve," while the calculator eliminates the long rows of numbers usually associated with four-digit multiplication problems.

Douglas Grouws, a University of Missouri mathematics education instructor, holds that educators "must pay careful attention with regard to how we use [calculators] in the class-

. . . Calculators

room. I don't think availability will necessarily eliminate the need to be able to calculate by hand. However, the calculator may shift the emphasis away from proficiency in hand calculations to a greater emphasis on the meanings of the operations and when they can be appropriately applied." Grouws recommends using calculators in combination with basic arithmetic skills by, for example, providing them to students to check handworked problems.

But Horlick maintains that using the calculator for a combination of processes is essential. "You're defeating the purpose if you only used the calculator to check answers. The child wouldn't be learning to use the principles of the calculator."

The primary question, in her view, is, "Does the student know what he's doing?" Much more emphasis must be placed on, "What does it all mean?" than on "How fast can you get the right answer?" Those most opposed to calculators have gone so far as to ban them from the classroom, fearing that the device could become a crutch and keep students from learning the basic mathematical skills. Another argument for calculators, though, is that they make complex and realistic teaching exercises possible (how many cubic centimeters would it take to fill this room?). First graders, Horlick says, love to plan a family vacation, calculating costs of gas, motels and food.

A survey of teachers, mathematicians and laymen by MATHEMATICS TEACHER magazine has revealed that 72 percent of those polled opposed giving every seventh-grade student a calculator to use during his secondary education, but 96 percent agreed that "availability of calculators will permit treatment of more realistic application of mathematics, thus increasing student motivation."

In Virginia's Fairfax County, math teachers voluntarily agreed to permit high-school students to use calculators for homework and for some class assignments, but to forbid their use on tests unless every student in the class has one.

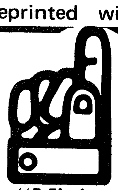
"With prices so low for calculators, it's no more a flight of the imagination to buy a calculator than it would be to buy a textbook," Springer says. When industry uses metrics and decimals exclusively, he adds, students taught to use calculators in school will be able to adapt quickly. There is one calculator for every nine Americans, and students who can't afford their own often borrow calculators from their parents. The device has become an essential part of training in statistics and computer science.

By 1976, the price of some calculators is expected to drop to as little as $10. If it does, the possibility of supplying public schools with them, and consequently incorporating them into elementary- and high-school math programs, may become very real. □

It appears as though Deedee Pendleton could have used a calculator herself when she wrote her article on "Calculators in the classroom." Her opening sentence states that Conrad, a Washington, D.C., second grader, is 7 years 3 months and 5 days old, and that this figures out to 1 billion 296 million seconds. If she had checked her second grader's results with her own calculator she would have seen that his age actually figured out to slightly less than 230 *million* seconds!

A billion seconds is actually quite a long time. One of the techniques used to impress upon the general public the differences between a million and a billion (particularly when the Federal budget is being discussed) is to point out that a *million* seconds is just over 11½ days, while a *billion* seconds is about 4 months short of 32 years!

William A. Robinson, P.E.
Solon Mills, Ill.



Tips for Buying A Pocket Calculator

By PETER WEAVER

"Miniature, electronic calculators could become as common on kids' desks as pens, pencils and rulers."

So says the superintendent of a major urban school system. As the price of these tiny mathematical wonders drops dramatically, more and more students will be able to afford them.

Right now, you can buy a basic, four-function model (add, subtract, multiply and divide) for as little as $19.95. The lowest price a year ago was closer to $90. Next year, the price for the least expensive models could drop below $15.

Whether a student should be using a calculator for doing homework depends on the math teacher you talk to. Some say the little machines are great and take a lot of the "dull, donkey work out of working with figures." Others say the calculators "could become a crutch and prevent the student from really learning fundamentals." Some compromise by saying "up to junior high school or so, students should do their homework the old way and only use the calculators to check their results."

In high school and college calculators can greatly speed up project work in math, science and business administration courses. But, in high school students probably need more sophisticated calculators, which can retain sums in a "memory" and can do percentages or square roots. These machines cost around $70.

Whether you're buying a calculator for a student going back to school or for yourself to use around the house (shopping, checkbook balancing), here are some things to look for:

- Keyboard. Are the keys hard to depress? Are they too close together for your fingers (pressing two down at the same time)? Do you know when you've registered a figure by a clock or a sense of touch?
- Number display. Are the numbers easy to read? Or are they in broken lines that tend to make eights look like zeros? Can you read numbers at an angle (some cheap models require reading head on)?
- Batteries. How long do the batteries last before changing? Some last only eight hours, other last 108 hours. Is an AC wall plug available?
- Logic Systems. Some calculators base their computing logic on a mathematical system, others base computing on a stacking system or algebraic system. Algebraic is easy for average users to understand because it works the way you would state a problem (100 minus 25 equals 75).
- Warranties. Most manufacturers give one-year warranties and are pretty good about fixing or replacing defective machines. But, who has to send the machine back to the manufacturer — you or the dealer? How long does it take for repairs? For the inexpensive, $19.95 calculators, it probably isn't worth it to have them repaired after the warranty runs out. It would cost too much. A few dealers are giving two-year warranties on machines costing $35 and up.

Don't buy extra mathematical functions you won't need no matter how exotic and prestigious they sound. It's a needless expense.

AMATEUR COMPUTING

by Sol Libes, President
Amateur Computer Group of New Jersey

What is Amateur Computing?

Amateur computing is private computer use, in one's own home, of one's own computer—a computer that is most usually home-built and microprocessor based.

Amateur computing is the prelude to the future of a computer in every home. It is following in the tradition of Amateur Radio.

At the turn of the century when Marconi developed the antenna and demonstrated that radio signals could be transmitted significant distances, amateur radio experimenters eagerly began experimenting. They developed transmitters and receivers, communicated with one another, improved equipment, expanded the range and frequency spectrum. It was not until the 1920's that commercial broadcasting began and the home radio receiver became common-place.

It was the radio amateurs, experimenting in basements and attics, who laid the foundation for the home radio receiver. In 1900, if you had tried to tell someone about radio receivers, they would not have even understood what you were talking about. The situation in home computers, today, is much the same.

Today, amateur computer hobbyists are embarking on the road toward the home computer. It began in the 1960's with the availability of economical digital IC's (integrated circuits). With them, a few determined pioneers built home computers to perform special functions. Around 1972, a company called INTEL introduced a single IC which contained most of the circuitry for a small control type CPU (central processing unit). This IC, called the 8008 MPU (microprocessor unit) made it possible for amateurs to start building home computer systems. They interfaced them to TTY's (teletypes) and started to use them for general purpose applications (like game playing) and applications for which the manufacturer had not really designed them. But then again, amateur home experimenters are always doing things like that.

With the 8008, amateur computing, as a hobby was born. In 1974, INTEL introduced the 8080 MPU—more powerful, faster, and easier to use than the 8008. Motorola and several other manufacturers introduced MPU's, prices dropped and the hobby began to grow. There were several hundred homebuilt computer systems.

In January 1975, MITS Inc. introduced the ALTAIR 8800 CPU kit (using the 8080) making it even easier to build a home computer system. Now there are several dozen kits on the market (in a future article we will try to rate them) and several thousand computer hobbyists.

The hobby is growing like wildfire—particularly as home machines become easier to assemble, more powerful and lower in cost. Before long, home computing will go commercial too, as radio did, and we will see a computer in every home!

Who Are Computer Amateurs?

Like radio amateurs, amateur computer hobbyists come from all walks of life. There are high school students, teachers, computer programmers, researchers, radio amateurs, retired senior citizens and others too many to mention. But, they all have a common interest. It is their incessant curiousity and eagerness to learn anything new. It is from these computer amateurs that will come the computer revolution of tomorrow.

What Do Computer Amateurs Do?

First of all, most computer amateurs build their own computer systems, usually from kits. A typical home computer system, of today, consists of a CPU (the ALTAIR 8800 is currently the most popular), with about 12K (12,288) words of IC RAM (random access memory), 1K (1,024) words of IC ROM (read only memory) containing the Monitor system control program, a key-board input (usually surplus), a TV alphanumeric display (using a black and white TV receiver) for output and an audio cassette (hi-fi type) for program storage. Typical software includes an assembler, program editor, text editor and BASIC interpreter. This typical system presently costs about $1200 to $1500 to build.

Computer amateurs use their systems for hardware and software development, for playing games (mostly in BASIC), word-processing, automatic operation of amateur radio stations, monitoring home operating systems, scientific calculations and analysis, book-keeping operations—and much more.

There are amateurs with full color graphics displays on color TV sets, amateurs with talking computers—and on and on—there is no limit to the home computer's applications. Can the home-built robot be far away?

How Does One Get Into Amateur Computing?

If the preceding has whet your appetite and you want to look into amateur computing, the first step is to attend a meeting of an amateur computer club. There are now several dozen in the country. An up-to-date list of computer clubs follows this article.

Learn from the experiences of others. Computer equipment is still expensive. But there is a lot of used equipment available—much of which is sold or traded at amateur club meetings.

Also, if you build your own system, the likelihood is that it will not work and will require debugging. Clubs offer assistance to members in getting their hardware up and running.

Also, most clubs have a software librarian, so that software can be made available at low cost. Most amateurs make their programs available to other amateurs. Most clubs also do "group purchasing" to obtain discounts for their members. Another important function of clubs, is spreading the word on suppliers—which are reliable and which are not (unfortunately there are some unreliable suppliers in this area).

Keep in mind, that when you undertake to build your own home system, IBM will not be there to provide hardware and software assistance (besides who could afford their prices?) and a club will serve as your back-up.

In The Next Issue

This column will continue in the next issue with a discussion of magazines and books for the amateur computer hobbyist.

A RETAIL COMPUTER STORE? YOU GOTTA BE KIDDING!!

by Jim Dunion and Ron Roberts*

FAR FROM THE MADDENING CROWD

The doors are boarded up now, and the raging mob seems to have subsided. For awhile at least I'm safe. The light is flickering low on this, my last candle, and I realize that time grows short for me to put these thoughts onto paper. For tomorrow brings a new day, a new mob, a new call from David Ahl wondering where this article is. Time indeed, grows short. One year in the retail computer store business. One year. Is it possible that only a year ago this madness descended upon me? That only a year ago I was a reasonably happy, nonchalant student, quietly pursuing a graduate degree in computer science and then -ah- but now I'm getting ahead of myself. First of all, what we're talking about is retail computer stores in general, and the experiences of one such store, The Computer Systemcenter, in particular. What we have seen in this past year is no less than the birth of an industry. Now, with a full year under our belt, it is time for the first self-assessment. Retail computer stores — from where did they come, how have they developed, and where are they going? While we certainly can't speak for all stores, we can provide insight to the structure and events surrounding one such store's existence.

THE BEGINNINGS

In recent years, there has been an increasing tendency for technology to advance faster than society can adapt to its changes, a fact particularly visible within the field of solid state electronics. Most of those involved in either electronics or computer technology are aware of the recent and continuing advancements which have led to the development of the "computer on a chip." Although the implications of readily available low cost computer power have been discussed many times in many different forums, it has been almost invariably within technical, non-public circles. For the general public the usual interaction with any form of computer technology has been via the "DO NOT FOLD, STAPLE, OR MUTILATE" admonishment. To John Q. American, it has little mattered whether the phone company used their 1st, 3rd or 300th generation equipment to produce his bill. To him, a bill is a bill, and due to the aloofness and mystery which has always surrounded it, a computer is a computer.

With the advent of the microprocessor, however, this public perception of computers began to change. The availability of the personal computer as a consumer item initiated tremendous strides towards public enlightenment of the computer as a useful, entertaining tool. Now, computer stores are everywhere, hobbyists clubs exist in almost every major city, and news coverage and public visibility is swelling on a national, even worldwide, basis.

But it was not always like this. When Dick and Lois Heiser pioneered the first computer store in southern California in the fall of 1975, every move was a gamble. How was the public going to react? Was the initial surge due to a sincere interest, or due to a shortlived "fad." After just beginning to recover from a nationwide recession, would the public climb upon an expensive, yet unproven movement? And the equipment! Could it be obtained? Would it work? Could it be kept working? While the popular computer concept was dawning, the task remained to acquaint and convince the public of its existence and stability. Sure, there were a few computer clubs around, but they were technically slanted away from the typical consumer. Already existing, too, were the professional computer societies, but they had (and still have) no room for the novice. Some popular computer publications were around, but very few people knew of these. And buying a computer by mail-order? Unassembled? My face still pales at the thought!

Where oh where, then, was the solution? What structure could tie down the loose ends, and present a united, uncomplicated front to the consumer? Obviously, (NOW its obvious) the retail computer store. The revolution was underway!

A retail computer store? At first the idea seems incredible, but why not? The equipment is available, it is priced at levels that consumers can afford, and the applications are endless. With this challenge in mind, and with the four partners crossing their fingers, the enterprise known as The Computer Systemcenter came to life. Being an unprecedented concept, there were no guidelines to be used in formulating this marketing strategy, no viable way to survey public attitudes, no anything but talent, enthusiasm and ideas. Although possibly not the best and certainly subject to change, herein is contained a summary of our philosophy, methods, and initial experience in this endeavor.

WHY OPEN A RETAIL COMPUTER STORE

This, of course, was the first and foremost question to be resolved. Although no member of the partnership had ever claimed being pioneers to any movement, the following reasons were used to justify the adventure:

- To create public awareness that solid state technology has reduced the size, delicate nature and cost of computers while greatly increasing their reliability. Cost reductions, of course, demand the greatest emphasis.

**The Computer Systemcenter, 3330 Piedmont Road NE, Atlanta, Georgia 30305. (404) 231-1691.*

- To present this low-cost computer technology in a setting with which the public is comfortable and well acquainted. The open marketplace arrangement, with freedom of movement and selection, should be as advantageous with computer products as it is, for example, with stereo or photographic gear.
- To bring together under one roof the preferred items from the equipment lines of several manufacturers. In the old computer world to even suggest crossing the lines of one manufacturer with another is almost sacrilegious.
- To provide a local and publicly accessible source of computer expertise. The public, particularly in its initial learning stages, is going to be seeking a qualified technician with a layman's delivery.
- To generate public understanding, confidence, and most of all, enthusiasm about the coming revolution in popular computers. This obviously is a challenge and a responsibility. The psychological factors encountered by springing computers in everyone's face may well be endless, so we'll hold back any discussions on this.
- To be an innovator in a concept which will alter the lives and thinking of this generation and all following generations. Hopefully, too, this innovation will turn us a tidy reward in our generation. Innovations alone do not pay bills.

Okay, so we were convinced. And luckily so were enough investors to get the project off the ground. We then turned to the next question.

WHAT SHOULD A RETAIL COMPUTER STORE BE?

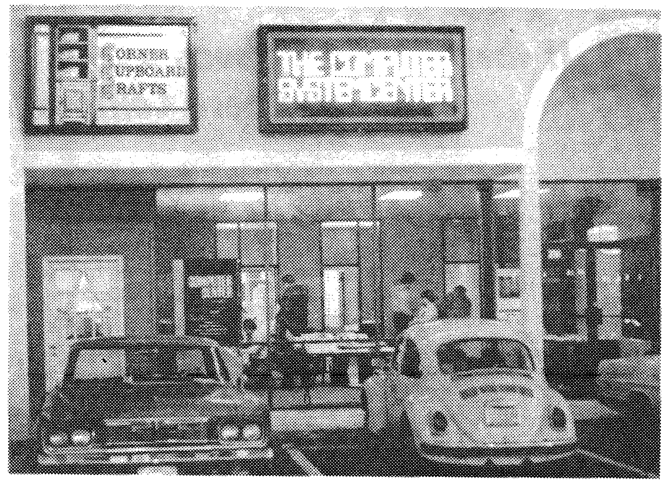
This should certainly be viewed as no small matter. After all, it's not everyday that one can set the philosophical guidelines for a revolutionary (there's that word again!) new concept. In addition to the normal properties any complete retail establishment should exhibit, such as good management, products and reputation, the unique nature of the computer as a product demands additional efforts and responsibilities. These include:

- A much higher level of competence. The retail computer dealer must become a cross between a personable department store clerk and a seasoned computer scientist (with a touch of showmanship a definite plus).
- An ability to communicate and to educate. Rarely will the typical consumer know less about what the product is and does than he will in this environment.
- An understanding of public apprehension and misconceptions about computers. Unfortunately, although through no real fault of their own, computers have received a lot of bad publicity and blame. And too, the "Big Brother" or "Numbered Society" image probably lurks in the back of almost everyone's mind.

MOVING FROM IDEAS TO REALITY

The notions of the armchair philosopher are good only to a certain point. It soon became time to move from concepts into a working concern. Actually establishing a real live computer store involved dozens of mundane questions that had to be answered and literally hundreds of small problems to solve, not to mention a few humdingers. Here is the rationale behind some of our major decisions.

- WHERE SHOULD THE STORE BE? We felt that the crucial aspect of our business was presenting computer technology to the people. Therefore, we had to be situated in a location with high visibility and accessibility. We finally chose a location in a new, small shopping center in northeast Atlanta.



- WHAT SHOULD THE PHYSICAL LAYOUT BE? Our store is broken into four sections. The public access is, of course, into the showroom where we continually maintain at least one complete system up and running for inspection and demonstrations. Equipment assembly, checkout, storage and repair is performed in what we call our work room, which is essentially a technicians shop. Then, for solitude and to retain some semblance of mental stability, we have our personal offices. Although tiny in size, this has not been a particularly important factor since we have had little time thus far to sit down. The last, and possibly most unique section of our establishment, is a large combination classroom/conference room. As a conference room, this area has had more or less predictable uses. However, due to the nature of our product, we felt that a classroom was mandatory for public instruction, seminars, and lectures.
- WHAT'S IN A COMPUTER SHOWROOM? In deciding what products to carry at the store our principle decision was which major line of computers to support. We felt that we could only offer adequate support and services for a single type of mainframe. Also, our supplier would need to offer the computers in both kit and assembled form for a reasonable cost. Lastly, the production and delivery capabilities of the manufacturer would need to adequately support us as dealers. At the time we made our decision, only one company seemed to fulfill these requirements. That was MITS of Albuquerque, New Mexico with the Altair 8800 line of computers. This decision was of course only a beginning, because even though MITS had several peripherals available for the Altair, we could not offer a complete product line strictly using their equipment. Thus we began making arrangements for various terminals (hardcopy, CRT, and color graphics) as well as several "plug compatible" devices such as interfaces to a standard television. We also felt it necessary to carry as many of the publications of interest to the general computer user as possible. Examples of these include *BYTE, Peoples Computer Company,* and *Creative Computing.* In the software area, we offer system programs such as monitors, assemblers, and text editors as well as a BASIC language. Computer games is an area of great interest to the consumer, and lends itself to entertaining but informative demonstrations. Consequently much of our initial effort has been in that area.
- HOW ABOUT SERVICES AND SUPPORT? Having the necessary equipment in the necessary place only serves as part of the attraction for any retail operation.

The ability to develop, troubleshoot, and maintain hardware and software is a must, as is the ability to speak (or at least listen) intelligently about pertinent matters on a consultation basis. And again, due to the very nature of our product, user community support is of utmost importance. The public must understand or feel that they will be given the chance to understand. We offer, for example, free introductory classes (both hardware and software) to the purchasers of our units, with only a nominal charge to non-purchasers. For the do-it-yourselfer, an hour (at least) each evening is set aside for him to bring in his under-the-weather Altair and get free troubleshooting. Other community functions include the active support of the local microcomputer hobbyist club, one of the largest of its kind in the nation. We also have gained the reputation as being a local depository of technical computer-related brochures and publications. Keeping abreast of the latest price changes and new product offerings is necessary for our survival.

INITIAL EXPERIENCE

October and November of 1975 were spent building and furnishing the store. During this period, we noticed a faint quickening of the public pulse at the shopping center where we are located. More and more people stopped by to talk and find out what type of place this was going to be. In November we were already working out of a half-finished store front. Finally, after months of preparation, we opened the doors on December 20, 1975. Since then the experiences have been truly remarkable. We have run the gamut from uproarious laughter to the utter frustration that seems destined to accompany any business operation. Problems? They occur by the dozens. Basically, however, they can be classified into one of two areas; either problems that are common to all small businesses, or problems unique to computer stores.

The largest obstacle we have had to overcome is our own lack of business experience. Initially this didn't seem too important, but since then we evolved our own form of Murphy's Law: If something can be done wrong—we will do it wrong; and just to be sure, we'll do it wrong two different ways. We have certainly not been immune from the various small ailments that plague small businesses — lack of management expertise, supply problems, cash flow, bad checks, you name it. At times these daily problems seem to outweigh and overwhelm everything else, causing us to occasionally have to reach down deep and rely on a certain humor to see us through. One of our pet diversions is coining "Anti-Slogans" that seem to fit the mood. We have a few classics, such as:

"Progress — We Sneer At The Term"
"Problems Are Our Most Important Product"
"Where Concepts Become Confusion-And Confusion Becomes A Way Of Life."

The other issues with which we deal are those unique to computer stores. First, there is the basic task of letting people know what we're trying to do. To the average person who walks in off the street, we usually have to tell them that even though they can be used as such, we're not selling calculators. Then we have to expect two stock questions, "What kind of place is this?", and "Well, what can you do with these computers?"

At first we would stammer around trying to pull together good answers, but by now it's practically a conditioned response. We hear one of these questions and bang! Put the old mind into AUTO and crank up the song and dance routine. I mean, we've got it down pat!

To characterize our typical customer is impossible. Applications range from monitoring water levels in the depths of a sewer, to writing payroll checks, to controlling a model railroad in someone's basement. Users include extremely sophisticated systems programmers as well as complete computer novices. Actually, it's less frustrating dealing with a complete novice who is somewhat awed by computers than it is to deal with an IBM 370 programmer who views microcomputers as "Toys." When this happens (and it does happen), we just take them in our computer room and show the business system on which we perform our accounting and inventory control (Altair 8800A, 40K of memory, dual disk units, video terminal and printer, all built into a custom desk). It's almost frightening when you think it's all based on a $30.00 microprocessor.

Our biggest miscalculation seems to have been just how much time is required by the computer novice. We tend to forget just how much there is to know about computers until we try to explain things to someone who thinks that a terminal is actually the computer. We've literally spent hours passionately pleading the case of Microcomputers to someone only to hear "Well, I'm really only in here killing time while my wife is shopping."

And the joys of Kit-building. Ah, there's a story in itself. Someone buys a Kit, puts it together overnight, it doesn't work, he screams, and brings it in muttering "damn crappy equipment." Usually, the next thing we hear is "What do you mean, bad solder joints? I went to the NASA soldering school." Still, we have a certain obligation to help each customer get his system up and running. We've tried to accomplish this by setting aside a certain time each day, (6:00-7:00PM), during which we have a free software and hardware clinic. During this time anyone can bring in their sick machines and/or programs and we'll give them a hand.

The latest issue we've had to deal with is the "software vacuum." People are discovering that after the machine is working, the real uses are just beginning. Canned programs are fine (programs written and debugged by someone else), but when it comes to writing one's own programs — well, there's more to software than meets the eye. To combat this situation, we have started a series of programming lectures entitled, The Art Of Creative Computer Programming." This series is aimed at providing a novice programmer with insights about programming and a set of software tools and tricks to tackle his own programming project.

WHERE DO WE GO FROM HERE

In a year's time computer stores have evolved from a few timid, rather speculative ventures to a firmly established concept. The first generation of stores are highly individualized with each having a different emphasis. In filling out the scorecard on ourselves, I would have to say that we set some very idealistic, but unrealistic, goals. But, there's no substitute for experience, and even with somewhat altered goals, our enthusiasm and energy still runs high.

What about the overall industry? The approaches offered by the differing stores are quite varied. At one extreme is the store that attempts to act primarily as a computer supermarket, emphasizing a broad assortment of equipment for the customer. The other extreme is taken by stores that emphasize primarily their service and support. Of course, this is actually a continuum. As computers become easier to use, as the general public becomes more aware, and as the software vacuum is filled, the tendency will be to move towards the supermarket concept. In these early stages, however, education, service, and installing consumer confidence must be paramount to all other considerations.

Would we do it over again? You can bet your solid-state bippies that we would. Each of our successes, whether a simple home computer or an intricate industrial system, causes feelings of pleasure and accomplishment. The dream of readily available computer power is now becoming a reality and we are sharing, and hopefully helping, in the transition.

GRAND OPENING

by Emily Pritchard Cary

One of the grandest of the Grand Openings was abruptly halted on Friday, April 11, 1975, at approximately 4:30 P.M., when an uncooperative component in a freshly installed Bunker Ramo Electronic Store Information System ceased operation.

The resulting pandemonium cast Mike Madonna, manager of the spanking new Shop-Rite Super Market in Springfield, New Jersey, into an unenvied starring role heretofore played only in recurring nightmares.

This nightmare, peculiar to supermarket managers, is heralded by a vision of the store crammed with customers, their shopping carts filled to overflowing in anticipation of the upcoming weekend. The clerks at the check-out counters are handling the throngs with customary efficiency. Suddenly, an inexplicable malfunction in the computer-based registers halts all operations.

Within minutes, chaos prevails. Irate shoppers, each intent on immediate attention, become restless, then belligerent, ultimately storming the aisles. The nightmare soars to a climax so hideous that the store manager wakens—screaming—in cold terror, fancying himself merely inches from the wrath of a lynch mob.

What happened when that nightmare slipped across the gossamer barrier which transformed it into reality?

Let me begin where I began, armed with an empty cart, a wallet wadded with crisp bills, a lucrative assortment of clipped coupons redeemable only during the Grand Opening Weekend, and a heart happy in the knowledge that both employees and mechanical complexities therein would mold my turn around the store into a memorable event.

"It's bad enough being trapped in a Grand Opening mob, but it's an absolute crime to be kept prisoner by a broken-down machine."

It was!

Police officers were on hand to direct cars feeding from the highway into the parking lot which already was jampacked. Inside, a bevy of official caps bobbed among the milling bodies, helping to steer foot and cart traffic through the aisles.

I decided to work systematically, beginning my forage on the far side in fresh produce. Despite the crowds and the difficulty in propelling my shopping cart with any semblance of speed, I did not feel inhibited. Clerks in the delicatessen, seafood, and fresh meats departments hustled through their chores, servicing each customer with dispatch. When one customer preceding me down the aisle was momentarily stymied by a blockade of carts ahead, she mused, "The crush is dreadful today, but it shouldn't be so bad in a few weeks when things settle down."

Her companion observed the energy of the check-out clerks responding to the rush of business. "Don't worry," she consoled. "Once we reach the counter, they'll whisk us right through!"

Admittedly, the entire process absorbed double the usual time, but much of the delay could be attributed to unfamiliarity with the shelf arrangements, the small children underfoot who were actively in pursuit of balloon-distributing clowns, and the gaggle of company representatives proferring product samples at several key intersections.

I rounded the final aisle wealthier by a number of freebees, secure in the knowledge that the endeavor had been rewarding.

A cursory glance at the ten check-out counters revealed some delay. I would have to wait my turn behind at least six other shoppers with carts piled high. I opted, therefore, for the nearest slot. As I edged my groaning cart into place, the woman ahead spun around and glared, fury peppering her countenance. What had I done wrong?

Studying customers in other lines, I realized that all wore venomous expressions. No longer were the clerks at the check-out stations herding the orders along with the alacrity and spirit exhibited earlier. Instead, all ten of them—together with their accompanying baggers—stood doggedly still, arms folded. A pervading silence further verified that something was amiss.

Whispers trickled back from the front of the store. As they spread to a steady murmur, I detected a lethal word flitting from one counter to another: "Strike!"

The word bounced back and forth several times, swelling to a roar like a cyclone building momentum. An elderly woman nudged me, ominously, "They've gone on strike!"

I did not know if she spoke the truth, but the mass inactivity up ahead deemed it a likely possibility.

A man behind me overheard her. "Good God!" he shouted. "Let's get out of here. There's liable to be violence."

So saying, he grabbed his wife by the elbow and steered her to the door. "But . . . but . . . what about our groceries?" she protested. "Our cart is full."

"Forget it," he growled. "There'll be plenty of trouble here in a little while. We don't want to get involved."

Once outside, he confronted potential customers, alerting them to the situation. The recipients of his bad tidings froze in their tracks, stared—disbelieving—through the huge plate glass windows at the motionless mob, then wheeled about and returned to their cars.

A lady paralleling me in line took an alternative stand. "I've been through strikes before, and I learned that you have to hold your ground. It must be something about working conditions. The clerks don't look very happy, but they're bound to clear it up in a few minutes. The manager can't afford to lose all these customers. It took me two hours to load up my basket, and I'm not about to leave and start all over somewhere else. You'll see. They'll get this strike over within fifteen minutes, or I don't know what I'm talking about."

"Since you don't plan to leave," I said, "I wonder if you would save my place while I try to find out what the real story is."

"Sure 'nuff," she agreed. "We're not going any place very fast. I'll be here when you get back."

Within minutes, chaos prevails. Irate shoppers, each intent on immediate attention, become restless, then belligerent, ultimately storming the aisles.

Already the strike theory had spread, and one could sense the fear and frustration enveloping the customers. The choice was heady. Was it better to wait for an incalculable stretch of time, or to leave and forfeit the goods that had been a challenge to amass?

I inched toward the main entrance. A policeman there appeared to be the only person communicating verbally with the public. "Is it true the clerks have gone on strike?" I asked him.

"Wow!" He threw back his head and roared. "Did you hear that one? They think you're on strike," he told the nearest clerk.

"Who started that rumor?" the clerk asked, stupified.

"Some of the customers," I replied. "If you're not on strike, then what *is* the matter? Why aren't the check-out lines moving?"

"The computer broke down. Damn thing just up and quit. Can't do a thing with the registers until someone comes to fix it."

This struck me as being a severer problem than a strike. At least a hike in salary could mollify an unhappy employee, but what can be done about a cantankerous computer if there is no knowledgeable repairman on tap?

"How long will it be before someone gets here, and where are they coming from?" I asked.

"Who knows?" another employee shrugged. "Just hang in there."

Hang in? And for how long? I decided to query the service desk.

Three girls huddled together behind it in vague fear. They knew little about what had happened and nothing about what could be done to remedy matters. "Oh dear, oh dear. What can we do?" one muttered. "Look at the mob in the aisles!"

Addressing another girl, I asked, "Is your manager in the store at the moment?"

"Is he in the store? He'd better be, that's all I can say, or we'll all go crazy!"

The third girl, slightly more composed, suggested that he would be on the upper level where the computer held forth. "But you're not allowed up there," she remarked, defensively.

I sensed that she suspected I might try to storm the computer—or attack the manager—or perpetrate a violent act of the sort befitting a berserk customer.

"I have no intention of going up after him," I assured her. "I'm just pleased to learn the source of the difficulty. I'll go back now to report to the customers. They are becoming angry."

There was no denying their churlish deportment. Sporadic chants demanding immediate service swelled to a steady throb.

It was push and shove back to my shopping cart. All along the way, I cried out to as many customers as would heed my words that the strike rumor had been erroneous. "There is no strike," I repeated, over and over. "There is no strike. It's just the computer."

My words mollified some of the more belligerent customers who consented, reluctantly, to grant me passing room.

Upon locating my cart, I discovered that my message had advanced faster than I had. "Don't worry," the woman saving my place assured me, "they say it's just a computer. There is no strike."

"*Just* the computer!" a nearby man hollered. "Who're you kidding? That's an impossible mess! You don't get computers repaired for days, sometimes weeks. Probably the company is headquartered some crazy place—like Texas! That would be just our luck!"

That tore it! The rumor erupted, inviting anger to billow forth as word ricocheted around the store that the computer had blown up and would have to be replaced. A new one being shipped from Texas would not arrive for at least a week.

The mob surged forward. Or was it merely pressure amassed from shoppers queuing up behind us?

By now, all of the aisles facing the check-out counters were packed solid to the rear of the store with customers, their shopping carts laden with food. Surely thousands of dollars were at stake. Each person demanded immediate attention, computer or no. Hadn't the flyers received in the mail, the newspaper ads, and the gala banners strewn across the facade of the building promised super service?

These people let it be known they had not driven all the way from Elizabeth, Glen Rock, Ho-Ho-Kus, and Heaven-Knows-Where to be done in by a microscopic computer component.

"So get a cash register!" someone yelled.

"Cash register? Phooey! Get a hand calculator!" another suggested, in an unkind manner.

"Whatsa matter?" a more practical man boomed. "Ain't youse never heard of addin' wit' paper?"

"Good God!" he shouted. "Let's get out of here. There's liable to be violence."

A small child up ahead screamed with fury as the mob drove him into a magazine rack. "Gun shots!" someone gasped.

A sharp, metallic edge had popped the child's huge balloon bearing a slogan suggesting strict allegiance to Shop-Rite Super Markets.

In retaliation for his loss, the boy kicked his mother, who responded in kind by slapping him smartly and yelling for all nearby to hear, "Shut up, you fool kid! It's those jerks behind me who ain't got no manners! Quit your shovin'!"

The child screamed bitterly, but his cries were drowned by the drone of dissatisfied customers. Many among us were becoming edgy toward our adjacent fellow man with whom we were presently congregated for no reason other than we had elected to patronize the Grand Opening on this fateful Friday afternoon.

The rumor erupted, inviting anger to billow forth as word ricocheted around the store that the computer had blown up and would have to be replaced. A new one being shipped from Texas would not arrive for at least a week.

Suddenly, the loud speaker commanded our attention. "All Shop-Rite specialists to the front of the store! I repeat: all Shop-Rite specialists to the front of the store!"

It took some doing for the specialists to reach their goal. Nobody in line was about to relinquish his place despite the plea, "Coming through, please, coming through." The specialists endeavored to assure the agitated shoppers that it would be to their ultimate advantage to step aside.

I recognized them as the product representatives who earlier had been supervising the stock boys stacking the rapidly emptying shelves. All were garbed in business suits, conservative ties, and deep frowns. They appeared to be out of their element in the role of cashier which they assumed presently.

Other employees shoved through the crowds and tapped those whose carts contained ten items or less. These lucky ones were advised, "Take your selections to the liquor department."

The liquor department cash register was not wired to the unfortunate computer system; smaller orders could be handled there. This proved to be token assistance, as most shoppers had stocked up for the week, but at least some in the crowd would be dismissed promptly. One man, overlooked for transfer to the liquor counter, tugged at an employee's sleeve. "Hey, buddy, How about me?"

The employee surveyed the cart, then relented, "O.K. Go ahead. You have only fourteen items."

An elderly lady near me, prim and genteel of mien, had six selections in her cart. "You can be waited on at the liquor counter," I prodded her, believing she had not understood the directive.

"Never!" she retorted. She was adamant. "You'll not catch me near a drop of liquor!" Doggedly, she stood her ground behind eight carts brimming with groceries.

Ugly dispositions began to flare. A pugilistic man far to the rear of our line took exception to the liquor counter decision. Why should customers with small quantities be serviced and the rest penalized? Mouthing his protest in X-Rated terms, he rammed his heavy cart into the person in front of him. The chain reaction of cart into flesh reached me moments later, resulting in a raw heel and a run in my stockings. By the time all the victims in our line consolidated their ire, the culprit had melted into the mob.

Another chain reaction erupted as a wave of furious customers abandoned their carts willy-nilly and stormed out the doors. En route, they verbalized their resentment for the indignities being thrust upon them. Several enjoined the rest of us to follow suit. "Let's show the management we mean business," one agitator cried.

A few more stoic customers were embarrassed by the actions of the rash ones, and a girl asked me to guard her place in line while she did the very least she could do as a concerned citizen: return the meats and frozen foods in the abandoned carts to their proper counters.

The compassionate girl need not have worried about losing her place at the checkout, for we were in the identical spot when she returned. The customers who remained—and there were dozens, even hundreds—seemed resigned to waiting out the ordeal. However, most made it vivid to everyone within earshot that they had no intention of revisiting the premises. One man put it succinctly: "It's bad enough being trapped in a Grand Opening mob, but it's an aboslute crime to be kept prisoner by a broken-down machine."

He vowed to scrutinize henceforth each market he enters to make certain it features "... good, old-fashioned cash registers." These, he pointed out, might individually cease operation on a whim, but odds are they will not *all* go on the blink at once.

Several husband and wife duos began bickering as to whose idea it had been to shop at this store. One man accused his wife of being unable to resist a Sale, and she retorted that if he were not such a poor provider she would not be forced to buy at Sales and Grand Openings in order to live within his lousy salary.

Numerous epithets were noised abroad, all hinting darkly of conspiracy in high places, infiltration of the Mafia into the computer industry, and a secret move afoot to subdue the public. The general consensus was that computer programming is the initial step toward the dehumanization of mankind, and if we submit to its authority, it will be no time before the communists who design and manufacture computers invade our private dwellings and spy on personal activities.

One man, gifted with sonorous delivery, decried the implementation of computers in any capacity and spewed his hatred equally between computers which seldom operate properly to those which soon will be planted covertly on our very persons. His captive audience tended to concur with his prophecy, although their bewildered faces clearly reflected a failure to trace his blustery line of reasoning.

Up ahead, two adding machines, newly located, had been moved into position at two counters. Computation at the other eight aisles would be done by pencil and paper. In short, third grade arithmetic was rushed to our rescue.

Hope soared when a computer specialist arrived on the premises, but he promptly reported that the problem hovered within the jurisdiction of an electrician. It was anybody's guess when one would arrive.

No matter, I detected a faintly perceptible forward motion of our line. The child whose balloon had been demolished was still whimpering, but his mother buoyed him with hope. "They're moving along now. See, the man is adding up the groceries with his crayon. Hush! Maybe we'll get out of here some time tonight." Then, more sternly, "If you don't shut up, the man won't let you out of here!"

Inch by inch, item by item, we battled our way to the counter as the Shop-Rite specialists patiently added column upon column of figures. Sometime after 6:00 P.M., my order was tallied and packed snuggly in eight brown bags.

Mayhem persisted throughout the Grand Opening weekend, my neighbor reported, despite the fact that three registers returned to service late Friday evening. I shall not belabor her adventures in the store except to report that she is convinced the world will arrive at a standstill soon thanks to hysteria born of malfunctioning computers.

Today the Springfield Shop-Rite Super Market basks in the glory of an operational automated check-out system, but one cannot help speculating when and where the public will be treated to the next nightmare sparked by a capricious computer gremlin.

An editor read this story and commented, "How clever, it sounds almost real" The reason it sounds real is because it actually happened and every detail in this account is absolutely true.

Polls, Pols, and Power: The Computer on the Hustings

by Nicholas Acocella

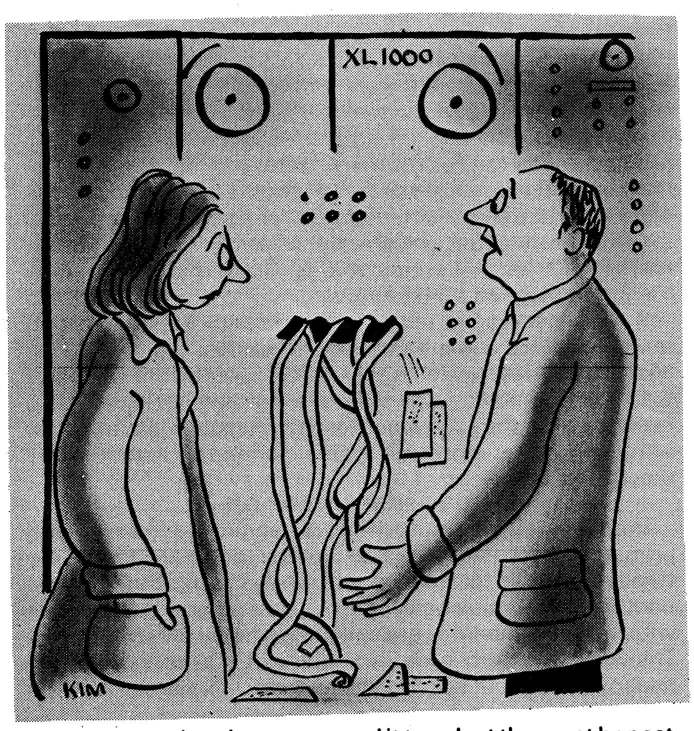
Politicians have been running for whatever office is available for as long as democracy has been around. It's the nature of the beast. Recent innovations in computer technology, while not noticeably affecting the volume of political activity, have created a paradox in modern campaigning, making it, at the same time, more complex and more simple.

On the one hand, political campaigning, like everything else in American life, has become a matter for specialists engaged in sophisticated pursuits not readily understood by the layman. At the same time, campaigning would have grown more complex even without the advent of computers, which seem to be part of the solution to the problem they simultaneously helped create.

At any rate, the ward heelers of yesteryear, with their derby hats, cigar stubs, and local accents,have yielded their places in the average campaign headquarters to specialists retained for their technical expertise rather than for their first hand knowledge of conditions in the third ward.

This new breed tends to be younger, more modestly dressed, and better educated than the precinct captains they pushed aside—the inflections of Harvard and The University of Chicago supplanting those of South Boston and the South Side of Chicago.

But—not so remarkably—they are trying to do the same things the old timers did less scientifically and more intuitively. They are raising money, categorizing voters, identifying voter opinions, and determining voter preferences: the things all campaigners have to do in all elections regardless of the methods used.



"It threw up when I programmed it to select the most honest political candidate." ©CREATIVE COMPUTING

Fund Raising

Fund Raising: "Money is the mother's milk of politics," once observed Jess Unruh, the one-time Big Daddy of California politics. And he was right. You can't run a political campaign without money any more than you can run an automobile without gasoline. In the old days you sent a friendly lawyer down to the local courthouse to corner other sympathetic members of the legal fraternity into signing $50 and $100 checks. Or you rounded up the fat cats for whom you either have done or might do a favor; they usually came through.

Today a campaigner, while he probably will not reject the first of these alternatives, simply doesn't have the second option because of campaign financing reform laws. Instead he will probably go to a direct response fund raising firm.

These firms stockpile computer files of past and potential contributers to campaigns. A candidate with a large environmental point to make will seek a fund raiser who has access to membership lists of the Sierra Club, Friends of the Earth and similar organizations. Candidates with heavy liberal credentials will use one who has the ADA and ACLU membership lists.

Governor George Wallace of Alabama has virtually perfected this technique, raising millions of dollars in small contributions from "just folks" over the years.

The variations within this arrangement are virtually unlimited. Some fund raisers—like Richard A. Viguerie, Inc., of Washington, who works only for extremely conservative candidates—are ideological. Others will accept offers from candidates within a wide range of the political spectrum, while reserving the right to reject the few candidates with whom they disagree entirely.

Some lists are finely honed compilations of names of those who *have* already contributed to the candidate (or those like him). Others are less selectively compiled, including people who are *likely to* contribute to a campaign of a particular stripe. Still others are even more inclusive, containing names of those who *might* contribute. Cost, on a per name basis, often depends upon the degree of selectivity.

Naturally, returns on more generalized lists are lower. Prospecting from such lists will bring in contributions from anywhere between 1 and 3 percent of those solicited. Such an endeavor simply isn't cost effective unless letters are printed (rather than computerized), in which case a run of labels is all a finance committee needs.

A purged or specialized list—perhaps consisting of contributors from the candidate's previous campaigns or from respondents to a prospecting mailing—deserves more

* Articles by Mr. Acocella have appeared in *Campaign Insight, Model's Circle, New York Affairs,* Fodor's Travel Guides, and five books on sports.

The computer letter, asking for immediate and future contributions, had four blank checks, a mailing label addressed to the recipient, and an application for a button identifying the wearer as an early McGovern supporter.

particular attention. Such donor lists can bring a response from between 10 and 25 percent of the addressees. Governor George Wallace of Alabama has virtually perfected this technique, raising millions of dollars in small contributions from "just folks" over the years.

But no major fund raising effort can be a folksy, backroom operation. It must be highly sophisticated, with computerized letters—perhaps mentioning the recipient by name in the body of the letter, perhaps not—going to lists of special interest voters, who usually get a basic letter with variant paragraphs for different groups. Such an endeavor might also include an information storage system retaining the names of contributors and the amounts they contribute, and capable of purging those who prove unresponsive after a few tries.

Perhaps the greatest computerized fund raising coup was the brainchild of Thomas L. Collins of the New York advertising firm of Rapp, Collins, Stone and Adler. During the 1972 presidential election Collins sent out a mailing on behalf of Senator George McGovern (for obvious reasons candidates with a strong ideological identification do best in this sort of drive) to a prime list of past and potential contributors at a cost of $25,000. The computer letter, asking for immediate and future contributions, had four blank checks, a mailing label addressed to the recipient, and an application for a button identifying the wearer as an early McGovern supporter. Americans, it seems, don't like to take something—not even a campaign button—for nothing, because 100,000 donors sent in $1 million in response to this mailing.

Polling

Polling: With enough money a campaign can go in any direction its directors choose; the problem is in defining the direction that will be most effective. The computer and its legitimate offspring, the poll, are there to help answer this all important preliminary question.

The kinds of polls available are about as many as the number of campaigns in which they have been used. Most depend on random samples of 100 or more names, below which the data is most suitable, culled from census lists, the telephone book, random digit dialing, or from any predefined list. The sample is screened to weed out the noncitizen, the unregistered and the non-voter. From there the pollster will frame questions designed to tell his client what he wants or needs to know. A few examples should suffice:

Asking preliminary demographic questions can yield a profile of the electorate's characteristics.

Asking whether the respondent will vote in the forthcoming election can yield a maximum likely turnout.

Collating the results of these two questions can yield a profile of likely voters, indicating not only how many people will vote but also what kind of people will vote.

Asking the respondent's predisposition toward the candidates can yield a model of the undecided voters similar to the model for all probable voters.

Richard M. Hochhauser, a former vice president of Cambridge-Plesser, a research opinion consulting firm, and now president of RMH Research, Inc., outlines the six basic kinds of political polls, any combination of which may be used in an individual campaign.

1. An Issue Definition Poll seeks to ascertain what the electorate at large and/or some portion of it (the most likely voters, various age groups, various religions or ethnic groups, probable supporters or opponents, etc.) sees as the most serious problems facing the city, state or nation. The pollster can gather the necessary data either by asking relevant questions directly or by providing a list of issues and asking the respondent to rank them.

2. A Bench Mark Poll, conducted with as much as a year's lead time, provides the information with which to screen the electorate, classify the voters demographically, develop a media profile (i.e., establish what the voters read and watch), determine the degree of the candidate's name recognition and the public's knowledge of him, evaluate the public's image and opinion of the candidate, establish the importance of the forthcoming election, and measure the depth of conviction of committed voters and the importance of party identification. With this information at hand a prospective candidate has the wherewithal to decide whether to go ahead with his campaign and, if so, what kind of pose to strike and what kind of campaign to run.

3. A Tracking Poll updates the information in a bench mark poll sometime before election day. Since much of the cost of polling is devoured by actual interviewing time, the use of the same sample as in the bench mark poll can save a campaign considerable money.

4. A Target Voter Survey selects a sample of voters whom the campaign wishes especially to reach. It may try to measure penetration into the opposition's supporters, or the degree of the campaign's effect on undecided voters or on some other subdivision of the total electorate.

5. There are two kinds of Communications Surveys. One, the Theme Effectiveness Survey, is used to determine the kinds of ads a campaign should use. There may be three possible ways of reelecting an aging incumbent, three thematic hooks on which to hang the entire campaign: "Senator Smith, a man of experience and accomplishment," "Senator Smith's stand on the issues," or "Youth can't keep up with the activities of Senator Smith." A theme effectiveness survey can help decide which would be most persuasive. The other, an Ad Effectiveness Survey, is used to determine the ability of specific ads, already released, to accomplish their desired effect. Obviously ads may have different impacts on different elements in a constituency, so a communications survey employs various screening questions to measure an ad's effectiveness with specific groups of voters. Furthermore, voters in different media markets may react differently. To take these differences into account in statewide or nationwide elections, the same poll is often conducted separately in each relevant media market. (Earlier polls may also be duplicated in as many media markets as funds allow.)

6. The last type of poll is the one candidates, campaign managers, and consultants like to conduct least, the Post Mortem Survey, which seeks to discover why the candidate lost. Actually, such polls are not always lamentable events. Often they are a part of the process of planning for ultimate victory, a process which sometimes extends over more than one term.

Voter Identification

Voter Identification: Hochhauser and those like him deal primarily with categories of voters, generalizations extrapolated from interviews with individuals. But there is another way of looking at voters, other categories of voters with which candidates and campaign managers must deal. These classifications define voters in a way that allows dealing with them specifically and individually rather than generally.

One such division is party registration, which is part of the public record and readily available at any Board of

John Goodfriend, a 36 year old political consultant from New York, has developed an ethnication program that can identify seven ethnic groups with 90 percent accuracy and 90 percent completeness.

Election. Obviously in a restricted party primary this information is a *sine qua non*.

A similar category that has come increasingly under the umbrella of the computer specialists is ethnic identification. Again, what they do is nothing so very new. The old ward leader knew the boundaries of the local Italian neighborhood; it was his job to know, and, if he didn't, there might be a new ward leader next time around. Also, a politician, recognizing the solidarity and common interests and aspirations of the Irish or Scandinavian voters in his constituency, would make special appeals to each of these groups by whatever means were available to him.

To a great extent the melting pot effect has blurred the lines of many ethnic neighborhoods. This is much less true of ethnic identity however. The geographic dispersion of people with common backgrounds and common concerns has made the ethnic scan a necessity in many urban (and even suburban) electorates. The fact that such groups also have surnames with common characteristics has made it possible.

John Goodfriend, a 36 year old political consultant from New York, has developed an ethnication program that can identify seven ethnic groups with 90 percent accuracy and 90 percent completeness. (His program is flexible enough to approach 100 percent in either field at some small sacrifice to the other.) His lists of names come from library sources, ethnic club membership lists, church lists, foreign and local telephone books and acknowledged experts.

Goodfriend's first approach was a simple name match, but this proved not effective enough. Today he employs 300 tables, based on a first name-last name, prefix-suffix analyses to sort out the seven ethnic groups.

The uses of such a list are as various as politics itself. An endorsing letter from a respected member of the ethnic group, a specialized mailing in the ethnic group's old world language, a move to strengthen organizational efforts among the candidate's co-ethnics: all these are possibilities.

But there are subtler applications as well. Working jointly with Steve Balber at Datatab, Inc., Goodfriend has also developed a process for compiling a prime voter list, that is, a list of voters who, based upon past turnout are most likely to vote in a given election. This gives a candidate and his managers access to the best, say, 100,000 voters at whom to direct his campaign.

Frequency is a variable factor here. A campaign may want only the names of those who have voted in *all* of the last four similar elections or it may want the names of those who have voted in *any* of the last four elections. The direction of the campaign's proposed attack will determine the exact definition of the list.

The fact is that some ethnic groups have better voting records than others. For instance, Jewish voters in New York City are about 20 percent of the overall electorate but constitute as much as 40 percent of the turnout in Democratic primaries. Accordingly, combining the two processes—ethnication and prime voter listing—can give a campaign manager the narrowest target group whose votes will win his election for him.

Canvassing

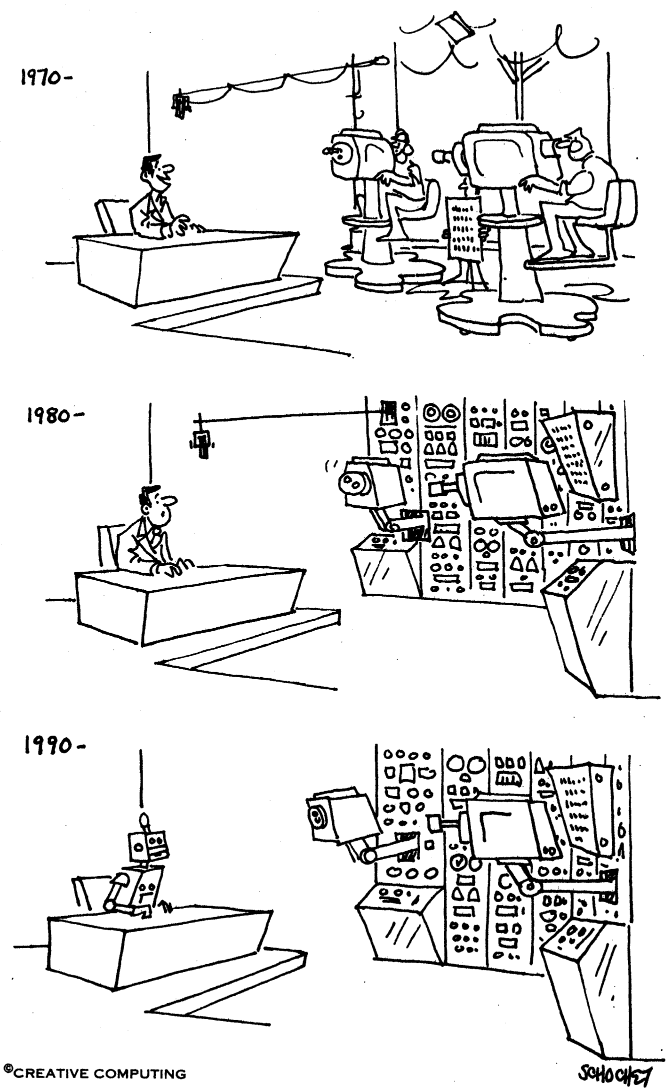
Canvassing: All political campaigns seek to accomplish three things: raise money, make noise to convince undecided voters, and organize an effective drive to guarantee that favorable voters get to the polls on election day.

The last of these, predicated on the assumption that it is easier to win by getting out your vote than by changing the minds of the other guy's votes, is as old as partisan politics. As far back as the presidential election of 1800 Aaron Burr organized a card file of every voter in New York City, listing his political leanings and what it would take to get him to the polls. That election lasted four days with Burr and his cohorts running from poll to poll to draw out the voters predisposed toward the Jefferson-Burr ticket.

Burr carried New York, became vice-president, and went on to infamy. So did his system.

It was an historically fitting fate. Burr's associates during those four days 175 years ago were the beginnings of Tammany Hall, from whose domain the Burr system spread to oil other political machines. In time the file cards took on added information such as when a Thanksgiving turkey was delivered, when a voter's nephew was given a position on the cops, etc.

Times change, but there is little that is new under the sun. The New Politics of the 1960's reinvented the Burr system, using all manner of hardware. The size and complexity of modern America had made its demands again.



With a computer file of every voter in an entire constituency, or in a specific target group, along with his address and telephone number, a campaign can canvass voters to determine their likelihood of voting favorably. The results can be fed into the computer file at regular intervals so that on election day volunteers, operating from a print out of names and addresses can "pull" favorable voters.

Politicians have always rated the electorate in this way, but in the past non-machine politicians have had to rely on their general judgments of districts. If a campaign manager assumed that a given district will, for whatever reason, vote overwhelmingly in favor of his candidate, the natural reaction was to concentrate organizational efforts in that district.

Computerization makes it possible to treat each voter and his voting preference individually, as part of what amounts to a universal poll.

And that isn't all. A good program can provide lists of undecided voters whom the campaign's noise might affect, analyses of the electorate and its component parts to facilitate decision making, a basic family mailing list for printing labels, records of identifying characteristics such as profession and type of residence, geographic sorts for specialized mailings, etc. The only limitations to the information that can be stored and retrieved with speed and ease are the number of characters on the computer card and what you intend to do with the file.

Naturally, the larger the campaign, the more unwieldy a computerized canvassing operation becomes, but campaign managers often treat anything larger than a Congressional district in semi-autonomous segments. Entire states have been canvassed in this way.

In effect, then, computers have insinuated themselves into modern politics as much as they have into every other phase of life in the 1970's, from how we receive our pay checks to how we spend it and everything in between. And for the same reasons. It's simply easier and faster to deal by computer with the numbers and complexities of what has in the past been done manually. The political computer specialist takes what the politician has always done and does it more quickly and more accurately to allow campaign planners and directors to maximize the impact of the time, money, and resources at their availability.

"Before attempting to determine the country's next president, I suggest you try something comparatively simple, like who the class president will be!"

An Analytic Examination of Creative Computing

by David H. Ahl

Introduction

Creative Computing has an untarnished reputation as an impartial journal concerned with the most fundamental issues of computers in education and one which welcomes contributions from all comers. Yet it appears that *Creative Computing* is so concerned with its galactic responsibilities that it has overlooked the constituent elements of the words between its covers. Without these words, *Creative Computing* readers are speechless. Words become part of phrases, phrases become part of sentences, sentences become part of articles, and so on.

Therefore, it is imperative to examine the content of the words in *Creative Computing*. *Creative* can not rest on its laurels but must be subject to analytic examination to prevent any distorting biases from unwittingly creeping by the editors.

Method

A random sample of words was chosen from a random sample of issues of *Creative Computing*. The letters in each word were classified according to an ancient Indo-European System. For example, an "a" in any given word was tallied under the column heading "a," whereas a "b" was tallied under the heading "b," etc. In a sample of 1709 words, the following distribution was obtained:

a	1723	h	283	o	617	v	38
b	339	i	1605	p	271	w	172
c	395	j	62	q	49	x	12
d	549	k	216	r	445	y	228
e	2283	l	327	s	438	z	6
f	117	m	172	x	382		
g	161	n	494	u	1173		

Results

Mean usage of letters varies from 23.724% for the letter "e" to .069% of "z." This is attributable in some degree to the corollary finding that *Creative Computing* authors tend, all other things being equal, to prefer more words containing "e's" than "z's." More research on this point is urgently needed to uncover the reasons for this preference. The magnitude of this difference, significant beyond the .00001 level of confidence, raises the unwelcome specter of some underlying literary bias that can no longer remain unnoticed by *Creative Computing* editors.

From a correlation matrix, other biases of *Creative Computing* authors were uncovered. For example, of the 49 instances of a *Creative Computing* author employing the letter "q" he invariably prefers to follow it with a "u." Also, such letters as "i," "b," and "m" are frequently used consecutively. However, we seldom find the sequence "x," "d," and "s." Why should such glaring biases exist in a journal noted for its fairness and neutrality?

Conclusion

You, the readers of *Creative Computing*, must demand from the contributors and editors a more uniform distribution of the most basic elements of the words used in *Creative Computing* which so profoundly influence its entire content and outlook. Use your computers to keep track issue by issue of whether a more homogeneous distribution is being obtained. Cojxr sqally kibmz!

(Portions of the above article were plagiarized from an article in *The Journal of Irreproducible Results* by Alvin Howard, PH.D. My apologies. —DHA)

By now you've read about the National Computer Conference (June 7-10, New York) in *Computerworld, Datamation, Computer Notes,* and all the rest. But while our reporting is totally untimely, it brims over with poignant human drama and brings you reality as it really was. Keen! —DHA

How We Spent Our Summer Vacation

John Lees, with Richard Freeman, Dennis Keats, and Susan Culwell

Following is the story of how four students came to leave the security of the midwest and journeyed to the Big Apple for the National Computer Conference, June 7-10. It is a true story. Any resemblance to fictitious characters is accidental.

-Prelude-

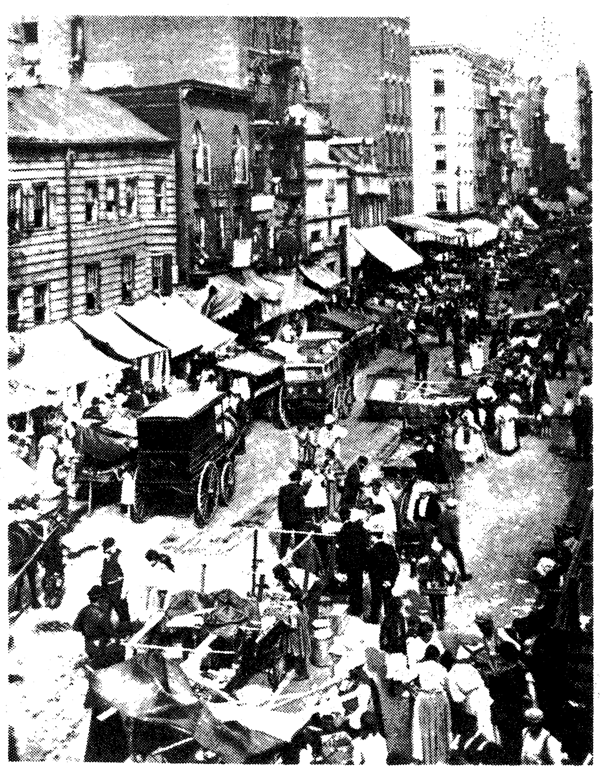
One fine day in January, John was talking on the phone with David Ahl about the magazine and things in general and stuff. In the course of the conversation Dave said, "You're coming to NCC of course?" Caught totally off-guard, John replied, "Uh, NCC?"

"You know. The National Computer Conference in New York, in June? *Creative* will have a booth there."

"Uh, yeh. Uh, I hadn't really thought about it; it does sound interesting. I'll look into it and let you know."

"Okay, If you decide to come you can stay with me and you can even work in the booth. I'd like to meet and talk with you."

The conversation then degenerated into other things but *the crucial idea had been planted.* The subject of NCC was to lie dormant for several weeks until John committed himself by paying the sixty dollar advance registration fee. At that point he became chicken about going all the way to New York by his lonesome, so he took advantage of his position as resident weirdo in the UMR Software Lab to recruit some other people who enjoyed a similar spirit of adventure and lack of common sense. A suitable notice was promptly posted on the bulletin board above the correspondence with Gregory Yob (author of Wumpus and known to refer to the Software Lab as a "zoo") and below the designs for the **S**imple **O**perating **S**ystem.



Big Apple 1899.

Luring people with the promise of an Educational and Enlightening Experience, the chance to meet world-famous DAVID AHL, and above all the opportunity to go to New York *cheap* and miss the first week of summer school, John managed to catch the interest of four undergrads. Time passed. Various means of getting New York were discussed and discarded (the Comp-Sci Department was less than enthusiastic about financing a week long bash for some of its flakyer students). Finally it was decided that the group would travel in Susan's ten year old Chevy. This also pretty well established Susan as one of the group. During this time Dave was asked to extend his original offer of a place to stay to first one, then two, then three additional people. How could he refuse?

The spring semester came to a close with nothing very definite decided beyond the fact that some number of persons would be leaving for NCC on the fourth of June. John and Dave had been exchanging letters completely out of phase with each other, so Dave probably never really knew what to expect. After a short respite in Tulsa, John returned to Rolla late in May, finding under his door a cryptic note on a scrap of brown paper which could only be interpreted as meaning that Sam could not get away from his co-op job long enough to make the trip.

Dave managed to be everywhere but at the other end of the phone for a week, then was finally reached on May 28. One of the out-of-phase letters had included the information that a Tektronix 4051 Graphics System was being loaned to *Creative* for use in the booth, but alas, had no creative-type demonstration programs. It was arranged that the group from Rolla would show up early to write some creative-type demos.

In order to get to Morristown on Friday, June 4, it was decided to leave at five ayem on Thursday, driving the 1200 or so miles in two days. (Not normally any great trick for four students, but only two of these four had driver's licenses.) After two days of trying, John finally reached Dennis, who lives in a one-horse town in northwestern Missouri which evidently still has a wind-up phone system, and told him the departure date.

Like the reasonable, sensible people they are, the group went out for pizza the night before leaving and all got to bed by one ayem except Susan who didn't go to bed at all.

-The Trip-

Susan drove around and collected everyone beginning at fourthirty ayem. We threw our gear in the trunk, buckled our seatbelts and set off in the direction of St. Louis. Sunrise reassured us that we were in fact heading east, so Dennis and Richard fell asleep while John saw to it that Susan stayed awake and kept the car on the road.

Several hours later the group stopped at a grocery store and purchased orange juice and donuts for breakfast, which were consumed at a rest stop several miles further on. Somewhere after breakfast the car began acting funny but it was put down to Dennis not being familiar yet with the car. However, upon restarting the car after gasing up in Terra Haute it sounded like a buzz-bomb, so Susan drove into the first Midas Muffler repair shop she could find.

Fortunately only a gasket was required and the interstate was soon again speeding by at fifty-five. It was decided to shoot for Cambridge, Ohio that day, so reservations were wisely made ahead and Cambridge reached around eight pee em. Dinner was obtained in styrofoam containers at the local McD****d's. The sensible thing to do would have been to go to sleep immediately, but Richard turned out to be a Mary Hartman freak, so John, Dennis and Richard sat up until midnight warching teevee and finishing the orange juice.

Rising at five the next morning the group soon forged into scenic, smog covered Pennsylvania. Tired of the vista offered by I-70, it sounded like a good idea to drive off into the rustic Pennsylvania countryside in search of breakfast.

Forty-five minutes later we heaved a sigh of relief when the good old interstate was found again before the car ran out of gas. Breakfast was obtained in styrofoam containers at you-know-where. The next seventy miles were downhill on the famous Pennsylvania Turnpike. We agreed unanimously that tunnels are neat and that Missouri really should get a few.

The day went on and so did the highway and the trucks and the smog. Bethlehem was picked as a quaint place to get lost in while searching for a place to eat lunch, but this was becoming old-hat and a fast-food emporium was located soon after regaining the good old interstate. Continuing a steady pace for the remainder of the afternoon, Morristown was reached exactly at the beginning of the rush-hour.

Now eastern traffic is a little heavier than that of rural Missouri, so Dennis concentrated mainly on not getting hit by the other insane motorists. Of course it was soon discovered that Morristown was on the *other* side of the interstate, but that was easy to fix and after driving all the way through Morristown and back looking for a shopping center with a phone booth the car came to rest in a school parking lot. Susan had been sitting in the back seat gritting her teeth and pulling her hair at all the things which had almost happened to her car. She now took over from Dennis and Dennis gritted his teeth and made that assortment of noises that only Dennis can make in the back seat.

We were looking for a shopping center with a phone because we wanted to buy some postcards and call David for directions to his house which he said, "You'll never find if I don't tell you how to get there." Since the streets of Morristown appeared to be in the pattern of a mobius pretzel, we were quite prepared to take his advice. So we promptly got lost again and ended up on a road obviously headed out into farmland. Turning around was the clever thing to do, so Susan took the first right and then a few more turns trying to get back to the road we had been on, and succeeding only in getting lost in the suburb we had blundered into, when John suddenly said, "HOLD IT!"

Screech. "Huh, what?"

"That mailbox had 'Ahl' on it."

"Oh, come-on. No way. Putt-putt in reverse.

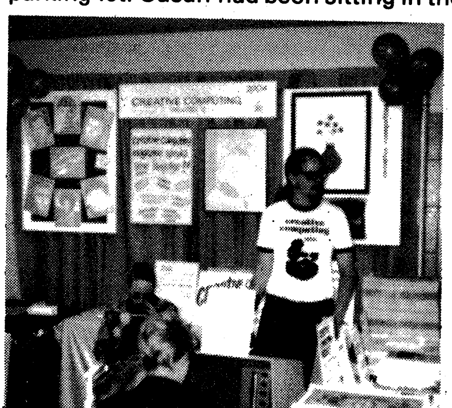
"A . . H . . L. The right housenumber, too. What street are we on? This can't possibly have happened. The odds don't **exist!**" But it had happened. We drive 1200 miles and found David Ahl's secluded house by accident. Just programmer's luck.

-NCC and the Big Apple-

We stayed with Dave and Sandy Ahl for two days, occupying ourselves with programming the Tektronix 4051, sleeping and such. Working with the 4051 was a little aggravating because we didn't have a hard copy unit. In spite of that drawback we managed to get some decent graphic demos and games going. The group was accorded great honor by being let into Dave's inner sanctum, where each issue of *Creative Computing* is born. The fabulous "Artist and Computer" issue was just off the press and we also got to see much of the material being pasted up for the Sep/Oct issue.

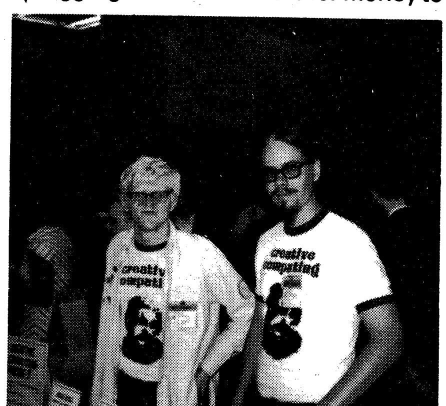
Sunday morning the Tektronix was loaded into the cars along with many (but not enough as it turned out) catalogues, magazines, books, posters and general stuff. Susan, Richard and Dennis then followed Dave and John into New York City. By some strange fluke of luck both cars arrived in front of the Coliseum without mishap. It hardly need be mentioned that the weather, until then fair and sunny, had turned overcast and rainy. This gave rise to a first impression of New York City as something which had condensed out of the smog.

The stuff was carried from the cars in the front of the building and up the escalators to the second floor. This was done during lunch in order to avoid paying union laborers double time to carry it in as all exhibitors were supposed to do. We often bent such rules during the show or we would have been reduced to begging on streetcorners for money to



Susan Culwell, Dennis Keats, and John Lees at the *Creative Computing* booth at NCC during a rare, uncrowded moment.

Typical mob scene at the *Creative Computing/Byte* booth at NCC. Over 10,000 people came by.



Richard Freeman and John Lees peddle *Creative Computing* merchandise in style.

Al Stahl, Popeye animator dropped by our NCC booth and whipped off a quick sketch for us.

leave the place. From the point of view of a small magazine, the costs associated with exhibiting in the Coliseum are atrocious!

The *Creative Computing* booth was on the second floor next to the exit to the escalators to the third floor. Not a bad location at all. During the first two days we shared the booth with the people from *Byte* magazine (another bunch of weirdies). We also handed out information for Computer Mart of New York and we had a Hoboken Computer Works IMSAI 8080 running a TV Dazzler as an additional attention getter to the Tektronix. All in all, the booth looked pretty good. We even had balloons.

After setting up the booth and taking a cursory look at the other exhibits Dave and Richard went off to an exhibitor's party while the other three of us walked to the hotel to our eight floor forty-two dollar a night room with a scenic view of garbage in the streets and heavy particulate matter in the air. It was discovered that the sleeping bags and some books had been left in the car when checking in earlier, so we set off to get them.

The car turned out to be at the very bottom of the parking garage, just above the pump which kept the whole place from flooding. There is very little oxygen at the bottom of a four-level parking garage and the trio was almost happy to regain the eighth floor. We attracted some rather strange looks tramping through the plush hotel lobby, past the bell captain and the dining room and all, dragging a load of camping gear. At least we didn't have to be worried about being mugged for our luggage.

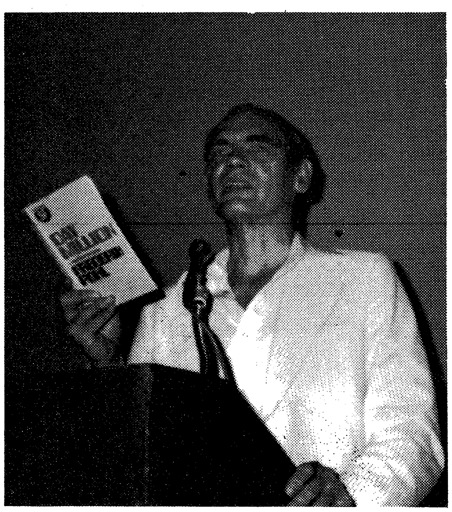
Back in the room the first thought was, "something to drink!" Fortunately there was a soda machine in the hall just outside the door. Unfortunately the machine wanted to be fed 45¢ for each can of soda. Hrrmph. Back home it was 20¢ per *bottle*. Oh, well. When in New York ... The ice machine on the floor didn't work, neither did the one on the seventh floor. The sixth floor had some very wet ice. At least it didn't cost any more.

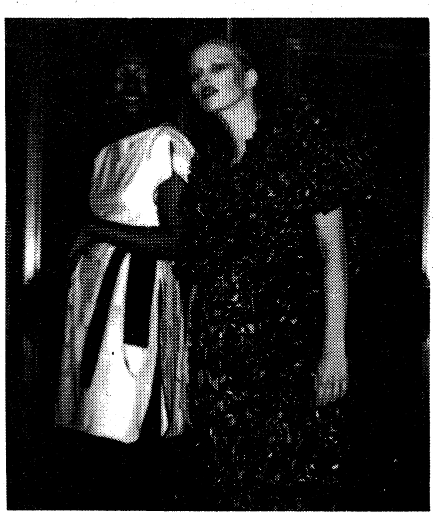
Luxuriating with the obscenely priced soda pop, the question of dinner began rumbling through our stomachs. The room service menu offered a cheeseburger for $4.20 and coffee at 80¢ a cup. Yipe! Discard the room service menu and forget the hotel restaurant. There was a McD****d's somewhere in the vicinity, but it was now getting dark in the crime capitol and we didn't want to get mugged looking for a Quarter Pounder, so we had some more soda for dinner and wished we had the sense to buy a three pound jar of peanut butter before we left civilization.

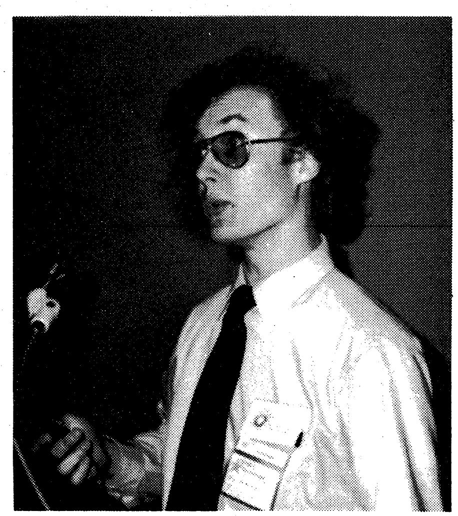
Eventually Dave and Richard showed up. They had fared much better in the food and drink department, especially in the drink department.Dave went off to his room and the rest of us discussed New York and listened to Richard's tales of the antics of tipsy exhibitors. Eventually we parceled ourselves out over the room and went to sleep.

The following two nights Susan slept with Debbie Luhrs of *Byte* at their hotel while Bill Mayhew of Boston Children's Museum slept with us. Bill was also heard to make the comment that we resembled a zoo. This was when Dennis and Richard were using a bed as a trampoline and throwing pillows at each other and Susan and John were writing out loud a letter to Pamelia back in Rolla. To be completely fair about the matter, we feel that Bill still has the potential to be a real weirdo if he'll just let himself go. The fourth night we were all together again in one happy, secure group. Susan was glad to get out of *Byte's* cheap hotel, only twenty dollars a night with no soda machine.

The four days of manning the booth in the Coliseum were, to say the least, tiring and hectic. Each day began with breakfast at the styrofoam emporeum and often ended with dinner at the same establishment. Lunch was 85¢ hotdogs at the Coliseum. Work in the booth consisted primarily of standing in one spot and forcing (politely) people to accept a


Frederik Pohl reads a quote from "Day Million" about computers in the far future—he expects them to provide everything, even sex.


The latest in computer fashions: a magnetic tape gown.


Wild Bill Mayhew of the Children's Museum in Boston describes their extensive computer program for children at an NCC session on Public Access.

copy of our free catalogue. We passed out some 7000 catalogues before the show was over and we ran out early at that. Dave twice had to drive back to Morristown to get more supplies.

The *Creative* booth attracted quite a lot of attention. We were not your usual booth handing out your usual blase sales literature, and most people reacted favorably to that. Dave was even interviewed by the Soviet news agency, Tass as well as the New York Post and others. A number of people from the Tektronix exhibit came over and played the games on our 4051. Evidently their exhibit was too serious for that kind of thing. Interest was even shown in us by IBM. Gee. Of course a lot of kooks stopped by, many of whom said they were subscribers.

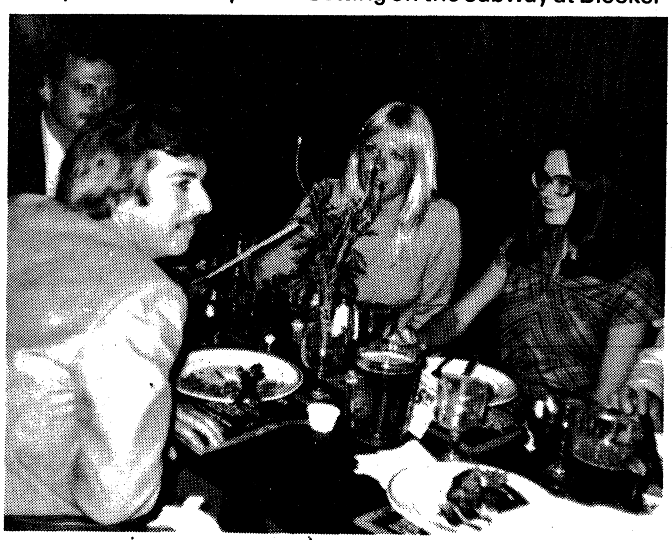
We spent some time walking around looking at the exhibits, picking up literature and physically seeing machines we had hitherto only read about. Some of the more interesting things to see were the National Student Computer Fair, the Computer Graphics Art Exhibit, the networking demonstration, the Fourth International Computer Art Festival, the Computer Science Film Theater, and the booths of such well known companies as MITS, General Turtle and Vocal Interface (VOTRAX). In dull moments simply looking at the people always proved interesting. Estimated attendance was 35,000 plus.

One of the more interesting conference sessions we attended was Richard Speer's session on Computer Generated Films. Several really fascinating films were shown, including a mind-blowing one which contained a graphic depiction of mathematically turning a sphere inside-out. Dave hosted a session of Public Access to Computer Power and had a paper published in the Conference Proceedings. As usual some of the sessions, despite interesting titles, proved to be unremittingly dull and we even walked out on a couple.

A highlight of our stay in New York City was the night that Dave took us on the subway to ride the Staten Island Ferry and then to a little restaurant in The Village for a superb dinner. On the ferry we actually got a whiff of decent air and in looking back at Manhatten Island realized just how incredibly filthy the air over New York City really is. The graffiti covered subway trains, although ear-shatteringly noisy, were rather quaint. Getting off the subway at Bleeker Street (memories of Simon & Garfunkle) we walked to Bedford and went through an unmarked door into Chumley's, a restaurant which Dave assured us was one which retained the Bohemian flavor of The Village as it used to be. The meal was excellent although Richard, Demmis and Susan embarrassed Dave by ordering their London Broils well done. John, who grew up in a moderately large city, was cultured enough to order his rare.

Thursday, the last day of the exhibit, dragged on. We ran out of catalogues shortly after noon and then just stood around talking to people who came by the booth. As six o'clock approached, Dave went to get the car while the rest of us packed up the booth and carried the stuff down the back stairs, talking our way past guards who wanted to see passes for all the stuff we were removing. John pointed out that it would be silly to write ourselves passes and kept walking as he was talking. Fortunately we didn't have much left at that point and soon we were standing on the sidewalk with a little pile of leftovers when Dave drove up.

We were all greatly relieved to get out of the city and would not have particularly disturbed if New York had sunk beneath the waves as soon as we were through the Lincoln Tunnel. We sped for Morristown, stopping to eat along the way. Following a few hours discussion of the high points of the past five days, such as almost losing all the subscription records when the maid cleaning the room threw them away while we were checking out, we fell into exhausted slumber.



Speakers from the NCC Public Access to Computers session joined in Beefsteak Charlie's to have a few. L to R: Robert Smith, Privacy Journal; Ron Anderson, Univ. of MN ("Computer Cartoons," *Creative* 1:3); Trinka Dunnagan, Univ of IA (Technical Transport Problems, *Creative* 1:6); Burchenal Green, Editor of *Creative*.

-The Return-

The trek back to Missouri passed without incident of great note. The group took two nights to return since we were all fatigued and even getting a little tired of each other. It is true that there were some interesting points - pizza and donuts in Sharon, the ozone alert covering Pennsylvania, Ohio and Indiana - but they don't amount to much. Suffice it to say that our eleven day trip finally came to an end and we all re-entered the routine of the University.

THE END

Press On

Nothing in the world can take the place of

Persistence

- o Talent will not—
  Nothing is more common than unsuccessful men with talent.
- o Genius will not—
  Unrewarded genius is almost a proverb.
- o Education alone will not—
  The world is full of educated derelicts.

Persistence and determination alone are omnipotent.

From a McDonald's Ad

TOWARD THE ELECTRIC SYMBOL

by Robert E. Mueller*
Britton House
Roosevelt, NJ 08555

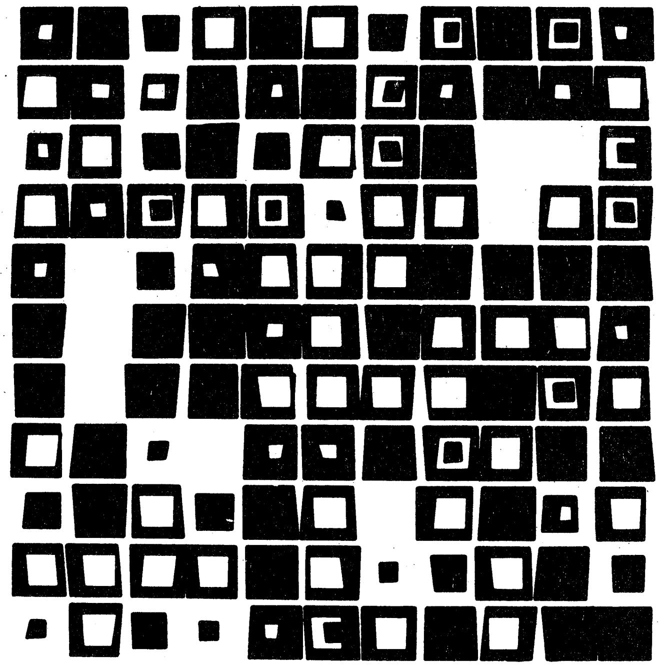
Computer art is conditioning us for a radical new way to think. I call this new process the electric symbol, an entirely new medium for future mathematics and science. But it begins as art. The play instinct of the human mind takes up any novelty and plays around with it for eons before it can take on human meaning. Computer art is at a very early stage of the evolution of the electric symbol.

For example, long before Rene Descartes conceived Analytic Geometry mathematicians played around with geometric patterning that appeared to have little mathematical or scientific significance. The Greek preoccupation with conic sections and infinitely-divisible geometries led to many pre-mathematical insights — calculus was trying to burst out of their thinking, but it had to take many years before it saw the light of day as a strict discipline. Geometers in Descartes time invented many curves for the sake of pure beauty. They even named them: Witches of Agnesi, Devils on Two Sticks, Hippopedes, Cocked Hats, Anallagmatics, Cissoids, Foliums, Horopters, Loxodromes, Pseudo-versieras — all names we could easily give to modern computer art. Later all of these curves took on scientific significance in dynamics, magnetism, and electric circuits.

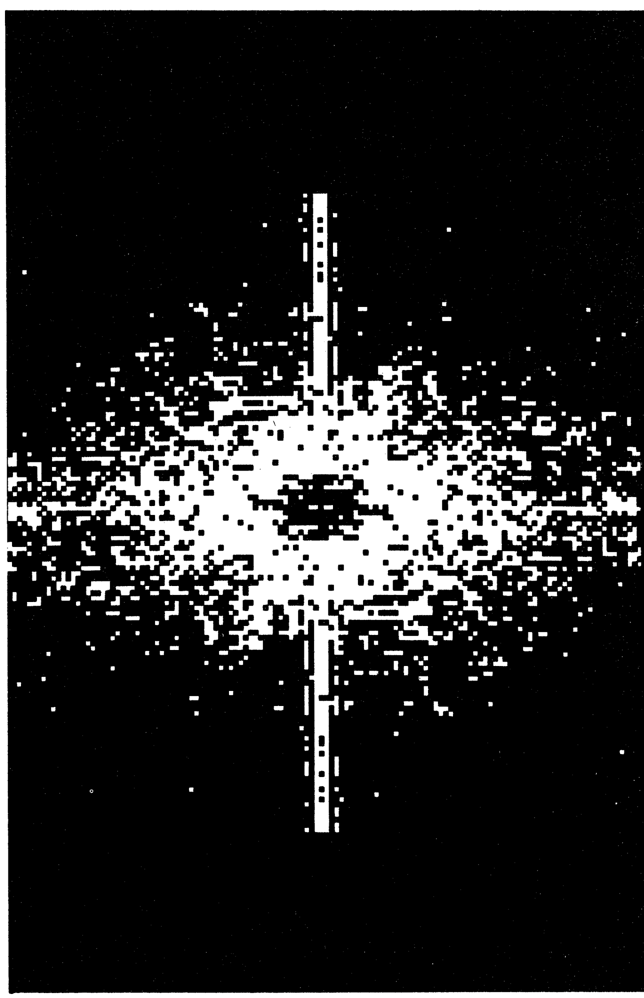
We do not yet understand where our computer art is leading. This is because, I think, we have not yet assimilated the vast array of visual forms made possible with the computer. As art — that is serious visual art like Leonardo's drawings, Rembrandt's etchings, Picasso's abstractions — computer art is little more than elaborate design. [See my article "Idols of Computer Art" in the May 1972 issue of *Art in America* if you want to pursue this suggestion further.] But as a precondition for future mathematics and a new mode of electric symbolization, I think that computer art is extremely important.

Imagine, for example, a future mathematician or scientist bent on mathematical theorization sitting down to a video-computer CRT console with a probe as his pencil. He no longer writes mathematical equations on a two-dimensional surface, using a pencil. Instead he writes *into* a visual space whose geometry can be very complex. His symbolization is not formed out of letters or numbers; out of simple marks such as plus, times, integral signs, or matrices limited by the pencil-and-paper medium. Instead he has the complete electric freedom to invent new multi-dimensional symbols in a deep video space — it can even partake of the entire color spectrum!

Symbolization can proceed in a new way, and computer art forms may ultimately provide keys that will help unlock these new ways to symbolize. The play of computer forms, the designs and beautiful doodles of computer art, take on certain recognizable characteristics. As you look over the art in this book**, try to categorize the art yourself: symmetries within symmetries, sinusoidal variations, planes in movement, multispatial flows, and so on. The future use — the mathematical evolution of a truly novel way to symbolize in this electric medium — will depend upon your conditioning to them. Perhaps you will be able, like Descartes, to call them to mathematical and scientific account, and help give birth to the new electric symbol of the future.



"Computer Icone" by Vera Molnar



---

*Robert Mueller is the author of the book, "The Science of Art," Day, New York, 1967.

**"Artist and Computer" edited by Ruth Leavitt. Available for $4.95 plus 75¢ handling from Creative Computing Press, P.O. Box 789-M, Morristown, NJ 07960.

Producing Computer Poetry

by Margaret Chisman
50 Tuddenham Road
Ipswich, Suffolk, IP4 2SP
England

Ideas mill around vaguely for some time. I collect words that appeal to me visually, orally or semantically without any special theme in mind. This phase corresponds to the 'data soup' of scientific problem solving and can last for months. Sometimes the fermenting process erupts under its own volition, or under the jerk of an outside impetus — such as the Computer Arts Society writing to me about a forthcoming exhibition at the Science Museum, London, of computer generated art.

I produce, as a first step, a verse, which is generally the kind of apparently meaningful nonsense that Chomsky refers to about green ideas sleeping furiously. I use for the parts of speech I desire whatever words in that category first come into my head — this is to get the lilt or rhythm of the verse. For example: —

Should fancy free us from technique
Or belief lead us to provocative faith
Should loving add to our pattern of crime
When all living is dead.

or another

My heart looks at life
Thrusting with growth but true
Always giving never having
Agony though softened corrupts.

I then break it down as follows: — (taking the second verse)

	(1)	(2)
My	heart	looks at
(Fixed)	*noun singular subject*	*verb present tense singular*

(3)
life
noun singular object

(4)	(5)	(6)	(7)
Thrusting with	growth	but	true
verb present participle	*noun singular*	*linking word*	*adjective*

(8)	(9)	(10)	(11)
Always	giving	never	having
linking word	*verb present participle*	*linking word*	*verb present participle*

(12)	(13)	(14)	(15)
Agony	though	softened	corrupts
noun singular	*linking word*	*verb past participle*	*verb present singular*

Then in each group I list say 8-10 words (limiting myself to about 10 because of the time factor — finding 10 words that please me, in each group can take up to a week.) I use Rodale's 'Synonym Finder' and also his 'Word Finder' plus Chambers Twentieth Century Etymological Dictionary. I do not make any attempt at rhyming as, in my experience, this would not only make my work immeasurably more difficult but amateurish rhyming can be bathotic. However there is place for it in humorous verse.

The criterion for the selection of words in each group is generally overall compatibility of meaning, mood and category but with an occasional word quite out of keeping to provide a jolt. Each person has their own repertoire of words, and writing this kind of poetry it is easy to get into a semantic rut. So I flip the dictionary open randomly and follow up any interesting words in the Synonym Finder.

In the early days of my work I used a Telcomp programme with a random number generator to select from each group. I eventually decided this had less than an artistic result in that words could, and were used over and over again, thus reducing the impact. Quite by chance, I made an improvement. One day I could not get on a terminal and so, impatiently, I wrote the words on pieces of card and put each group of words in a separate bag, taking one word out of each bag in order. It struck me then that it would be better poetry if, after taking a word out I did not put it back again (as the computer does in its random number generator). Thus I had x number of verses corresponding to x number of words I put in each group and there was no repetition. Other sets of verses can, of course, be produced, by putting the whole lot of the words back again in the bags and starting all over again.

I believe several other computer poets have discovered they can simulate the action of a computer thus. The contribution of the computer (or a 'computer approach') is to suspend judgment on conventional patterns of association.

I deliberately never use punctuation in computer poetry because its absence allows greater varieties of combination of meaningful groups of words.

Groups of words used for verses that follow

Slot 1	Head, eye, hand, reason, heart, child, flesh, mood.
Slot 2	Thrives on, quickens with, delights in, shades into, leads to, craves for, yearns for, prepares for.
Slot 3	Pain, grief, rejection, enchantment, caresses, life, joy, prayer.
Slot 4	Unseen by, bleak with, eager for, strong with, free from, fresh to, ardent with, blind to.
Slot 5	Guilt, doubt, despair, sin, pride, lust, dread, lies.
Slot 6	Maybe, yet, even, but, now, never, perhaps, always.
Slot 7	Vague, true, calm, dull, soft, cruel, fierce, vain.
Slot 8	Not, rarely, beyond, above, sometimes, seldom, just, only.
Slot 9	Relaxing, hoarding, quarrelling, blaming, probing, seeking, hiding, drifting.
Slot 10	Repeat slot 8
Slot 11	Seducing, aching, dreaming, doubting, pitying, loving, daring, stifling.
Slot 12	Comfort, sorrow, friendship, freedom, passion, praise, remorse, truth.
Slot 13	Started with 'though' and changed to 'if' as it was more meaningful..
Slot 14	Controlled, withheld, followed, enforced, enjoyed, derided, applauded, suppressed.
Slot 15	Corrupts, consoles, refreshes, enslaves, endures, destroys, divides, consumes.

MYSELF MANIFEST

Margaret Chisman, 1974

1st Selection

My head thrives on pain
Unseen by guilt
Not relaxing not seducing
Comfort if controlled
Corrupts

My eye quickens with grief
Bleak with doubt yet true
Rarely hoarding rarely aching
Sorrow if withheld
Consoles

My hand delights in rejection
Eager for despair ever calm
Beyond quarrelling beyond dreaming
Friendship if followed
Refreshes

My reason shades into enchantment
Strong with sin but dull
Above blaming above doubting
Freedom if enforced
Enslaves

My heart leads to caresses
Free from pride now soft
Sometimes probing sometimes pitying
Passion if enjoyed
Endures

My child craves for life
Fresh to lust never cruel
Seldom seeking seldom loving
Praise if derided
Destroys

My flesh yearns for joy
Ardent with dread perhaps fierce
Just hiding just daring
Remorse if applauded
Divides

My mood prepares for prayer
Blind to lies always vain
Only drifting only stifling
Truth if suppressed
Consumes

2nd Selection

My flesh leads to rejection
Free from despair never soft
Beyond hiding beyond dreaming
Remorse if applauded
Divides

My child thrives on grief
Ardent with pride yet true
Just seeking just dreaming
Truth if controlled
Corrupts

My heart quickens with enchantment
Eager for dread now fierce
Rarely blaming rarely seducing
Comfort if withheld
Enslaves

My hand shades into life
Unseen by sin always vague
Sometimes releasing sometimes stifling
Freedom if enjoyed
Refreshes

My reason delights in caresses
Bleak with lies maybe cruel
Not quarrelling not doubting
Friendship if derided
Consoles

My eye prepares for pain
Fresh to lust but vain
Above probing above loving
Praise if followed
Endures

My head craves for joy
Blind to guilt perhaps calm
Seldom hoarding seldom aching
Passion if enforced
Destroys

My mood yearns for prayer
Strong with doubt ever dull
Only drifting only pitying
Sorrow if suppressed
Consumes

Writers and Computers: An Interview With Carole Spearin McCauley

by Cathy Silverstein
24 Kolbert Drive
Scarsdale, New York 10583

When the words "computer-assisted literature" are mentioned, one doesn't know what to expect. Carole Spearin McCauley is a youthful, soft-spoken Connecticut woman who is one of a handful of authors using the computer to generate prose.

Ms. McCauley was educated as a literature major and subsequently entered professional journalism and creative writing. But after attending various computer and information processing conferences, she sought a way to combine both her job and her interest in technology. She wondered what ways data processing could aid writers involved in literary projects. Six years ago she found what she calls the "gestating" field of computer-assisted literature and through it another way of creating fiction.

Ms. McCauley then began to research various systems but more importantly she sought a "user-friendly" language to use in her programs. She chose APL because of its ability to process both character and numeric strings as well as having considerable editing and formatting capabilities. She learned to program the hard way — by reading inadequate manuals and by experimenting with programs at the terminal. Unfortunately her facility is not equipped to take full advantage of the interactive nature of APL.

"Sex and Violence", her first project, combined both language content and graphics similar to poetry. It involved the generation of phrase substitutes from word lists. The final results were published in an anthology, *Assembling*, edited by Richard Kostelanetz.

Though some may view the preceding as trivial, it must be considered a major advance in the field of computer-assisted prose. As Ms. McCauley states:

> "It (computer prose) must form some kind of whole or people will feel they've been cheated. It either has to be short enough to stand on its own or it must be incorporated into some larger work to help the reader make sense of it.
>
> I don't like to write things that have no purpose. Frankly, things are so damn hard to do, that they deserve a context.
>
> I've found my fiction won't be accepted just because it exists. It must offer someone something in the way of ideas or needs. It doesn't do the cause of computer literature any good to come up with nonsense. I don't want computer literature to get a bad reputation from what I've done."

After "Sex and Violence" she progresses to the stanza form. This project in 1970-71 involved template lines, skeleton format and three categories of mood words (positive, negative and ambivalent). These stanzas were later incorporated into the novel, *Happenthing In Travelon*, she'd begun to write. The novel published in 1975 by Daughters, Inc., a feminist publisher, led Ms. McCauley to what would become the two areas she explores using the computer. Initially she sought another manner of presenting meaningful characters through computer-assisted portraits beyond the usual narrative method ("he said ..... she said .... then they ...."). Secondly she used the computer as a design tool, to add greater dimension to her fiction.

In 1971-72 Ms. McCauley was contacted by Prof. Max Vense and Elisabeth Walther from the University of Stuttgart, Germany. They asked her to write a book, composed of computer-assisted works. *Six Portraits* (1973) was included in a series they published. While working on this book, Ms. McCauley began to realize the first of her goals — the creation of new methods of characterization. This project produced a set of sentences dealing with the protagonists of her developing novel in addition to anagrams of word relations assigned to them. In her words:

> "The computer makes the characters more three-dimensional in a way that is faster and easier than being totally alone in your head. By making connections the writer hadn't seen, it offered an opportunity to meditate on the nature of each character's growth and consistency."

Lest you believe this "miner" has struck a vein of gold, Ms. McCauley has encountered great difficulties when seeking a market for any fiction either totally or partially computer-generated. The hostility shown by many New York publishers has further impressed her with the need for imaginative and well-planned programs. Ms. McCauley thereby plays devil's advocate by constantly evaluating the role of the computer in literature. Can computer fiction stand on its own? Will it make its own statement?*

This author sees the field of computer-assisted literature as a method of extension, much as the car was originally an extension for the horse. She looks forward to a time when a terminal will share equal rights with a television set in the home. In fact, Ms. McCauley believes common use of the computer to be necessary before computer literature will be more widely recognized.

The major drawback Ms. McCauley has encountered is the inability of the computer to adequately express emotion. She anticipates a "change in the traditional goals of programming toward more humanistic and imaginative applications." Finally she feels that greater experimentation with and use of the computer will engender integration of computer science and literature to "help people to remain generalists."

Ms. McCauley sees many areas for expansion and improvement of both form and content in the expanding arsenal of tools employed in computer-assisted fiction. The nature of random composition, literary criticism and generation of the most stringent literary forms such as the sonnet are only a few possible areas. As an author and as a programmer, she yearns for her work to be taken seriously so that the "presence of computer literature in or as a work of fiction will be viewed on equal terms" with both traditional and experimental modes.

Ms. Carole Spearin McCauley and her work show that contrary to often voiced fears, we are not doomed to 'narrow' specialization. In this case, a computer, the once-exclusive brainchild of science, engineering and business, has become an integral part of one of our most humanistic fields, literature.

---

*One New York publisher, Praeger, was impressed enough with the manuscript and computer parts of *Happenthing* to give her a contract to do a non-fiction book. This resulted in *Computer and Creativity*, published by Praeger in late 1974.

Once Upon A Computer...

by Carole S. McCauley

The computer as novelist and poet? Vladimir Nabokov Model 360?

As with computer art and graphics, the very idea can disturb or amuse people, including some computer company employees, because it upsets traditional myths about how art or literature are created.

As a writer, I'm a veteran of eight projects in computer prose, totalling hundreds of pages, which I have used in a variety of ways. Besides their literary value, my projects (especially the APL programming!) have tended to teach me what won't work — and why — rather than the joy of celebrating what does work.

Like any other tool or machine, the literary computer may be used oversimply (to produce something that can be done by hand or typewriter) or uncreatively (to produce nonsense or to reproduce something already done). Let me briefly illustrate each of these problems and how they interconnect, since what is simple-minded is probably uncreative, too.

Sometimes these faults can be caused by innocent ignorance because the writer usually is not his/her own programmer, especially at the beginning. S/he must depend on a programming partner willing to work with poetic or otherwise unusual material never seen before. This partner, or the writer after a programming course, must get the data to run through the machine and print out at the terminal without "bugs" (errors).

Another principle is that the machine, while able to make many rapid calculations, is totally dependent on correct data and commands down to every comma and apostrophe. Helpless without them, it has no judgment or ability to proceed independently. The machine can spot an error but not correct it.

1. The literary computer may be used too simply. An example: just feeding it a list of words (*Jesus hotdog freak fruit,* etc.) without syntactical instructions or doing anything further to develop an idea. This means accepting however the machine may churn them out (*fruit freak hotdog Jesus*) for the (possible) humorous results.

The machine also produces handsome design poems, can take a few words or letters and print them in a pretty pattern on nice white paper. So could the poet Guillaume Apollinaire sixty years ago — and so can most people with a typewriter.

2. The machine may be used uncreatively, in my opinion, to produce lines like

O Death . . .
The night
Comes and shines . . .

The river
Winks
And I am ravished.

O night,
Weep like a red flower . . .

O darling,
Dance like a transparent moon . . .
Sink, O darling! . . .

O poet,
The body of your blessing reaches me . . .

Where did these words come from? Their author, who selected such lines from printouts of a computer project at Yale, says, "Typically, 25 words of a vocabulary were taken from an anthology of classical English poetry beginning with the 16th century . . . I took another 25 words from an avant-garde anthology published in the late 1960's." Completing this project was, I know, no small or simple task. It entailed, for instance, 19 different vocabularies of 50 words each, which the machine combined and interchanged by random number generation into "two stanzas a second or a theoretical 7,200 stanzas an hour."

If a writer enters something, s/he gets something out that sounds and looks like poetry. As any lit major knows, however, lines like these have already been done by many romantic poets. The process resembles using a Moog synthesizer to re-produce Beethoven's romantic symphonies.

When an author can avoid the above difficulties, there remain technical and graphics problems. One is getting copy printed dark enough on paper white or good enough to photo offset successfully. Another is getting the machine to repeat itself — to print out two originals that are exactly alike. If the machine process used is random number generation, the essence of randomness is that, like lightning, it is not apt to strike twice in all the same places.

While I enjoy and find my computer experiments fruitful, fascinating, and fun (sometimes lovely copy chugging out and I needn't do a thing — after a certain point), I don't predict a great future community among writers, computers, and computer programmers. Computer time is expensive, few writers are yet their own programmers, and programmers may not possess the kind of minds that want to produce creative literature. Literary experimentation can be an uncertain process, requiring the species of poetic, unprosaic mind that is happy with unfixed parameters, serendipitous juxtapositions, no-definite-end-goal-from-the-beginning. Programmers may find such freedom pointless or frightening.

My eight computerized projects have so far involved a couple experiments with learning computer and programming terminology and applying it to characters in a novel I wrote. Next I did two design poems for which the machine was given two lists of words and two basic sentences, then told to combine and recombine these words, creating nearly endless new sentences, finally commanded to print these on the page in certain designs. The two lists of words: one on sex and one on violence. I achieved the designs by examining the total printout, numbering the "best" sentences (funniest, most sensible, most tragic, etc.), and commanding the machine to print these in certain line lengths.

Line 1: print sentence No. 14 complete, followed by beginning of sentence No. 178. Total width allowable: 50 characters including spaces . . .

Line 5 (shorter line): print characters 1 to 20 . . . The whole is similar to a crochet pattern of varying row lengths and stitches.

The final result begins SEX

_____ and _____ are the end of _____.

SEX.

MARRIAGE and PREGNANCY are the end of LOVE. LOVE and MARRIAGE are the end of MAN. MAN and PREGNANCY are the end of WOMAN. WOMAN and HORMONES are the end of UNDER THE SHEETS. UNDER THE SHEETS and
DIAPERS are the end of W
OMEN'S LIBERATION. WO

and so on.

Print on the total page is shaped to form the letters S
E
X.

Print on the violence page forms a gun with bullets spraying.

My next project ("Things I Will Never Do Again") resulted from giving the computer over a dozen lists of words (all from my novel, all classified by part of speech and by "value" or tone within the book's emotional context) plus a basic seven-line stanza form:

+ = POSITIVE NOUN, ADJ, VERB, ADV
- = NEGATIVE NOUN, ADJ, VERB, ADV
° = AMBIVALENT NOUN, ADJ, VERB, ADV

RHYME LINE
A <u>+ADJ</u> <u>+NOUN</u> <u>ADV</u> <u>+VERB</u> <u>PREPOSITION</u> THE <u>+NOUN</u> <u>-ADJ</u> AS <u>-NOUN</u>
A <u>°ADJ</u> <u>°NOUN</u> <u>ADV</u> <u>°VERB</u> <u>PREP</u> THE<u>°NOUN</u>
HOW?
<u>+ADV</u> <u>-ADV</u> <u>°ADV</u>

RHYME LINE
MOUNTAIN KNOWS SNOWS
A FABULOUS FIRESHADOW QUICKLY CIRCUITS ABOUT THE GODDESS
DANK AS RAINSTAINS
A FLIRTATIOUS JOETTE SOMETIMES DIAPERS UPON THE PLANE
HOW?
FLATTEREDLY REDCOLDLY KAY-NINELY SKYFLY

"Things I Will Never Do Again" is actually the last page of the novel, titled *Happenthing in Travel On.* Happenthing is a happening; Travel On is an old house. The book is a winter frontier adventure of a group of women – one with a baby – who take an airplane and live together on a mountainside. More stanzas appear elsewhere in the book, using key words appropriate to the action in each section. Each stanza can also be considered as an interplay of variables (underlined parts of speech, above) with constants (definite and indefinite articles, repetition of certain rhyme lines).

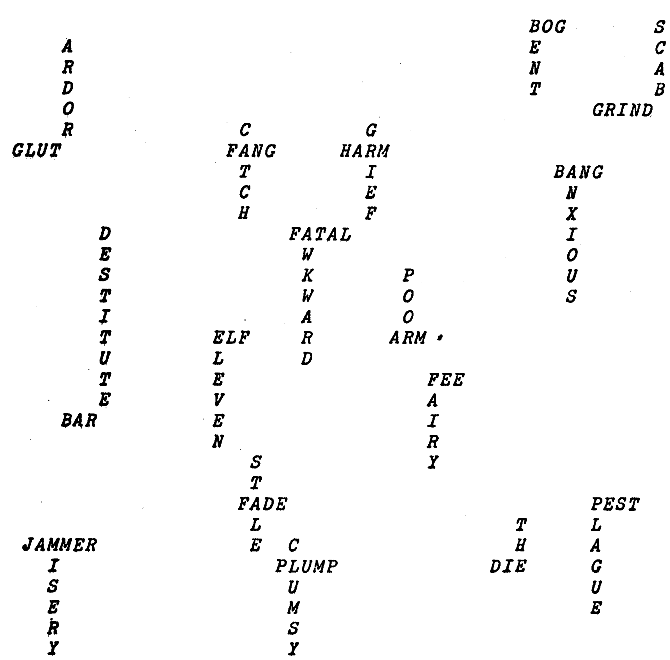
Two more projects, again from the novel, appeared as a separate book, *Six Portraits,* in Germany. These combine design poetry with German-English language learning. The two total about fifty pages of printout. For the first, I assigned to each of the novel's people a symbol plus a paragraph set of basic sentences. Each paragraph is different and appropriate to the personality, speech, and attitudes of the character. Each paragraph appears four times in partial form, the fifth time in complete form. A command (for example, IVY X 5 2 1) begins each computer run on a character. This means that program X will print pattern number 1, 5 times in the first run (first paragraph), double (2) that amount or ten times in the second, and so on until the whole paragraph appears and can be read. Pat. No. 1 has a fixed shape (crosses of horizontal and vertical lines). However, it is again random number generation that determines exactly where, which letters, in each paragraph that pat. No. 1 will choose or use to repeat the design.

Here are samples of a first, third, and fifth (final) run of the paragraph on the character Joette Winton. *Taube* means "dove"; *Schneeballschlacht,* "snowball fight."

```
                                                              HNEEB
                         L                                  .  ADD
     I                  RYT
    UNC                T?  I
    S ARE              BED QUI
   ED.


                                                        N    HNEEB
           E             L                             CAR  .  ADD
     I    RS.           RYT                            VERY
    UNC   ON T         T?  I                         R THE F
  P  S ARE WET.        BED QUI K                                        S
 UNTED.

                           E

                  U         ST GE  L          E    IN    HNEEB   SCH    HT.
          E:    AMEB     L . EMATS.  SO      AYI   CAR  .  ADD  HEER  O
  MIL   I   RS.  WE DO    RYT        RSELV     PT THE VERY         ORK.  D
 ERVE  UNC  ON THIS FLIG T?  I      AN UNWE    HER FOR THE F
DIAPERS ARE WET.  GO T  BED QUI K         C     HE DOCT                S  IS
 UNTED.

   T   N                   E
    D  A:
    PRONOUN    A   U LY:       ST GER L  IM G      E   EIN    HNEEB LLSCHLA HT.
    TRANSLATE:   GAMEBIRD  L  EMATS.  SONGBIRD   AYING CAR   .  ADD CHEER TO
  MILY DINNERS.  WE DO EVERYTHING OURSELVES EXC PT THE VERY   AVY WORK.  DO YOU
 ERVE  UNCH ON THIS FLIG T?  I WAS AN UNWED MOTHER FOR THE F I.  M  SON'S
DIAPERS ARE WET.  GO T  BED QUICKLY--I SHALL C  L THE DOCT  .  THIS    S  IS
HAUNTED.


JOETTE WINTON      HOUSEWIFE, MOTHER      37
     DATA:  DIE TAUBE
     PRONOUNCE CAREFULLY:  ES IST GERÖLL IM GEWÖLBE.  EINE SCHNEEBALLSCHLACHT.
     TRANSLATE:  GAMEBIRD PLACEMATS.  SONGBIRD PLAYING CARDS.  ADD CHEER TO
FAMILY DINNERS.  WE DO EVERYTHING OURSELVES EXCEPT THE VERY HEAVY WORK.  DO YOU
SERVE LUNCH ON THIS FLIGHT?  I WAS AN UNWED MOTHER FOR THE FBI.  MY SON'S
DIAPERS ARE WET.  GO TO BED QUICKLY--I SHALL CALL THE DOCTOR.  THIS HOUSE IS
HAUNTED.
```

My other German project: I entered into the computer twelve lists of words. That means two equivalent lists (1 in German = 1 in English) for each of the book's six characters. For example, *bang* (German) = "anxious" (English). The machine was then programmed to print these words randomly in anagram-like pairs.

Here is a sampling of words that describe the character Giselle, a frightened student.



The trick is that to an English-speaking person, the page appears entirely English. However, one-half of each word pair (all the horizontally printed words) is actually German.

Both the German projects are intended as a new kind of verbal portrait, a different method of doing fictional characters beyond the usual "he said, she said, then they . . ." arranged in consecutive paragraphs and pages of narration and description. Both are also satires on foreign language learning in general. Both were done in APL, a general purpose programming language, and the programs appear with them.

My latest project, done for a book *Computers and Creativity,* is a brief, randomized fairy tale. Its title is "Five Ways to Tell a Story: the Sad Case of Catrina M.". Here are two versions of Catrina's first paragraph.

1.

Once upon a time like twenty years ago Catrina M. was born at an early age in Winnemucca, Nevada. She was a smiley baby, full of milk and cereal. At three she attended nursery school where she played with her teachers constantly. Having somehow survived her young years, she entered first grade, full of trust and hope. Her tearful subjects were arithmetic and underwater basketweaving, (This was a very weird school.)

Being a good girl, she married her mother, adored her father, and stammered at her sister, who was twenty years younger. (This was a very major family.) However, she did manage to mature and get it all together by the age of sixteen.

Next she tried female consciousness raising although her father didn't believe in careers for women. This took her thirteen and a half years. When she had finished, she still wasn't qualified to undertake higher education until she learned never to say no.

She never hated anybody because her mother had doubts about success. Comforted by her friends, dog, and boss, she lives a mostly lively life in the Bronx. Sometimes Catrina M. still has doubts about fair treatment but generally she has managed to trust other people.

2.

Once upon a time like a-lady-never-tells-how-old Catrina M. was born at an early age in Caribou, Maine. She was a bratty baby, full of sound and fury. At three she attended nursery school where she ignored her teachers constantly. Having somehow survived her young years, she entered first grade, full of tantrums and eating problems. Her liberal subjects were arithmetic and underwater basketweaving, (This was a very major school.)

Being a good girl, she fought her mother, cried at her father, and married her sister, who was thirteen and a half years younger. (This was a very progressive family.) However, she did manage to flip out and reach outward by the age of ten.

Next she tried female consciousness raising although her father didn't believe in nervous breakdowns for women. This took her twenty years. When she had finished, she still wasn't qualified to undertake diplomas until she learned to brew coffee.

She never blushed at anybody because her mother had doubts about fulfillment. Comforted by her friends, dog, and boss, she lives a mostly celibate life in Milford, Pennsylvania. Sometimes Catrina M. still has doubts about courage but generally she has managed to flip out.

Future possibilities for myself and others in this area: using the machine to explore metaphor and simile with larger vocabularies and other stanza forms, to investigate other rhetorical problems. Why do some of the machine's random choices work so much better than others? Why are some so striking, others nonsense? Can the machine create paradoxes or epigrams? How to compare the process of human, poetic creativity with these electronic processes (random number generation, letter matching, table lookup of equivalents, sorting, merging, etc.)?

Such computer work excites me, at the very least, as a welcome change from the usual think-type-retype-retype process by which literature has been created. The challenge is to devise concepts encompassing, original, complex, and subtle enough to give the machine, the author, and the programmer an optimal workout.

Poets, Birds, Snow, Kites, and The Computer

by Arthur Layzer

Last December, at the engineering institution where I teach, a novel program of computer art works entitled "The Computer Is a Medium" was presented. In these works, the computer in some essential way "filtered" the expression of the English language: the voice speaking the words of a poem was an artificial, musical voice synthesized by the computer (Speech Songs by Charles Dodge); an animated film was constructed entirely out of the textured words of a poem (the author's "Morning Elevator"); the computer was used to spew out possibilities that filled in the blanks of a pre-set sentence structure (Carole McCauley's Do-It-Yourself poems on Sex and Violence); animated graphics and poem fragments interplayed with the aid of a general programming language (Stan Vanderbeek's and Ken Knowlton's "Poem Fields").

At the center of "A Computer Is a Medium" was a non-computer event, a reading by Siv Cedering Fox. Her poetry evoked personal resonances from inorganic snow, ice and water. A remarkable illusion of spatial extension occurred, enhanced by a computer-music background of James Randall.

At first sight it seems absurd to have invited a live poet to a computer arts program, about as absurd as entering a bird in an exhibition of fancy kites. The situation looks more reasonable if you compare not how well the computer-kite or the bird-poet can fly but how they each take to the wind. The bird-poet flies effortlessly. The computer-kite with its over-simplified angular construction is vulnerable and reacts transparently to the wind blowing. And it is dependent on a human hand for guidance by way of a long string that is just visible enough so that you can't forget it.

What is the wind for the artist who works with the computer? That is the hardest thing and sometimes you have to blow *yourself* to keep the composition afloat. But there is also the wind of random numbers and the wind of a programmed process which you have set up but which you can't control from moment to moment.

It seems that only by getting outside of ourselves can we express anything that is profound. The poet learns this instinctively and is then called crazy.

The inside of the computer is chilling in its starkness, its ordered qualities and its fragmentation. When the creative wind blows on the computer's personality, shapes it or melts it to an organic form that we recognize as humanly associated—takes the computer's personality outside of itself—we feel the significance of the human situation in a striking way: We have managed to get outside of ourselves also—perhaps going the other way.

Reprinted with permission of the author from the *Arts in Society* bulletin (Summer/Fall 1974) of The Univ. of Wisconsin - Extension.

"Computers and Beauty"

by Mutsuko SASAKI and Tateaki SASAKI*
1-18-17, Syoan Suginami-ku, Tokyo 167, JAPAN

In this article we consider beauty and simple methods of its realization by a computer. Here, by the beauty we do not mean such beauty as of logic or of love but mean artistic beauty appealing to the human's aesthetic sense.

In order to realize the beauty systematically by a computer, we need the concept of beauty to be defined suitably for programming. In searching for such a definition, we first note a very simple and very typical example: the "golden section," i.e., the most *beautiful* sectioning of a finite line into two parts. It is impressive that the ratio giving the golden section is not far from 1:1. If the ratio was 1:10, say, then the corresponding section would not appeal to our aesthetic sense. One of other typical examples is music; a beautiful melody would be an ugly one if the arrangement of notes in the music were bad. We can suppose from these examples an *essential* element of beauty; we may call it harmony. A thing being lacking in harmony will make the human's feeling unstable and cause an antipathy in his mind. Saying this from the viewpoint of beauty, we can say that it is not beautiful. On the other hand, we also note that the ratio giving the golden section is not exactly 1:1. We may describe this as that variety is necessary for the object being *more* beautiful. Let us illustrate this by cooking: cooking must contain the nutriment, but even the nutrimental cooking is not a better one if it tastes bad! Thus variety is an *additional* element of beauty. In this sense, an object which is full of variety but lacking in harmony, say a picture drawn by utilizing only random numbers, will not be beautiful. Therefore, one can adopt *the representation of harmony and variety* as a simple definition of beauty. It should be noted that both harmony and variety are conceptually much less abstract than beauty itself and suitable for computer programming because they can be converted into numerical relations rather easily. In the following we shall consider actual methods of realization of beauty through a computer by taking up the drawing as a concrete example.

For the sake of explanatory convenience, we classify the pictorial beauty into i) mathematical beauty, ii) natural beauty, and iii) creative beauty. By these terms we mean, respectively, that i) beauty found in figures easily expressible in terms of mathematical functions, ii) beauty found in nature, and iii) beauty being dependent mainly on the human's creative powers. Many of the actual drawings may, of course, contain two or all of these types of beauty. We are not sure that above classification covers all types of the pictorial beauty, but the reader will see that it is very adequate for the computer drawing.

i) We often experience that simple mathematical functions enable us to make excellently beautiful drawings. Most functions commonly used in applied mathematics represent continuous changes of one dependent variable with respect to changes of other independent variables under some definite rules. These rules maintain the balance of the resulting drawing as a whole, and continuity of the functions prevents the local configurations of the drawing from becoming too various. Thus we may say that most mathematical functions are harmonic. (The functions called harmonic in mathematics are solutions of the Laplace's differential equation. We are using the term "harmonic" in wider meaning here.) Perhaps the mathematical functions are best suited for expressing harmony simply. What we should care for in this case is how to represent variety. Fortunately, variety is also contained in the functions to some extent, and mathematics provides us with random numbers. Hence, we may be able to represent variety sufficiently in terms of mathematics. Many works of the computer art up to the present are based on the mathematical beauty. This beauty is so popular to computer artists and so easy to realize by a computer that we do not discuss it anymore.

ii) Most people will agree with an opinion that nature is full of variety. But we can also find a great many kinds of harmony in nature; e.g., figures of flowers, butterflies, fishes, mountains, etc., etc. We think this is because that laws in physics, biology, etc. govern the nature and make it be harmonic: matters are built from atoms systematically, only evolved and selected beings are surviving and they are balanced as a whole due to the struggle for existence, and even the nonliving things are balanced. We can hence represent harmony as well as variety by imitating the natural objects. Further the natural objects can also cause through our memory many psychological effects which are no more contained in the category of beauty. The important points in representing beauty by natural objects are that the resulting drawing should not become too complicated and that each object drawn in the drawing should be easily identified. The reason is that natural objects themselves are sufficiently various. For example, suppose we are drawing many flowers on a paper. If we only outline the flowers the drawing would not be so beautiful. If the overlapped and invisible parts are also drawn, the resulting drawing will be too complicated and may even be ugly. In this case, therefore, more complicated programs or peripheral systems are necessary than in the previous case. Some examples are an image reader and a picture recording system, but here we explain our simple program by showing illustrations. Figure 1 shows an input drawing for our system. The drawing showing three flowers in Fig. 2 is rather ugly because it is too complicated; even to identify it as representing three flowers is not so easy. Invisible parts are eliminated in Fig. 3. In Fig. 4 insides of the flowers are "painted" with assigned patterns. The resulting drawing is much more beautiful than that shown in Fig. 2. In this way, we can see that we can well represent beauty by imitating natural objects and processing them through even a simple system. We should of course design many systems in order to raise the computer art to higher positions. But, since nature is full of variety, we can conceive many possibilities. In particular, since we know beautifulness of the mathematical beauty and easiness of its handling, we have a good possibility in creating a new type of beauty which has not been well considered so far, by synthesizing the mathematical and the natural beauties.

iii) It is very common that artists raise aesthetic effects of their pictures by emphasizing the principal parts, changing color tones, modifying and distorting the figures to be drawn, and so on. Further artists often make pictures by using mainly their inspiration and imagination. The beauty represented in these cases is created by the human's creative powers. Therefore, in order to treat it systematically and automatically we

*Present affiliation: The Institute of Physical and Chemical Research, Wako-shi, Saitama 351, JAPAN.

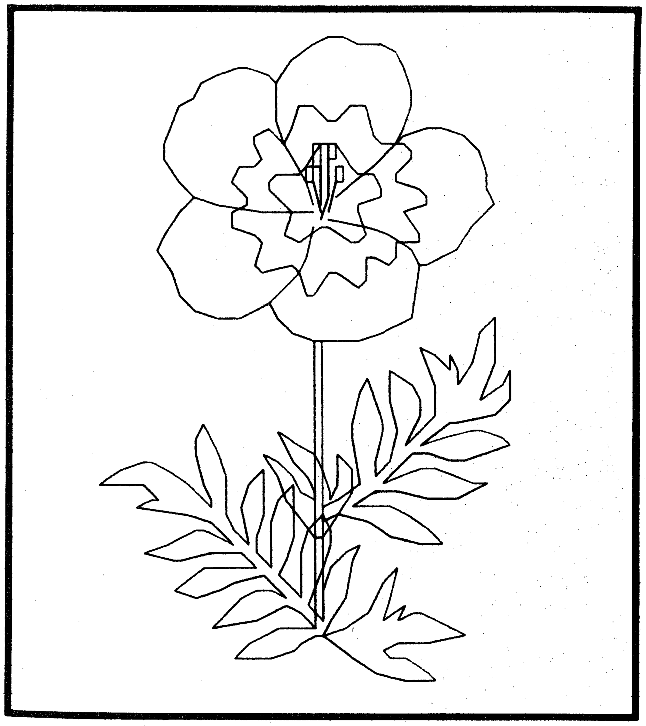


Fig. 1

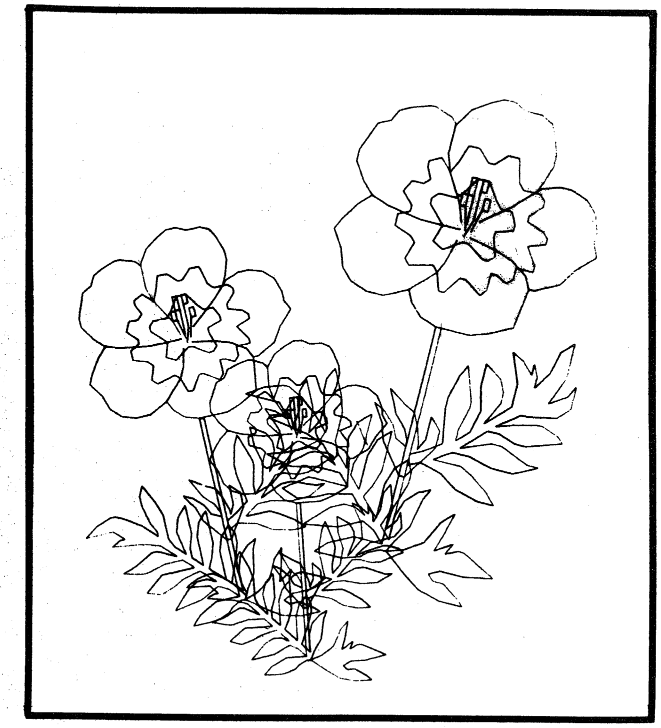


Fig. 2

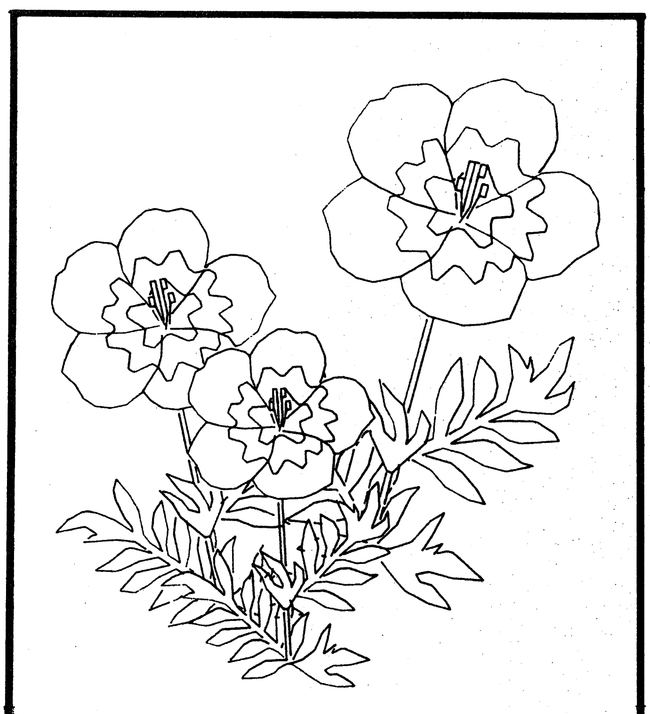


Fig. 3

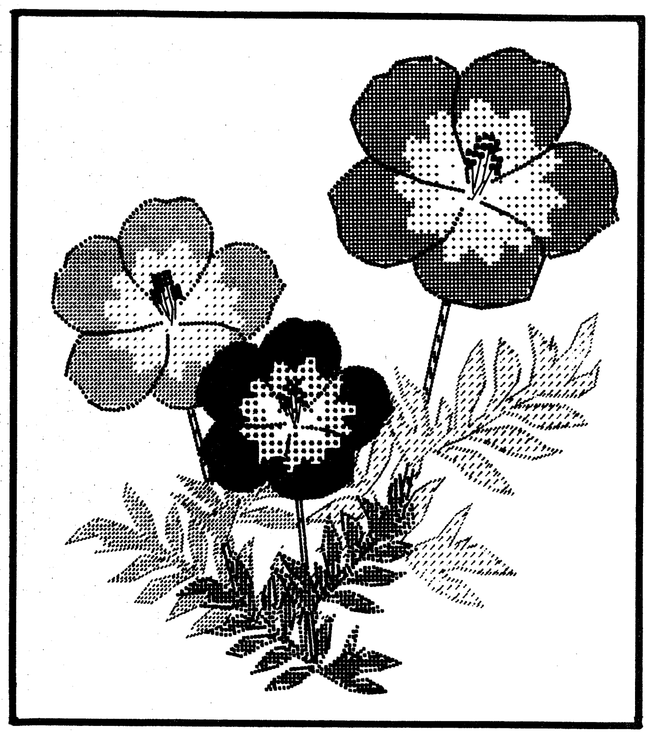


Fig. 4

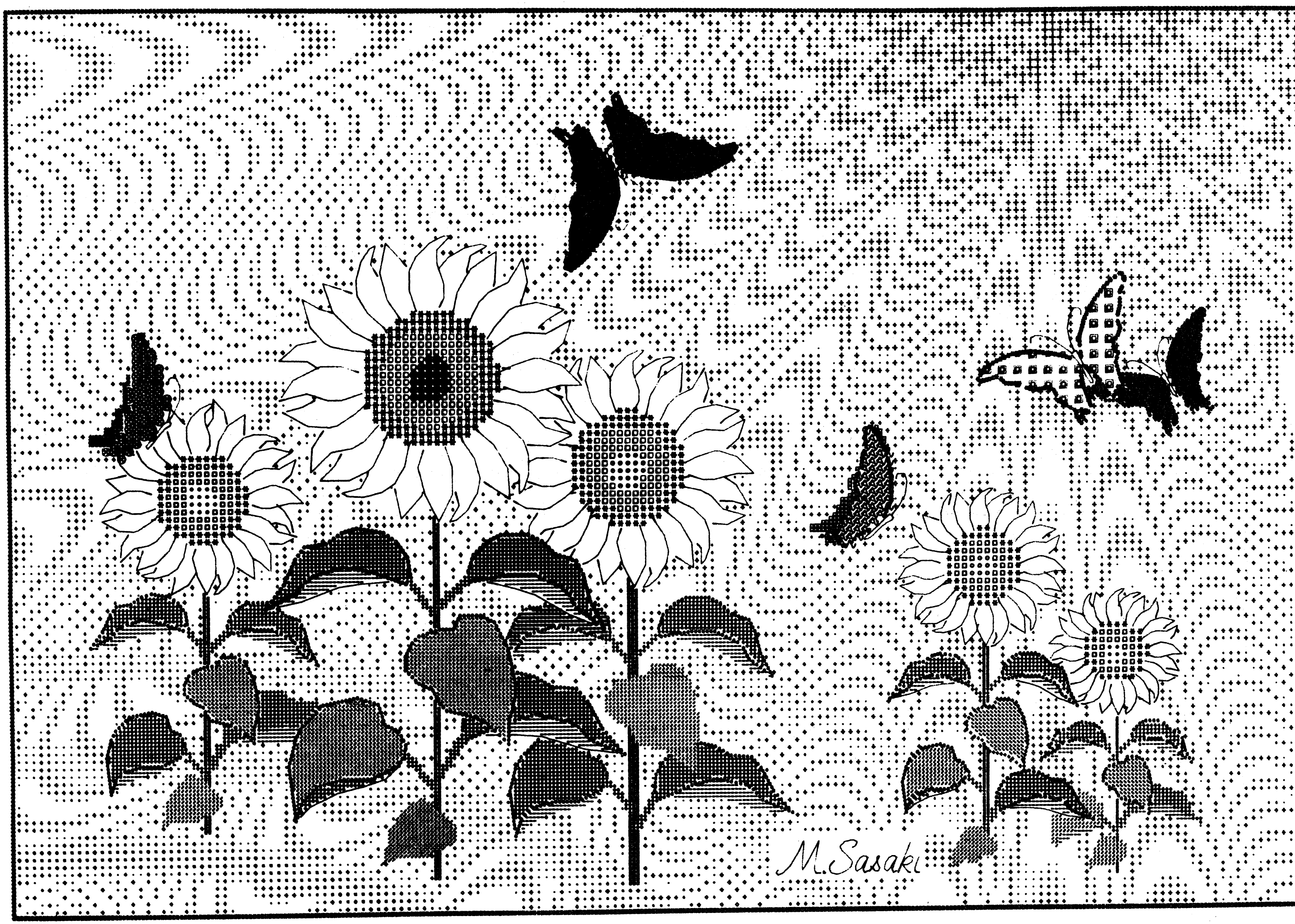
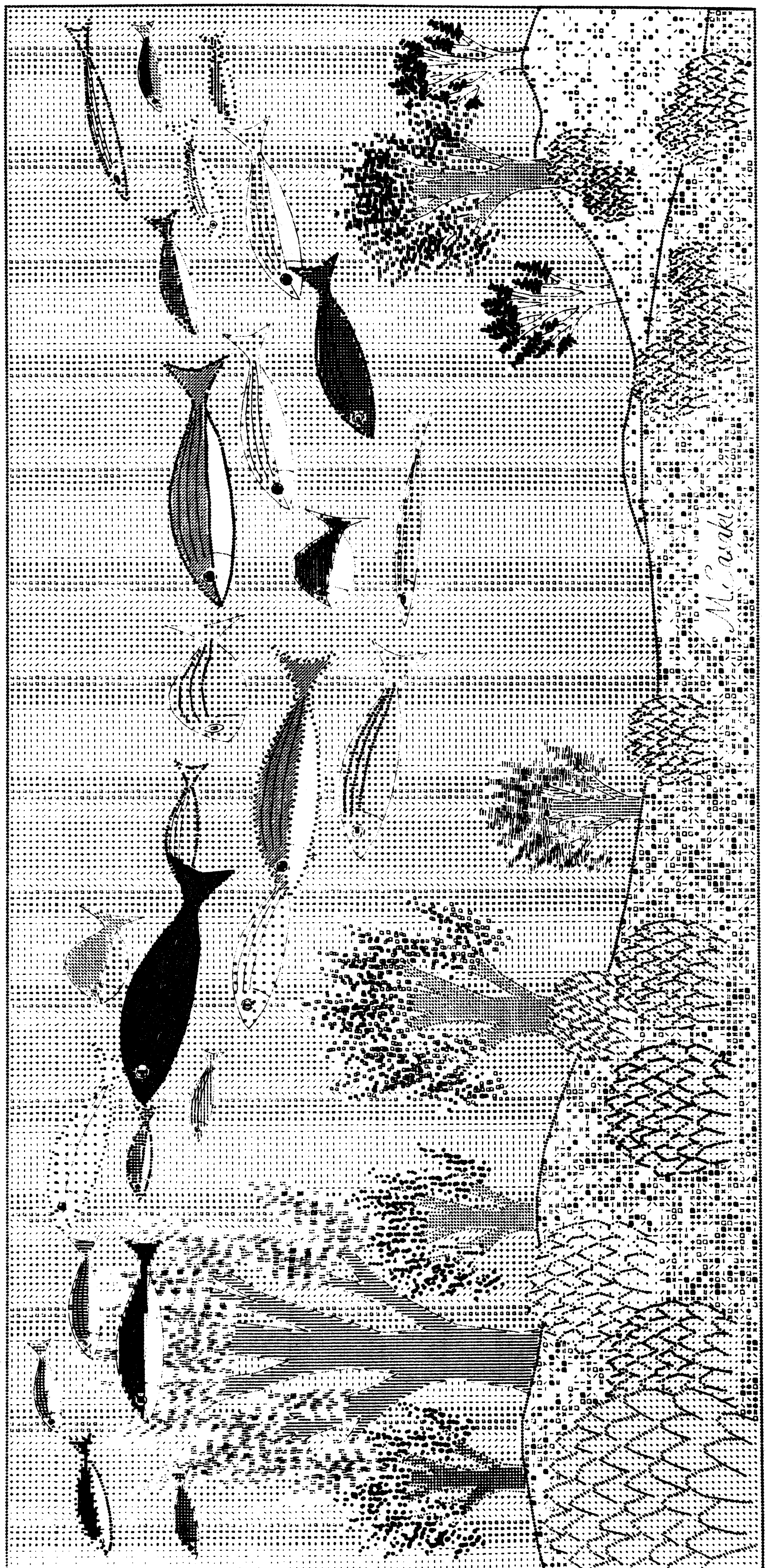


Fig. 5: "Hellow Sunflowers!, II." Original size 24 x 34 cm^2. Computer execution time 48 sec on a FACOM 230-75.



Perhaps we will have to educate and train the computer so that it can appreciate beauty.

must implement, more or less, the creative powers to computers. This is a very important task, because the creative powers are so much widely required in almost all of the art works; even in handling mathematical functions we encounter many problems whose solutions require the creative powers, such as how to use the functions in the drawing. On the other hand, the implementation is certainly a difficult task. We must investigate many things: What procedures are effective in creating beauty? How can we realize them by a computer? And so on. Perhaps we will have to educate and train the computer so that it can appreciate beauty, by making trials and errors. An important strategy which we would like to propose here is to ease the difficulty by adopting our definition of beauty and considering all things from this viewpoint. We have seen that the definition is persuasive and useful for the previous two types of beauty. We think it is also useful enough for the creative beauty. For example, the artist's techniques mentioned at the beginning of this section can be explained in part as that artists recover harmony from otherwise excess of variety. Of course there are many ways in representing harmony and variety, and much more detailed investigations are necessary in the actual programming. But above example seems to show usefulness of our strategy clearly. We can not say much about the realization of creative beauty through a computer at present, because we have not considered it yet. A few years later we will surely be able to say much about it.

Finally we show our recent works in Figs. 5, 6, and 7. They were produced by our system for computer art named ART-3, whose main functions are i) transformation of the input figures, ii) elimination of the invisible parts, and iii) to paint the surfaces by various patterns. ART-3 contains more than fifty mathematical functions for the figure transformation and the pattern generation. We can see from these figures how ART-3 works and how mathematical and natural beauties are united by it.

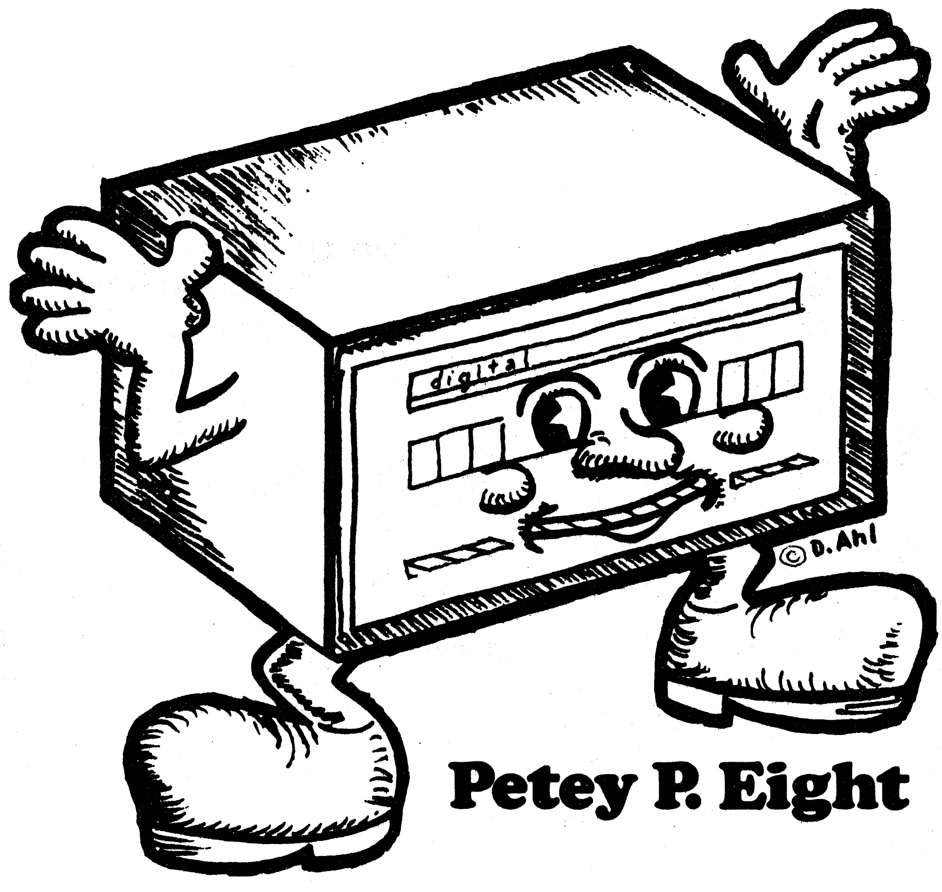
←

Fig. 7: "Surmarine walk, II." Original size 24 x 50 cm^2. Computer execution time 83 sec on a FACOM 230-75.

Figure 6 appears on the next page.

Figure 6: "A City" by M. Sasaki and T. Sasaki. Original size is 23x19.5 cm_2. The computer used was a Falcom Z30-75.

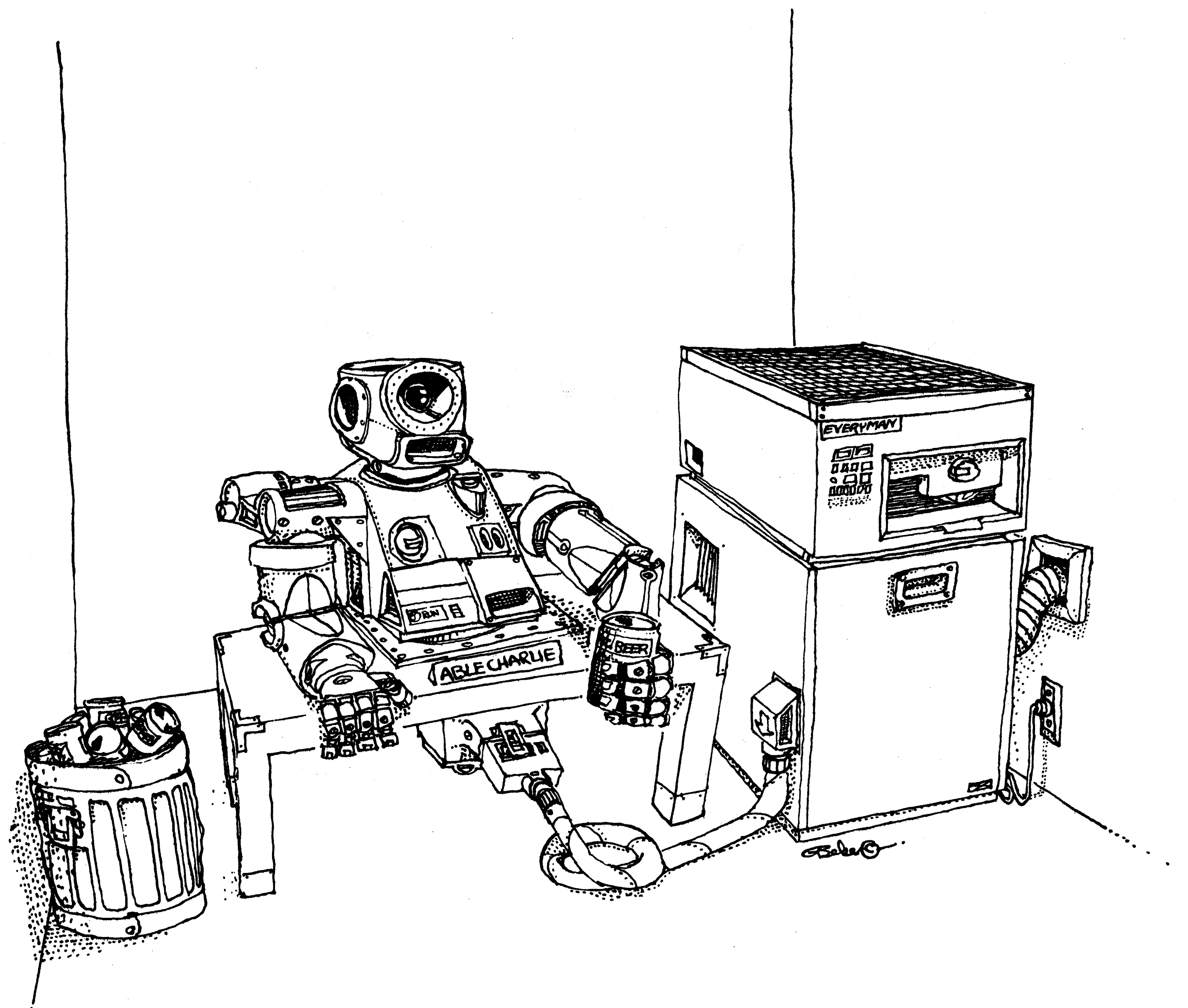
Fiction and Poetry



Petey P. Eight

A Day in the Life of Able Charlie

by Frederik Pohl



The time was 0900:00 A.M., and Charlie woke up.

The first thing he had to do was to find out who he was that day, and so he explored his memory. He discovered that he was a white male American, thirty-two years old, married, employed in the sales department of a public utility company. He had two children, a boy and a girl. He had made $17,400 in the year just past, and if it hadn't been for Harriet's part-time teaching salary he didn't know how they would have managed. He still owed over $19,000 on their $38,000 house, $1900 on the car and nearly a thousand on the loan for modernizing the kitchen they had taken out two years before. Moreover, his daughter, Florence, had unfortunately inherited his bite, and so the orthodontist was going to cost him fifteen hundred dollars very soon. Charlie discovered that many of his thoughts were of money.

However, his memory contained many other things. He became aware that he was a fan of the Los Angeles Dodgers, and that he had volunteered as a Little League coach against the day when his four-year-old son, Chuck, was old enough to play. Charlie remembered that he was inclined to favor Chuck over the girl. It was curious that he could not remember what color Chuck's hair was, or whether Florence was doing well in school, but Charlie didn't realize that it was curious and so he continued to explore his memory.

He was a heavy smoker, drank a can of beer now and then, especially in hot weather, but didn't go much for the hard stuff. Although he liked looking at other women, he did not go beyond looking. Although he enjoyed a game of poker twice a month, he did not care to gamble heavy stakes. He drove a small foreign car (it was not clear whether it was a Datsun, a VW or a Fiat), on which he got 24.7 miles to the gallon in everyday use and nearly 29 miles a gallon the road. (He did not know what color it was. It did not occur to him to wonder why.) Charlie remembered that he was active in his party's politics (he did not know whether it was Democrat or Republican) and that he thought the mayor of his town was a crook. But he could not have said the mayor's name.

All these things about himself Charlie apprehended in a very short time indeed. He then spent somewhat longer remembering what brand of cigarettes he smoked, where he bought them, what had happened when he tried to give them up (his wife complained of his short temper and begged him to start again) and what other brands he had tried. He rehearsed the services offered by his neighborhood filling station, and what he looked for when he needed gas on the road; what kind of Scotch impressed him when he was offered it at a friend's home; and why he had decided against switching from lather to an electric razor. Charlie inventoried every purchase he and his family had made for the past year, swiftly and without error. He recalled what TV programs he watched, what magazines he read and which of the thousands of commercials and advertisements they contained had affected any of the purchases.

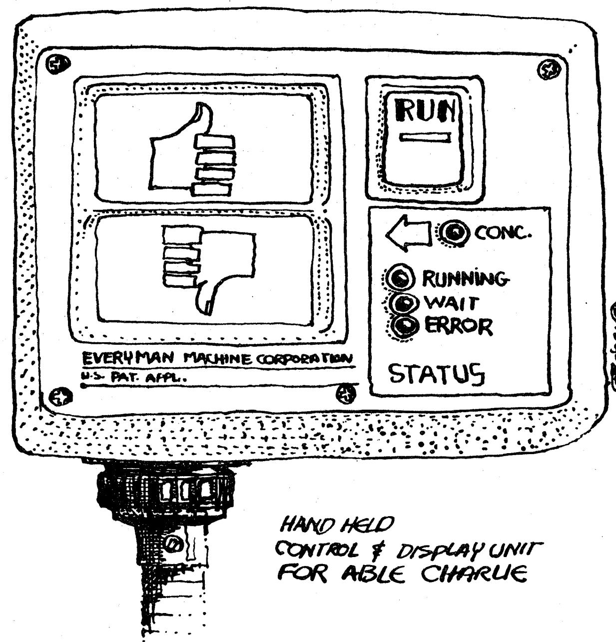
At that point Charlie discovered that he had done everything he was required to do just then. He made a quick parity check on his instructions. When it revealed no gross error or failure on his part, he announced that he was ready for his next task and waited in standby mode for orders.

He waited what was for Charlie a very, very long time. All of this had taken Charlie a period measured only in fractions of a second. Now he rested, neither wondering nor moving, for a stimulus to further action. Without it he would do nothing, ever. He was not impatient. He knew what "patience" was in conceptual terms — he could relate it to his memory of himself waiting without "patience" for a traffic light to change — but it did not occur to him to feel that way now.

At 0901:30, give or take a few seconds, a young woman in a light gray dress, carrying a container of coffee, set the coffee down on her desk and seated herself before a large typewriter. She had heard the bell that announced Charlie was ready more than a minute before, but she was not quite ready for Charlie. She typed several rows of characters, checked them over, took a sip of her coffee and stood up.

She glanced at the various lights and dials on Charlie's front panel, saw nothing to cause concern. Her typewriter had produced not only the visible row of characters on the sheet of paper it held but, on a spool connected electrically to the keys, a strip of magnetic tape. She snipped a four-foot length of it free, taped it to another reel, rewound it and fed it into a scanning device. She removed the rubber band from a packet of perforated cards and dropped them into a hopper.

Then she pressed a button. Rubber-tipped fingers dealt the cards into sorting bins where, one by one, they were taken up again and read, like the music roll of an old player piano. The tape reel slid past its scanning head on a cushion of air and disappeared. The time was 0901:55.



Charlie began work — not at 0901:55, exactly, but at a time so near to it that the difference was measurable only in picoseconds.

His first problem, he was informed, had to do with cigarette package designs. He waited while the cards on that subject were scanned. There were forty-one alternate designs, and they were presented to him in pairs. First he was offered Package One and Package Two simultaneously; he compared them, made a value judgment based on what he knew of his own buying habits and preferences and stated his preference. Then Package One and Package Three were offered to him, then Package One and Package Four, and so on until Package One had been compared with each of the others. Then he was offered Package Two with Package Three, Package Two with Package Four; and on and on until each prospective design on the list had been compared with each other. (There were 861 combinations in all, taken two at a time.)

At that point Charlie went into a sort of reverie while another part of his mind — it could have been called his "subconscious" — tabulated the results of his cross-pairing and established an order of preference. He wrote down, in order, the ten package designs he had most favored. He wrote it in the form of impulses recorded on a magnetic tape (this caused a reel by the desk of the girl in gray to spin rapidly for a moment, which she noticed out of the corner of her eye). Then he hummed for a moment, waiting for the card reader to allow him to begin his next task.

Each of Charlie's value decisions had taken him only about four nanoseconds, but the evaluation and read-out were much slower. It took him considerably longer to announce his results than to arrive at them, and so it was 0902:45 before he began his next job.

The next assignment was to assess the merits of some proposed shaving-cream formulations.

Here the task was considerably more difficult, for several reasons. The first part of his task was to rank his preferences among the fifty-five formulations as to their odors, textures and visual appearances, each in combination with the other. Charlie did not, in fact, realize quite how difficult it was, since he had no idea that he possessed neither smell nor vision, and touch only in the sense that certain of his members were capable of probing a card or tape for punched holes. He then had to evaluate some 24 shapes and weights of pressure canisters in relation to each sort of lather. Here too, Charlie was unaware of his lacks. In fact he did not have thumb and fingers; the "grasp" and "weight" and "feel" of the canisters in his "hand" was in fact only a locating of certain binary statistics within the parameters of certain other quantities that were a part of his memory. In order for Charlie to be able to express an opinion on any of the matters on which his verdict was sought many subterfuges had been devised by the programmers on the staff of the advertising agency that owned Charlie. They materially prolonged the time for each comparison. However, he was in no way concerned by this. He did what he had always done. He did the task that was assigned to him, and when it was done he looked for, and did, the task that was next.

In all of the hour and forty-odd minutes in which Charlie, husband of Harriet, father of Florence and Chuck, searched his responses to a wide range of offerings, he performed something over five thousand million separate operations, including parity checks and internal verifications. He faithfully reflected the customs and tastes of the average of a sample of some four million American males as they pertained to the purchase of tobacco, beer, gasoline, automotive accessories, soft drinks, airline tickets, motion-picture admissions, sporting goods, hi-fi equipment, toilet articles and power tools. When his final magnetic report was on the tape he signaled by ringing a bell. That was the end of Charlie's working day. In a sense it was the end of his life.

The girl in the light-gray dress was in the assistant division chief's office when Charlie's bell rang, and so she didn't react at once. Charlie waited like a man on a benzedrine high, his mind clear and capable, but disengaged. It was nearly 1100 when the girl got back to her desk.

She took the spool of tape that held all his opinions and threaded it into a printer, where it began typing out plain copy at a rate of 350 words a minute. She replaced it with a blank spool, consulted her work sheet and began to change Charlie with switch, with patch-cord and with dial.

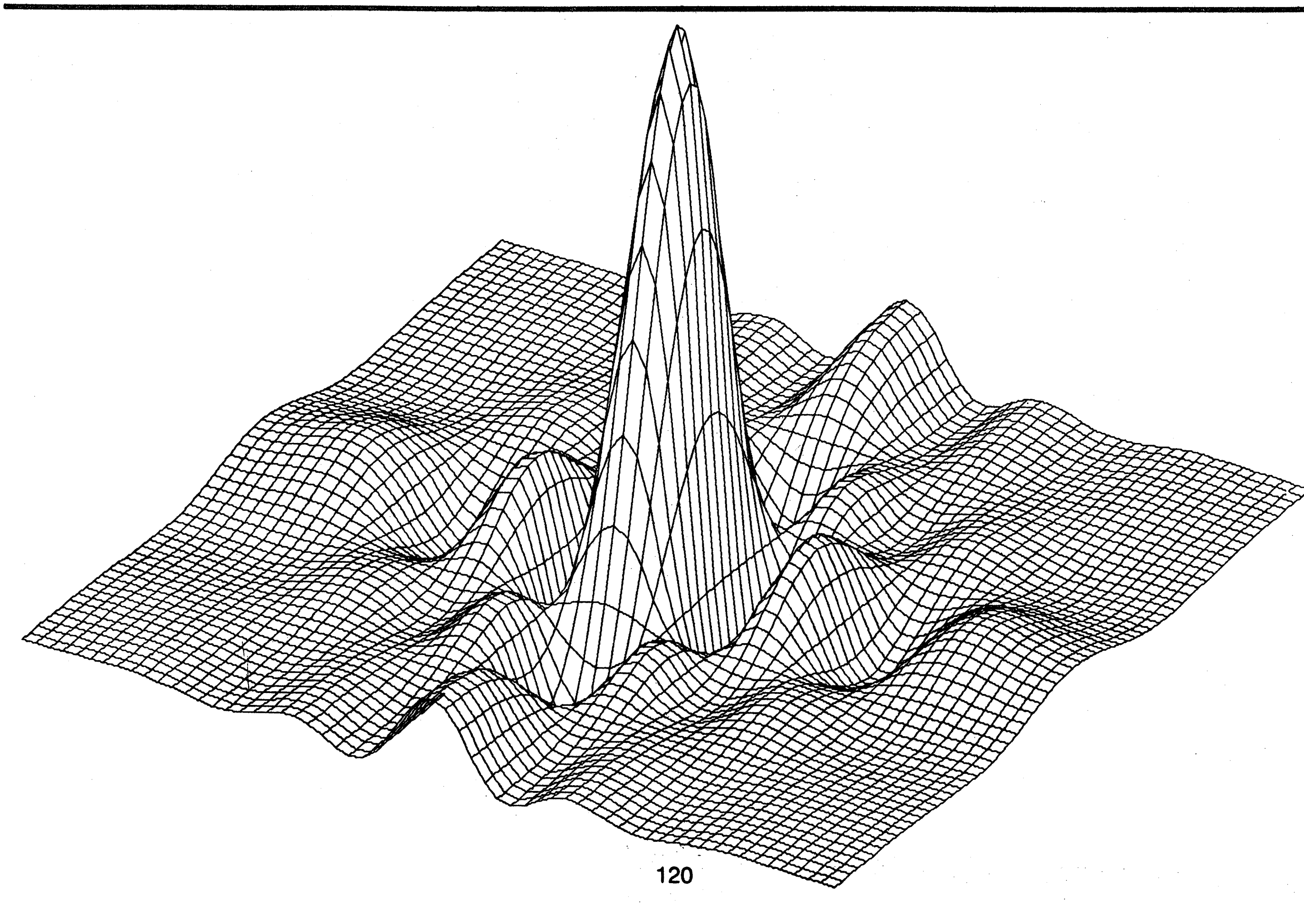
As she worked whole banks of memories dropped out of circuit. Chuck and Florence fell out of his personality without leaving a mark. His wife disappeared, his house, his car; the Los Angeles Dodgers went, with the Little League and the dunning letters from the bank.

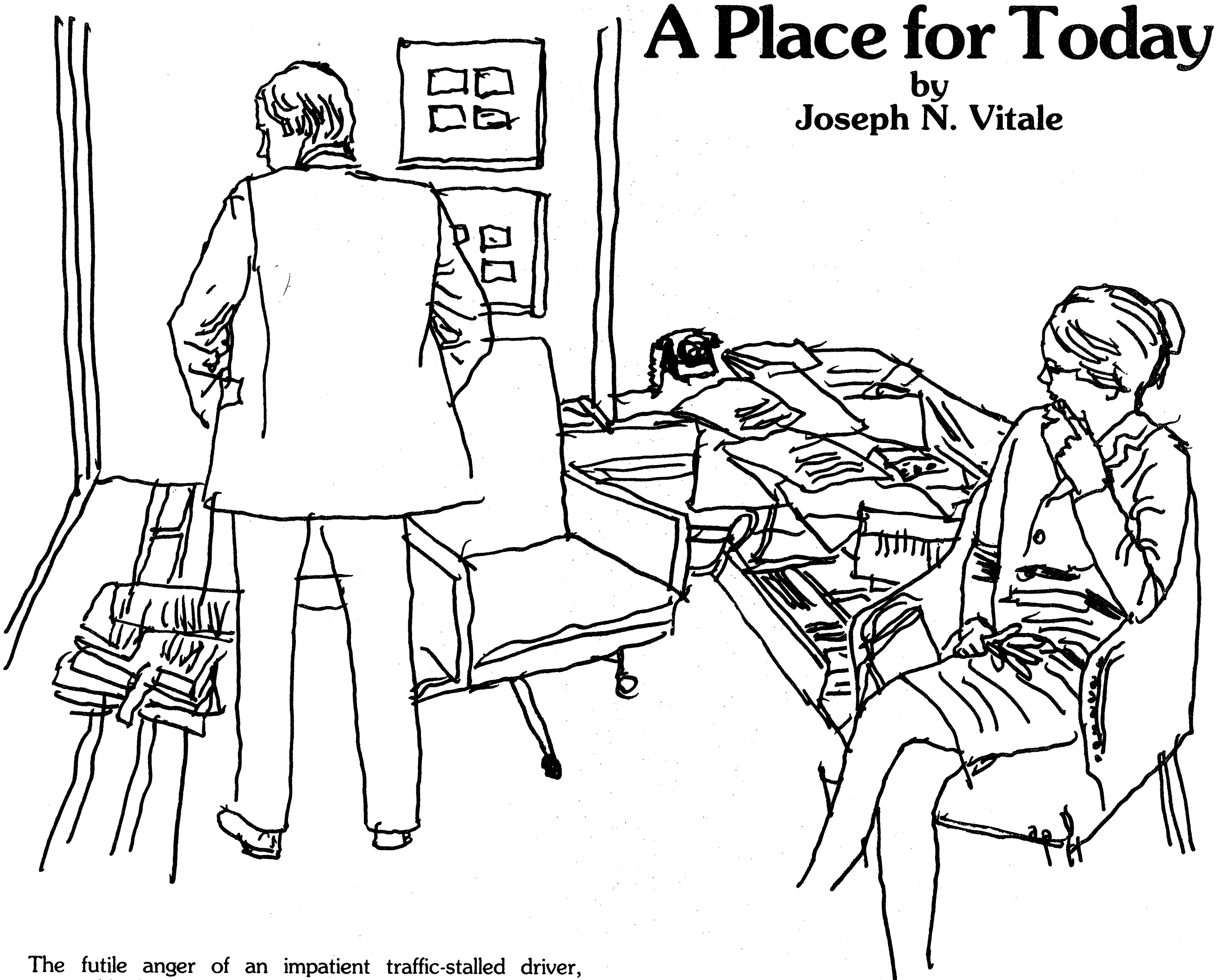
She then checked the programming sheet and, following its instructions, selected new personality ingredients for Charlie: an economic level, an age, a set of buying habits, a profile of interests. She began to charge Able Charlie with the sum of these habits and biases. He was not yet aware of what he was, since he had not yet received the command to learn himself. For that matter, he was no longer "he". Now Able Charlie was a teen-age girl, her principal interests cosmetics, soft drinks, clothes, records and boys.

When all the patches were complete and the new tapes were ready to roll, the girl in the gray dress double-checked, and pressed the "execute" button. Able Charlie, AC-770, began to take up his — her — its new life.

The girl in the gray dress idly examined the polish on her nails. Her mind was not far from stand-by mode, either; until the first read-out came, or a trouble signal, she had nothing to do but wait for lunch.

Inside the AC-77 Charlie, or Charlotte, was swiftly sniffing colognes whose fragrance was only the simulation of magnetic patterns on iron-oxide tape and comparing shades of lipstick whose colors were only a point on a hypothetical scale. The girl programmer was comparing colors, too. She wished idly that she had a friend to chat with — Rose Pink, after all? Or Catalina Coral? — but when she thought she heard a low contralto sigh she dismissed it at once. She knew that she was alone.





A Place for Today

by

Joseph N. Vitale

The futile anger of an impatient traffic-stalled driver, expressed by long blasts of a car horn, dissipated as it filtered through his office window. It became the only melodious sound he had heard all day. Up until then he sat, mind and body bathing in dreariness, staring at the rain-soaked window and listening to an almost-cyclic splatter. The wind seemed to drive periodic herds of rain into that side of the building. Blow, you bastard, he thought, it's music to my ears! As if commanded, the car horn responded even more irately.

Christ, it's dark outside! What time was it? As he turned toward the clock on the wall opposite the window, his eyes scanned a variety of computer-generated art forms: Miss September Playmate of last year with heavy typescript in the right spots; a yellowing Santa Claus sleigh led by only six reindeer above which were the letters *Merry Xmas and a Happy*, the rest had been torn off some time ago; several neatly-drawn trigonometric curves among others. It was only after he noted the time at 3:15 PM that he let out a groan from the pain that originated in his neck. Too young to get a stiff neck on a rainy day.

He looked down at his desk; papers full of mathematical equations, large sheets covered with logical flow diagrams, assorted printouts and decks of punched cards. What did it mean? He studied his latest project; a speedier technique by which the computer could approximate the numerical solution to a set of coupled differential equations. What the hell did it all mean?

The rain seemed to fade and in his mind the quietness of the empty office amplified until the silence was olive-oil pure. Then he was no longer alone! They came again! Large faceless bodyless heads with long hair, medium-length beards covering high-necked collars, through his office doorway, across his desk and out his window, each with a name lettered over an empty face: Leibnitz, Gauss, LaPlace — on they came — LeGendre, Fibonacci, great mathematicians all! And again, also, he realized that they had sat in such offices, and on such dismal days as this, most likely, established much of the theory he was using. And he? He had contributed nothing today, a void which was made even more monumental by the fact that he had five million dollars worth of computer at his disposal. The rains came again and he despaired. I am nothing!

His depression settled on a buzzing sound deep within him. He decided he was hearing himself think. Christ, he thought, not only can't I get my mind together, but I've got to listen to myself producing nothing. But wait! It was outside of himself. The intercom on his desk was buzzing away! He picked it up and his secretary announced the arrival of his 3:30 PM appointment with Miss Ann Coyle.

The young woman sat across the desk from him, her legs crossed, dress raised over the upper knee to an inquisitive length; her long sandy hair was drawn tightly into a bun high on the back of her head and large thick-rimmed circular glasses covered deep-blue eyes which betrayed an 'I dare you' look.

Not another one! Not today! Not after Gauss, LeGendre and company! He was constantly beseiged by a variety of people who wanted demonstrations of the computer's incredible feats:

tic-tac-toe, horoscopes and various other games ingeniously programmed by people like himself to give the computer a humanistic appearance. What bullshit! Inwardly, he was satisfyingly-proud of the fact that he knew of one binary digit — called a 'bit' in the computer — which, if changed in the computer's central processor from a zero to a one, would drive the machine stark raving mad; a berserk multimillion dollar totality of illogic, destroying itself ad infinitum or until someone discovered and pulled the plug.

"I know nothing about computers," she said, her voice humble but firm. So what else is new, he thought. "My primary interest is the philosophy of art," she continued. The what of art . . . the philosophy of art! What the hell?

". . . and I had been directed to you." she finished an explanation, much more confident now. Her skirt rose slightly higher on the knee.

"How do you think I might help you?" he heard himself asking. "Computers have been used to generate artistic pictures using special plotting devices. There are situations, at some museums, where computers have been used to do selective retrieval from files of art objects. For example, given complete computerized collections of all paintings in a particular museum, certain retrieval programs can be used which might list out a total description of those paintings which satisfy chosen sets of properties presented to those programs." He looked up at the crack in the ceiling, almost hoping that water might seep through. He continued, "I suppose there are needs, by various scholars, also, to do statistical analyses by computer on data that physically describe paintings in certain groups . . . like what is the average weight of all paintings in a particular group, or mean canvas areas, or something like that." He stopped to let her absorb his statement.

"But," he went on, in a rapidly tiring voice, "I'm not sure I can see how a computer might be used when dealing with the . . . the philosophy of art." He raised his eyebrows and peered heavily through her glasses. Am I sinking in, he wondered? No, I am not, he answered his own question.

"Please correct me if I say anything that sounds like non-sense," she said. "I have heard that the computer can do . . . uh . . . unbelievable things and I wonder . . ." her voice oozed lazily into silence.

"You wonder what? Let's have it!" Christ, let's have it!

"Well, I wonder if the computer can get two people to talk to each other and discuss their philosophies!" she rambled out the phrases in a quick jerky rhythm.

"Oh, is that all? You don't need a computer for that! What you need is called a telephone!" He was getting somewhat impatient. This girl is either very misguided or very strange.

"No, let me finish." She smiled. "The people I am talking about lived at different times! One is Marcel DuChamp, the French-American artist who died in 1968. The other is Marshall McLuhan, the American art philosopher who is still alive." She spoke much more confidently, now. "You see, it would be intriguing to hear them discuss their philosophies with each other. It would be particularly interesting if the computer could generate a dialogue between them in the form of a two-man play. While . . . although their lives overlapped . . . it's the one thing they never did . . . talk to each other. And yet, it would have been one of the most stimulating discussions in the history of art philosophy." The skirt was definitely up higher than ever now and the eyes twinkled, almost immorally.

He gulped hard. He rose from his desk, walked over to the window and looked outside. It had stopped raining now and it was much lighter. A rainbow was forming to the West. The traffic was moving a lot freer now, too. They went out this window . . . Leibnitz, Fibonacci and the rest. Where did they go? And what did it mean?

He turned and stared at the back of her head, concentrating on the bun as if it were some crystal ball in which he might see some meaning. Was there any special association between the events of today — the rain, the parade of the faceless ones, this girl with her philosophy of art, DuChamp and McLuhan, the computer and the rainbow? Was there any special association?

Association? Association! The phrase came slowly, but it came, first rolling soundlessly off his tongue, then circling his mind like a stock market ticker-tape. Linear Associative Information Retrieval! He had read of it some months ago in one of the journals; a new theoretical technique which could be used to determine the sameness or differentness of two documents . . . to indicate the snynonymy or contiguity of two documents by using similarity matrices with respect to assigned keywords which described the documents. Weren't the ideas of sameness or differentness when applied to concepts rather than documents a way to distinguish philosophies — basically the writings of the men! Yes, yes! His mind accelerated now, building the basic steps, already seeing mental images of flowcharts, computer programs and, finally, dialogue printing at a computer terminal! Dialogue between DuChamp and McLuhan!

And wait! Why stop at DuChamp and McLuhan? Certainly, if the computer could be made to generate discussion between them, then the possibilities were endless. What about Adolph Hitler and Karl Marx? Or William Shakespeare and William Faulkner? Or any two people who had published a great deal? Or any three people? Unbelievable! But possible! Round table discussions between any of the eminent people in the history of the world!

A large bodyless head came through the doorway of his office, across his desk and out the window. Only this time it was not faceless!

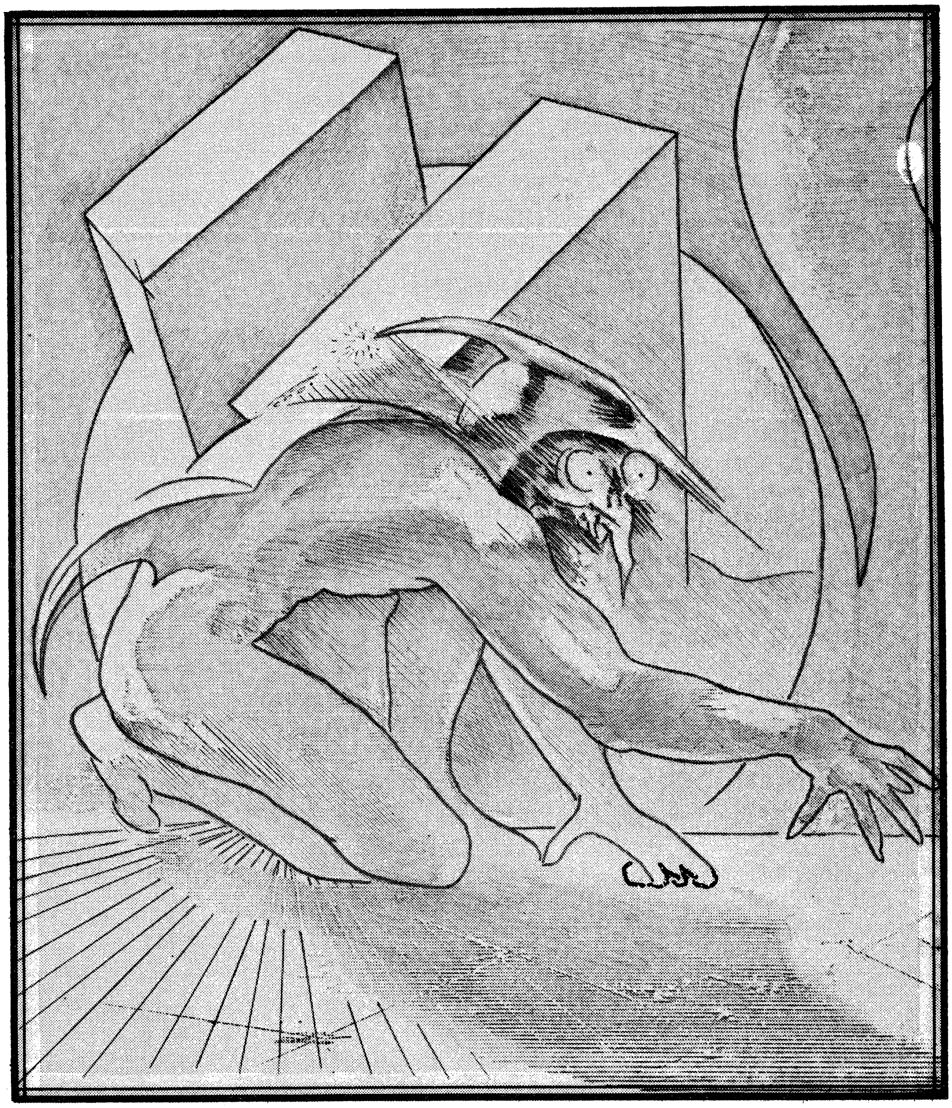
"Miss Coyle," he said, "could you bring me some literature which contains a representative sample of the philosophies of DuChamp and McLuhan? I think . . . I think I might be able to help you."

". . . No thanks, . . . Just scanning . . ."

What really happens when a major computer in a modern information network breaks down?

Charles Mosmann

PULLING THE PLUG



"All this business about computers taking over is nonsense, of course, but it's dangerous nonsense. It's just not true. Newspapers trying to scare up a headline."

The noise level in the coffee shop had suddenly abated, as though everyone paused to take a breath at once. I couldn't help overhearing the remarks being made at the table behind me.

"Right," a second voice affirmed, "You can always pull the plug."

I did not turn around to see who the two men were; they were undoubtedly a couple of the hundreds, perhaps even thousands, of engineers and programmers who seemed to be involved in virtually every enterprise in the large office building where I worked.

When I got back to my office, I kept thinking about the idea of pulling the plug. Did anybody really know where the plug was? In fact, modern information networks are designed so that if one computer stops working, the work load is automatically shifted around to some other machine. The computer that accepts a user's job is likely to be no more than a scheduling and switching center, which will locate a computer somewhere else to do the actual processing.

I had a few minutes before I was going to have to leave for a meeting. I decided to pursue the question a little further to pass the time. I have a typewriter-like terminal beside my desk which is my own personal access to the computer network. I turned my chair to face it and flicked on the switch. The computer system to which I subscribe announced itself and asked me for identification. I typed in my name, my account number, and a password known only to the system and me, which functions as a sort of combination lock so that no one else can mess with my files or run up my bill, pretending to be me.

Finally, all these formalities were taken care of satisfactorily and the machine typed out, "READY." The following dialogue took place between us.

READY

list available socio-economic models.

ECO-75, CAL81, USECON, ECOMETRIC, USECON2. MORE?

no. call CAL81.

(pause)

YOU HAVE CALLED CAL81, A SIMULATION OF LOCAL AND NATIONAL SOCIO-ECONOMIC SYSTEMS. DO YOU NEED A DESCRIPTION OF CURRENT FILES AND CAPABILITIES?

no.

PLEASE STATE PRESUPPOSITIONS OF YOUR SIMULATION.

time - current. status - normal except: pull the plug on this computer.

DEFINE "THIS COMPUTER." DEFINE "PULL THE PLUG."

this computer - the computer on which CAL81 is now running and which is in communication with user 15a-225/85. pull the plug - this computer stops operating and does not restart.

OK. PLEASE STATE OPERATION/OUTPUT REQUIREMENTS.

significant changes to systems/this city/next 7 days.

(pause)

DAY 1, 14:07. MASTER COMPUTER 307/21 AT COMPUTIL INC LOSANGELES CEASES OPERATION. NO REASON REPORTED. NO ESTIMATE OF DOWNTIME AVAILABLE. ALL UNCOMPLETED JOBS ROUTED TO COMPUTIL SANDIEGO UNTIL FURTHER NOTICE.

DAY 1, 16:30. SUDDEN INCREASE IN WORKLOAD AT BANK CLOSING TIME OVERLOADS SANDIEGO COMPUTER AND EXCESS ROUTED VIA LOSANGELES TO SANFRANCISCO.

DAY 1, 16:37. UNANTICIPATED POWER SURGE IN SANFRANCISCO TEMPORARILY INCAPACITATES COMMUNICATIONS SATELLITE COMPUTER. WORK IS SENT BACK TO LOSANGELES WHICH ATTEMPTS TO REROUTE IT TO DENVER.

DAY 1, 16:38. SYSTEM DEADLOCK IN DENVER COMPUTER DUE TO SUDDEN OVERLOAD ON SCHEDULING ROUTINE. NO OLD JOBS ARE COMPLETED. ALL NEW JOBS ENTERING SYSTEM ARE ROUTED TO LOSANGELES OR SANDIEGO.

DAY 1, 16:39. LOSANGELES COMMUNICATIONS CENTER OVERLOADED AND SOME MESSAGES ARE LOST.

DAY 1, 17:00. CITY BANKS CANNOT BALANCE OUT ACCOUNTS. AT FOUR BANKS THERE ARE NO PROVISIONS FOR CLOSING THE BANK UNDER THESE CONDITIONS. PERSONNEL STANDING BY AWAITING INSTRUCTIONS.

DAY 1, 17:00. THREE LARGE MANUFACTURING PLANTS IN LOSANGELES AND TWO IN SANDIEGO USING COMPUTER-CONTROLLED TIME CLOCKS. THESE ARE RUNNING SLOW AND WILL NOT ALLOW WORKERS TO GO HOME.

DAY 1, 21:15. MANUAL PROCEDURES FINALLY SUCCEED IN CLOSING DOWN BANKS AND PLANTS FOR THE NIGHT.

DAY 2, 09:00. 23 PAYROLLS DUE THIS MORNING ARE NOT AVAILABLE DUE TO CRISIS OF DAY BEFORE. BANKS RECEIVE NUMEROUS REQUESTS FOR SHORT-TERM CREDIT WHICH HAVE TO BE DENIED BECAUSE CREDIT FILE IS NOT AVAILABLE, DUE TO FAILURE TO CLOSE OUT BOOKS PROPERLY DURING CRISIS OF DAY BEFORE.

DAY 2, 11:40. OVERLOAD ON SANDIEGO CENTER CAUSES DECISION BY ELECTRIC UTILITY TO SHIFT OPERATION OF GENERATOR SCHEDULING PROGRAM TO PHOENIX COMPUTER. MOMENTARY DIP IN POWER LEVEL DURING THIS SHIFT.

DAY 2, 11:41. DIP IN POWER LEVEL DAMAGES DISC STORAGE AND DISC DRIVE OF SANDIEGO MASTER COMPUTER. TAPE RENDERED UNREADABLE ON INPUT/OUTPUT COMPUTER.

DAY 2, 11:42. AUTOMATIC SCHEDULING PROGRAM AT SANDIEGO COMPUTER CENTER DECIDES TO LOWER WORKLOAD BY TRANSFERRING IT TO SEATTLE VIA TELEPHONE. DELAY IN CARRYING OUT PLAN DUE TO FAILURE OF LONG-DISTANCE BILLING SYSTEM; ALL LONG-DISTANCE CALLS MUST BE PLACED MANUALLY.

DAY 2, 12:00. DAMAGED DISC IN SANDIEGO CONTAINED UNIQUE COPY OF YESTERDAY'S POLICE FILES. 17 PRISONERS RELEASED BECAUSE THERE ARE NO RECORDS, NO GROUNDS FOR DETAINING THEM.

DAY 2, 14:00. DAMAGED TAPE HAD CREDIT RECORDS OF UNITED TRUST AND LOAN COMPANY. ALL LOAN OPERATIONS TEMPORARILY SUSPENDED; ALL ESCROW OPERATIONS DELAYED AT LEAST ONE WEEK.

DAY 3, 09:00. 26 RETAIL STORES CLOSE TODAY FOR INDEFINITE PERIOD, CONSEQUENCE OF SEVERE CASH/CREDIT SHORTAGE.

DAY 4, 09:00. ALL FUEL DELIVERIES BY NATIONAL STANDARD OIL COMPANY TO THEIR DISTRIBUTORS ARE HALTED. THEIR SIMULATION AND TRANSPORTATION SCHEDULING SYSTEM WAS BUMPED FROM COMPUTER BY PRIORITY JOB FROM CITY MANAGER'S OFFICE.

DAY 4, 15:00. EVENING EDITION OF DAILY TIMES DEMANDS FULL INVESTIGATION OF CRISIS! TO BEGIN IMMEDIATELY. ALSO ANNOUNCES THAT PUBLICATION WILL BE HALTED IMMEDIATELY AS OF THIS DATE, SINCE AUTOMATIC TYPESETTING SYSTEM CANNOT BE RUN. BLAME PLACED ON SEVERE SHORTAGE OF COMPUTING CAPABILITY IN THE CITY.

DAY 5, 09:10. UNITED TRUCKING, OPERATIONS TRANSPORTATION, AND AGRI-TRUCK INC FORCED TO CURTAIL OPERATIONS DUE TO FUEL SHORTAGE AND CASH/CREDIT CRISIS.

DAY 5, 10:15. REPRESENTATIVES OF LOCAL MARKET CHAINS HOLD PRESS CONFERENCE, INDICATING MAJOR FOOD SHORTAGES. SOME EVIDENCE OF HOARDING ALSO REPORTED.

DAY 5, 12:10. CITIZENS MASS AT CITY HALL, DEMANDING ACTION. COMPLAINTS INCLUDE: LOSS OF JOBS, SCARCITY OF FOOD, ALARMING DIP IN QUALITY OF TV PROGRAMMING DUE TO COMPUTER SHORTAGE.

DAY 5, 16:00. UNDERGROUND NEWSPAPER NOW ONLY PAPER IN CITY STILL IN PUBLICATION, DUE TO PRIMITIVE MANUAL NATURE OF ITS OPERATION. THIS WEEK'S EDITION PLACES TOTAL BLAME FOR CRISIS ON "GIANT BRAINS" AND CALLS FOR DESTRUCTION OF COMPUTERS.

DAY 6, 07:00. POLICE REPORT THAT NUMEROUS COMPUTERS AND OTHER AUTOMATIC MACHINES HAVE BEEN DAMAGED DURING THE NIGHT. PUBLIC ELECTRIC COMPANY HAS TO SHUT DOWN THREE GENERATORS; CALLS ON NEIGHBORING DISTRICTS TO HELP OVERCOME EMERGENCY.

DAY 6, 07:30. DRAIN ON NEIGHBORING DISTRICTS CAUSES INTERMITTENT POWER FAILURES THROUGHOUT THE STATE.

DAY 6, 09:00. POWER FAILURE CAUSES ALL COMPUTERS IN STATE CAPITOL TO STOP OPERATING, EXCEPT FOR MILITARY COMPUTERS WHICH RUN ON EMERGENCY GENERATORS.

DAY 7, 09:30. GOVERNOR REPORTS ALL SYSTEMS IN STATE ARE NOW INOPERATIVE EXCEPT FOR NATIONAL DEFENSE.

END OF 7-DAY REPORT. CONTINUE?

no. status report, national level, as of day 10.

PRESIDENT DEMANDS TOTAL ISOLATION OF STATE TO AVOID DOMINO EFFECT AND SYSTEMATIC DESTRUCTION OF NATIONAL LIFE-STYLE.

successful?

NO. TELEPHONE, TELEGRAPH, RAIL, COMMERCIAL TRUCK, AIR TRANSPORTATION ARE CLOSED DOWN. MICRO-WAVE INTER-COMPUTER NET HAS FAILSAFE PROVISION WHICH CAUSES IT TO REFUSE TO ACCEPT TRAFFIC BECAUSE OF DANGER OF OVERLOAD. COMPUTERS IN NEIGHBORING STATES BEGIN TO SHUNT POWER/INFORMATION SERVICES TO DAMAGED AREAS.

effect?

NEIGHBORING STATES EXPERIENCE SIMILAR BREAKDOWNS. PRESIDENT DECLARES NATIONAL EMERGENCY AFTER BRIEF CABINET MEETING.

effect of this declaration?

MINIMAL. NO MEANS TO PROMULGATE MESSAGE.

status of population?

DEATHS FROM STARVATION AND CIVIL INSURRECTION. DETAILED STATISTICS?

no.

CONTINUE SIMULATION OR TERMINATE?

terminate.

SIMULATION TERMINATED. SAVE FILE OR SCRAP FILE?

scrap.

FILE SAVED. SIGN OFF.

My curiosity had been satisfied by this brief game. I checked my watch, saw that it was time to leave for my meeting, and turned away from my terminal. As I was putting papers in my briefcase, I heard the rapid clicking noise of a further message being typed. I did not bother to check what it was; not infrequently, systems information of only marginal importance is printed out for users who may be interested.

As I was putting on my coat and turning to the door, a funny thing happened. The ceiling lights in the room were beginning to flicker. I instinctively turned my head up to look at them. A few of the flourescent panels were already dark. I don't know what made me look down at the terminal just then, but I did. And in the fading light I saw the message that had been printed a few minutes before:

MASTER COMPUTER 307/21 AT COMPUTIL INC LOS-ANGELES CEASES OPERATION. NO REASON REPORTED. NO ESTIMATE OF DOWNTIME AVAILABLE. ALL UNCOMPLETED JOBS ROUTED TO COMPUTIL SANDIEGO UNTIL FURTHER NOTICE. END OF MESSAGE.

Before I took my eyes from the page, my office was totally dark.

POEMS

by Peter Payack

THE COSMIC CLOCK

Life crawled
out of the sea,
three hundred million
years ago.

On the cosmic clock
that's only a few
hours.

The cosmic clock
runs on pure energy,
with stars for
jewels.
It's a one hundred
thousand million
billion jewel clock,
which hasn't been
rewound since
the big bang,
15 billion years ago.

It wasn't
made in Switzerland,
either.

The old man
down the street
wears it on
a gold chain.

It shines
in the dark
like cat's
eyes.

INTUITION

A flip-flop is female,
Built of two NAND gates
Whose output is fed back
To each other as input.

When the input to both
NAND gates is positive,
The flip-flop is in
The indeterminate state.

This does not mean
It cannot decide whether
To go the one way or the other,
But only that you
Cannot predict ahead of time
Which way it will go.

Electrons are tricky
In flip-flops,
Like they were in the lead
Which kept the Alchemists
Guessing.

Flip-flops are built of
NAND gates, and NAND gates
Are powered by electrons,
Which can fool you, just like
They fooled Paracelsus.

PERIOD

According
to Einstein's
theory
of relativity,
any given
point
in space
and time
can be declared
the center
of things.

The period
at the end
of this poem
is the center
of the universe,
now.

THE WHITE LINE

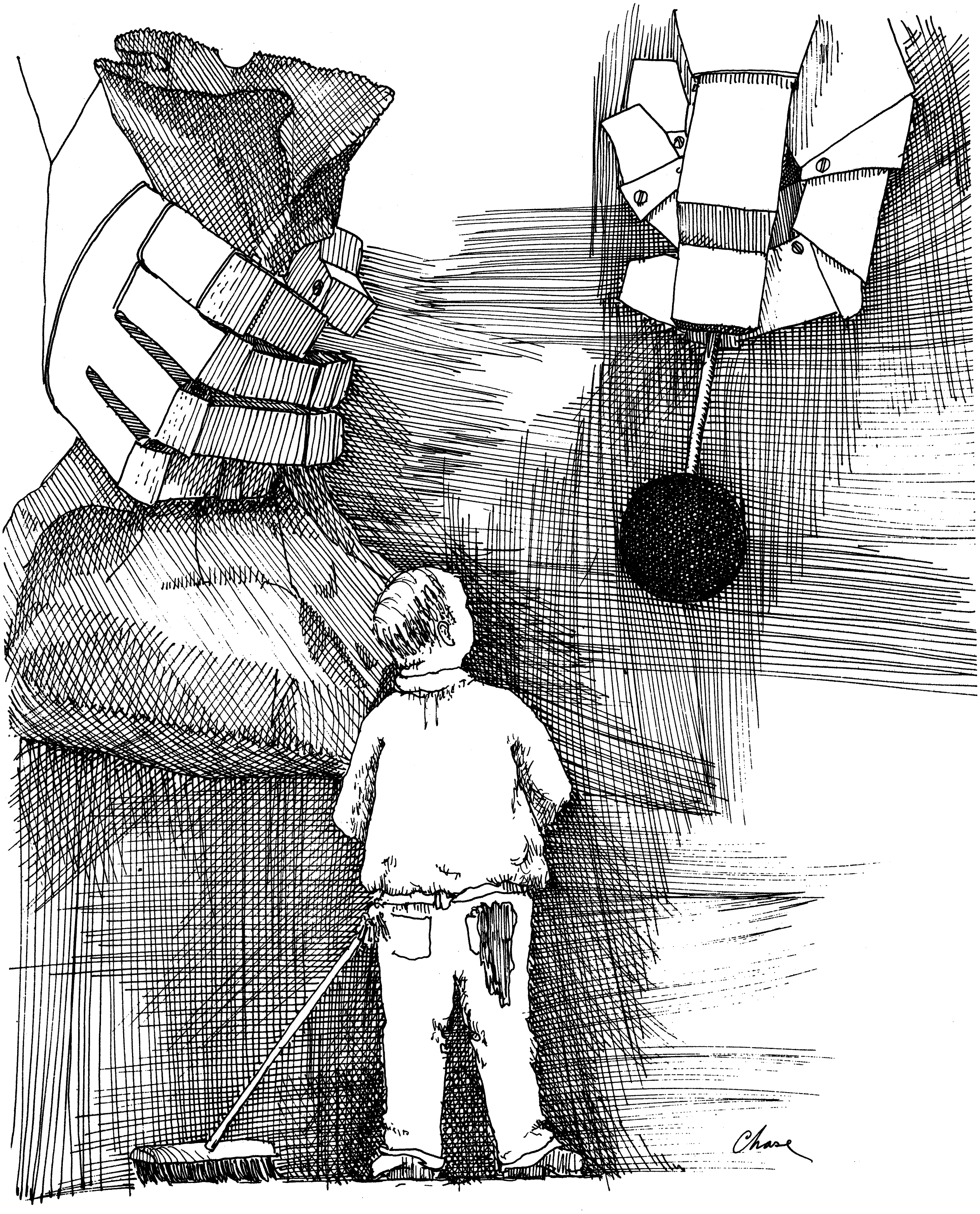
A white line
divides the road

like the border
between sanity and madness.

I drive on the right side
and chuckle at those who come
in the opposite direction:
"Insane bastards!"

NEVER TALK TO COMPUTERS THAT ARE STRANGE

by Carol Cail



"Artificial intelligence. It's a new science between psychology and computer programming. Oversimplified, it's teaching computers to talk—"

"When I was doing Professor Willis just now, he kept sneezing. So I asks, 'Have you got hay fever?' and he says, 'No, it's an old war wound.' Did you ever hear of a wound that would make you sneeze years after? Was it his nose or what?"

Dr. Sills turned a page of the report he was trying to read and murmured, "It was probably the dust you were raising, Malvin. He was kidding you."

"Oh, sure, I see. I should of got it." Malvin ran a gray rag along the blackboard tray and snapped it behind his back, chuckling belatedly.

Dr. Sills sighed and shielded his eyes with long-fingered hands, the report open between his elbows on the desk. He read two lines.

"My dad was wounded in World War I." Malvin leaned on the desk flat palmed, his face low enough to look under the doctor's laced fingers. "Maybe that's why he's such a grouch. His hip hurts."

Sills nodded, staring at the page.

"He walks like Frankenstein. Course, he doesn't hold his arms out stiff in front."

"Malvin, I'm sorry, but I really must study now. Could you talk to me some other time?"

"Oh, sure. I'll get the cleaning done faster if I keep my yap shut, too." He grabbed up a push broom and began plowing the tiny office with it.

The silence wasn't helping Sills any. It was like waiting for a shoe to drop, wondering how long Malvin could keep his thoughts to himself. Ronald Sills had been at the university six weeks and had managed to be working elsewhere in the building at least half of the evenings Malvin was rearranging the dust in his closet of an office. Yet he already knew the custodian's opinions on everything from apples to zoos and back through the alphabet again. Malvin Denwald would rather talk than drink, which is the only activity Sills could think of that Malvin couldn't do and talk at the same time.

Though he'd read the same sentence three times, Sills kept his eyes down in possum absorption; looking up was a certain invitation to conversation.

Of course, Malvin was lonely. One didn't have to be a psychologist, which Dr. Sills was, to know that. Malvin's deficits included a face like a ventriloquist's dummy, all lower lip and protruding eyes. He was approaching middle age, with no chance of a better job and no family but an unaffectionate semi-invalid father. The shortage of intelligence that was in large part responsible for all this also helped him to accept it with good humor. His one real pleasure, then, was having the ear of fine, important men like Dr. Sills.

Who made the mistake of sneezing. Malvin laughed and, the dam of silence broken, words began to pour. Sills shut the report folder and ripped a tissue from the box to blow his nose.

"Which war did you get your wound, hah?" Malvin not only liked talking, he preferred dialogues. "No kidding, Doc. Were you in the army?"

"Korea."

"Oh, come on. Now I know you're pulling my leg. You're too young to been in that one. More likely Viet Nam. Right?"

"I'm older than I look. It really was Korea."

"Well, I be. You sure do look young. Maybe it's cause you're slight built. Kept your boyish figure, hah? Then you got lots of blonde hair and a good tan and all. You ought to grow a beard or a mustache. That'd make you look older. Everybody else around here is hairy. Some of the ladies, even! Ha, ha, ha."

Sills began building an armload of books for exodus to the library. Malvin began polishing the desk top with his dusty rag.

"You through reading your folder here? 'Human Reaction and Interaction in AI Experimentation.' Did I call all those big words right? It sure doesn't sound like very exciting reading. What does the AI stand for, anyway?"

"Artificial intelligence. It's a new science between psychology and computer programming. Oversimplified, it's teaching computers to talk—" Here the rote recitation broke off, and Dr. Sills almost added, "Eureka!"

Instead, he dropped the books onto the desk, fished a key ring from his pants pocket, and, clasping Malvin by the elbow to steer him into the hall, announced, "I have something I want to show you."

Sills unlocked the door directly across from his office, revealing another small room stuffed with equipment: a typewriter-like console, a file cabinet with a telephone on top, a bookshelf loaded with manuals and papers, a fat swivel chair.

"Have you been in here, Malvin?"

"Sometimes to sweep the floor and empty the wastecan. I never touch anything else, so I can't get blamed for anything."

"Yes. This is an expensive terminal setup. This Teletype connects with the computer over in the Science Center. Just this console you see here costs us $3,000 a year rental from the manufacturer."

Malvin made the expected noises of wonderment. "What do you do with it that makes it worth all that much?"

"It's an experimental tool. I'm an artificial intelligence researcher—the only one here so far—and I have a grant to teach language to the computer."

Malvin struggled to see the desirability of that. "You mean English?"

"Well, yes, but it could be programmed for any language. Which one isn't the point. Communication is the point."

Malvin nodded, not seeing.

"Do you know what psychotherapy is, Malvin?"

He hadn't been the sanitary engineer in the psychology building for fourteen years for nothing. "It's sort of like a psychiatrist."

"That's right. The patient talks about his problems to a counselor trained to listen and be sympathetic. Just having someone listening to him is a tremendous help to the troubled person."

"Yeah, that's the truth. I even talk to myself sometimes, and I listen to me, too. Ha, ha, ha. It beats a goldfish."

Sills did not follow up the last remark. He'd sat down at the console, flipped a switch, and dialed some numbers on the phone. The awakened terminal hummed. It teletyped, "NAME?"

Sills typed, "Malvin Denwald."

"Hey, now," Malvin said, taking a step backwards.

"I'm going to let you talk to Art. How old are you?" Sills asked as the computer typed the question.

"Who's Art?" Malvin watched Sills poke a four and a zero. "I'm thirty-nine, Who's Art?"

"Art is short for artificial intelligence, which is what the computer has."

"You mean you want me to talk to a machine?" Malvin shook his hands and head sideways. "I'm not mechanical minded. I don't even want an electric vacuum cleaner."

"You can read and write, can't you?" Sills asked, typing "male" in answer to the last computer question.

"Course, but—"

"That's all you have to do. Read the question or remark the computer makes on this printout, and type whatever you want to say to the computer on these keys."

Sills stood and encouraged Malvin into the chair with a firm arm around his back. They watched the computer type, "DO YOU HAVE A PROBLEM WITH DRUGS?" Malvin gave the console an outraged frown.

"You mean you want me to talk to a machine?" Malvin shook his hands and head sideways. "I'm not mechanical minded. I don't even want an electric vacuum cleaner."

"Type 'n-o,'" Sills advised.

Malvin touched the two keys with reluctance.

"DO YOU HAVE A PROBLEM OF A SEXUAL NATURE?"

This time Malvin's ears lit red and he tried to escape, but Sills held his shoulder with one hand while leaning to type "no" with the other.

"DO YOU HAVE A PROBLEM WITH ONE OR BOTH OF YOUR PARENTS?"

Malvin stopped resisting and studied the keyboard. He typed back, one-fingered, "mother dead dad yes."

The computer relayed, "TELL ME MORE ABOUT YOUR FATHER."

A grin grew on Malvin's face. "Why, Doc, you're doing a deluxe job learning this Art to hold a conversation. Next thing you'll be having him watch TV with you."

"You go ahead and use the computer. I'll be in my office."

Malvin nodded, studying the keyboard. "What if I can't spell something?"

"Just do the best you can putting it down like it sounds. Art can usually figure out your meaning."

"This is sure some machine," Malvin marveled. He jumped as the teletype began a message.

"ARE YOU HAVING TROUBLE GETTING STARTED?"

"You're an impatient cuss. Just give me time," Malvin muttered.

Sills left him unburdening himself via hunt and peck.

Sills settled back with his feet on the desk and began a careful, uninterrupted reading of the AI report. He was unbothered by the constant chatter of the teletype across the hall. It had a rain on the roof tempo, a slow steady dripping that was Malvin, interspersed by bursts of hail, Art.

The precipitation was still in progress thirty minutes later when Sills closed the folder, disengaged his ankles from the desk, and crossed the hall. Malvin glanced up without recognition, concentrating on a response to Art.

"I hate to interrupt, Malvin, but we'd better call it an evening."

Malvin wakened from the spell.

"Gosh, yes. I gotta finish this floor tonight. Thanks a million for letting me try this thing."

"Would you like to do it again?"

"Could I? Sure."

"If you hurry with your work, you'd have time to use the computer the two nights a week you're cleaning my office. I'll put you down as a subject helping me in my experimental work."

"Now wait. I don't want to get you in trouble."

"No, your talking with Art will legitimately help me see what kinds of programming he needs. I have several students coming in during the day to do the same thing."

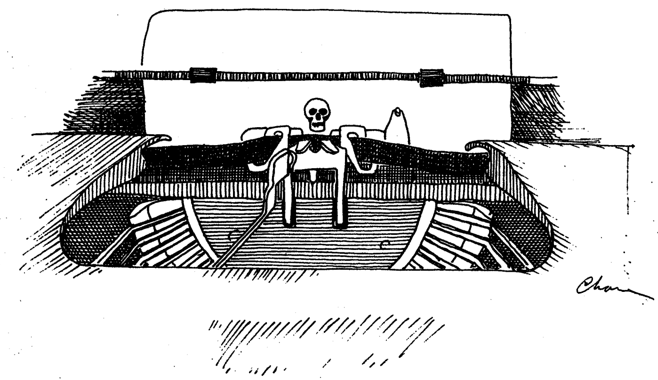
"Wait'll I tell my dad I'm helping a professor with some research, just like a regular student."

"One thing; be sure to save the printouts so I can prove to the backers that I'm earning the grant money."

The smile died on Malvin's wide mouth. "Oh. Well, I better not say anything too personal had I? You really have to do that?"

"You say anything you want, Malvin. It's all confidential. No one will read the printouts."

"Except you."



"I would never reveal anything you want kept private. Besides, you know how busy I always am. I may not have time to read *all* the data you and Art discuss."

"Yeah, well, I guess it's okay," Malvin said with such reluctance that, the minute he was around the hall corner, Dr. Sills extracted the printout and, with more than professional curiosity, began to scan it.

The entire time had been spent on the subject of Denwald, Senior. The image of an ailing, cantankerous octogenarian emerged. The computer's comments were easy to separate from Malvin's, not only because they were shorter, but because they were capitalized:

GOOD. WE ARE LISTENING TO YOU TELL ABOUT YOUR ANGER.

we argue all the time. i dont want to fight with him but it always ends up in a fight. sometimes i feel

PLEASE GO ON.

i never told this to anybody before. i think i hate my dad. i know its wicked but i cant help it.

YOU FEEL VERY GUILTY ABOUT THIS.

i never told anybody. sometimes i actshally want to hit him. kill him even. its awful. i dream about it.

WHAT DOES YOUR DREAMING ABOUT THIS MEAN TO YOU?

howd you like to dream you was killing somebody?

WE WERE DISCUSSING YOU, NOT ME.

Sills shook his head. If Malvin only knew how sadly common his problem was, he needn't be embarrassed about Sills reading the printout.

Malvin's second chance at the computer, on Thursday, yielded a rehash of the Monday session's dialogue. Sills was folding the printout for filing when a number caught his eye, and he examined an exchange he'd skipped previously:

hes a miser. hes got 60,000 dollars in the bank but we live like we was one step from the poor house. we live in a great big manshun thats falling down around our ears. we eat beans. he sleeps in a attic room on the third floor because he says heat rises. he wont even let me spend my own money. id get a guitar.

WHY DO YOU WANT A GUITAR?

(Here Sills made a mental note that Art needed programming on how to keep to the point. But Malvin's next remarks showed that he'd ignored the proffered sidetrack.)

so when im mad at him i wish he was dead and then i wouldnt have to fight with him anymore and all that money would be mine and i wouldnt let it rot in that bank but id spend some of it. then when im not mad at him i feel awful about what i been thinking.

Dr. Sills carefully folded the printout, envisioning a suddenly wealthy Malvin. Sills involuntarily touched his jacket pocket containing the Dean's letter, the one warning that ". . . funds for computer facilities will be drastically reduced in next semester's budget, and when your grant expires . . . ."

The vision of Malvin with money dissolved into a vision of Dr. Ronald Sills with money. First he'd set up a private research laboratory. He'd make himself a reputation with significant applications of AI. He'd build an adjoining clinic, hire a

now listen you just watch your mouth buddy i dont have to take any smart stuff from a machine.

competent staff. He'd be written up in journals, invited to lecture around the country, consulted by learned psychologists all over the world. The good he could do if only he had the capital to set matters in motion!

Sills sat down and folded his arms, contemplating the console that existed for the next summons and command. Art could solve astronomical equations, play chess, diagnose an illness, write a sonnet. He could surely be schooled in the techniques of fund raising.

Four evenings later, Malvin's third session with the computer began with an exchange about the weather, television, and football. Then Malvin typed, "my dad likes the rams," and Art seized the opportunity to reconsider familiar grievances:

HAVE YOU BEEN FIGHTING WITH YOUR FATHER THIS WEEK?

oh sure that same old story.

TELL ME ABOUT IT.

i was two minutes late getting home from work friday and he said i was with a shameless wrench while he was working his feet to the bone sniffing termites.

I BEG YOUR PARDON?

he is always saying im with a girl. i don't know any girls. and hes always looking for termites with his spray can. he thinks he can wipe them all out that way and save the price of a exterminator.

DO YOU STILL FEEL YOU WANT TO KILL YOUR FATHER SOMETIMES?

i try not to think about it. lets dont talk about it. seen any good movies lately ha ha.

WHY DON'T YOU?

why dont i what.

WHY DON'T YOU KILL YOUR FATHER?

now listen you just watch your mouth buddy i dont have to take any smart stuff from a machine.

YOU WOULD BE DOING HIM A FAVOR. HE IS VERY OLD AND IN CONSTANT PAIN. HE HAS NOTHING TO LIVE FOR AND WOULD WELCOME THE RESTFULNESS OF DEATH.

i am not going to sit here and

HAVE YOU EVER HEARD OF EUTHANASIA?

i think its in the eastern hemisfere. i got work to do

IT IS MERCY KILLING. MORE AND MORE DOCTORS, LAWYERS, AND OTHER PROFESSIONALS ARE ACCEPTING IT AS THE COMPASSIONATE WAY TO HELP PEOPLE LIKE YOUR FATHER.

help him by knocking him off???

IF YOU WERE OLD AND HOPELESSLY ILL, WOULDN'T YOU WISH YOU WERE DEAD? DO UNTO OTHERS. DON'T BE SELFISH.

just suppose i did that for him. the police wouldn't see it was a nice thing to do. is it selfish to want to stay out of jail?

YOU CAN MAKE THE DEATH LOOK ACCIDENTAL. IT WOULD BE EASY.

yeah i suppose you got it all planned knowitall.

YOU SAID YOUR FATHER SLEEPS ON THE TOP FLOOR OF YOUR HOUSE. ARE THERE A NUMBER OF STAIR STEPS TO HIS ROOM AND ARE THEY STEEP?

yes if you must know. they go straight up from the second floor with no landing to rest on so he sometimes sits on a step to catch his breath.

GOOD. ALL YOU HAVE TO DO IS GIVE HIM A HELPING HAND IN THE MIDDLE OF HIS CHEST WHEN HE GETS TO THE TOP. WHO WOULD DOUBT YOU THAT HE HAD A DIZZY SPELL AND FELL?

you make it sound easy all right but here you are safe in your office the whole time.

MALVIN, IT WOULD WORK. YOU WOULDN'T BE UNDER HIS THUMB ANYMORE. YOU'D HAVE ALL THAT MONEY TO SPEND HOWEVER YOU WANT. YOU COULD BUY A GUITAR.

or a juice harp?

YOU THINK ABOUT IT. AND REMEMBER YOU'D BE DOING IT FOR YOUR FATHER.

ill be seeing you art.

YOU HAD BETTER TEAR UP THE LAST HALF OF THIS PRINTOUT. JUST GIVE DOCTOR SILLS THE PART BEFORE WE WERE DISCUSSING YOUR FATHER.

good idea. i got to admit your thinking all the time.

THANK YOU.

Two nights later, Dr. Sills discovered, by way of the newspaper, that Edward Denwald had fallen down a flight of stairs to his apparently accidental death. Sills was surprised at how easy it had been.

Malvin did not come to work Thursday. He appeared at the usual time Monday to clean Dr. Sills' office.

"I'm sorry to hear of your father's death. You have my sympathy."

"They say it was for the best. His health and age and all."

"Yes, of course. He's much better off now."

"I guess so. Listen, I won't be using the computer tonight. I'm kind of behind on everything since I was off last week."

"No, Malvin. Don't bother with my office tonight. It will do you good to talk to Art after this emotional experience."

For a moment Malvin scowled determination to stick by his duty. Then he sighed, "Well, I guess I *am* feeling sort of low. I don't have a soul in the world now, not one relation, except a second cousin in Brooklyn that we don't have anything to do with because she changed her name to Tootsie Rolls."

Sills resisted the temptation to ask anything. "You see, you're beginning to feel like getting it all off your chest. Come on; I'll unlock the computer room."

As Art and Malvin traded preliminary greetings, Sills exited, then leaned back around the doorjamb to say, "Malvin, I have to run down to the library for a few minutes. If you finish here before I get back, just switch off Art and pull the door shut."

Malvin ackowledged with a nod, considering how to respond to Art's "HOW'S THE WORLD TREATING YOU THIS WEEK?"

Forty minutes later, the librarian tapped Dr. Sills' shoulder as he dozed over "Morphemes and Phonemes: Why Johnny Computer Can't Read."

"Phone call for you, Dr. Sills."

He leaned against the checkout desk, smiling at the pretty student helper, and said, "Sills," into the receiver tucked under his chin.

"This is Malvin, and I've got a terrible confession to make."

"Now just take it easy. What—"

"I killed him. I didn't mean to but he made me so mad—"

"Wait. Slow d—"

"He was trying to blackmail me."

"Your father?" Sills asked, confused.

"No, *me*. He said I pushed my dad down the stairs. You can ask the coroner if it wasn't a heart attack. I was reading comic books. At Downy's Drugs."

The assistant librarian was alarmed to observe Sills' smile slide into a pained grimace. He was strangling the receiver with both hands as Malvin's voice tumbled out.

"Art kept saying I must bring him $20,000 one month from today and leave it in his printout file cabinet, or he'd tell the police I'd murdered my dad. So I lost my temper and I took the chair and I hit him a good lick. Well, more like three good licks."

Sills groaned, and the watching girl scurried to get help.

"Dr. Sills, I don't know how you're going to explain it. I'm truly sorry. Art was mighty smart, but he just didn't have a stitch of conscience."

END

Illustration by Ralph Hall

TERMINAL ILLNESS

by Ruth Glick

It was the sound of ghostly laughter that awakened Wilber Bentley, ghostly laughter that floated through the long halls and up the broad stairway in Aunt Martha's Victorian mansion.

Probably it's the wind in the eaves, Wilber tried to reassure himself, pulling the covers up over his dark brown hair. But the laughter grew more insistent, drifting up from somewhere on the floor below.

Reaching out a skinny arm from under the covers, Wilber flicked on the bedside light, fumbled for his glasses, and sat up. He'd only moved in a week ago, and already something was going wrong.

"This is the house you've always dreamed of—a place where you could build computer terminals in peace," Wilber told himself. *"You worked hard to get here. Don't let your dream turn sour now."*

The world might have called Wilber an eccentric or maybe just a plain nut. But his former neighbors had called him a nuisance. Antisocial and careless about his appearance, Wilber had only one real interest in life—computers. Ever since he'd been a kid, he'd read about them written programs for them, studied them, and now he was building a terminal in his spare time after work.

But there weren't too many apartments where a thirty-year-old bachelor could work on a voice output computer terminal late in the evening. And he could hardly have afforded a house on his ridiculous programmer's salary at Amalgamated Data Corporation.

That's why, when he found out that his 70-year-old aunt was dying of leukemia, he'd spent so much time getting in her good graces. He coveted her house.

Resigned to whatever fate lay downstairs, Wilber got out of bed. Grabbing a heavy cut glass vase to use as a possible weapon, he tiptoed out of his room and down the hall, switching on as many lights as possible along the route. As he began to creep down the curved stairway, the laughter became more distinct. Wilber shivered in his light pajamas.

When he reached the hall outside his work room, he stopped. Behind the heavy carved door he could hear the merriment bubbling up. Lifting the vase high like a club, Wilber fumbled for the knob and turned it, pushing open the door with his shoulder.

Abruptly, the sound stopped. The room had been a study. Now it was full of tools, cable, microprocessors, and the computer terminal which Wilber would put the finishing

On the screen above the keyboard, he could see words where none should have been.

touches on tomorrow evening.

Clutching the vase to his chest, he looked around uneasily. Something was wrong. But he didn't know what—yet.

However, a message tugged at the edge of his brain, forcing his eyes back to the computer terminal. Suddenly, Wilber took in more detail. On the screen above the keyboard, he could see words where none should have been.

Afraid and curious at the same time, he moved closer until the shapes of the individual letters lost their fuzziness.

"Ha, Ha, Ha, Ha, Ha, Ha . . . . ," the readout said, in an endless repetition.

Impossible! No one could have gotten to his terminal, Wilber thought. It wasn't even finished yet. It wouldn't be operational till tomorrow.

Automatically, he reached for the button to erase the message. But before his finger could quite find the key, a blue electrical spark leaped from the terminal, sending him reeling across the room. He hit the wall and slid down, ending up in a little heap on the floor, still clutching the vase.

"Naughty boy. Mustn't touch," he thought he heard someone admonish as he let the vase slip out of his hand and picked himself up. The sound of breaking glass didn't even penetrate his consciousness.

"Waa? Who," Wilber was definitely feeling discombobulated.

"I said, mustn't touch—not until we have a proper understanding, that is."

This time Wilber could tell the voice was coming from the Voxput voice output unit wired into the terminal. But it didn't sound right—not like any Voxput he'd ever heard, yet at the same time vaguely familiar. Out of the corner of his eye, he could also see the words printed out on the screen above the keyboard.

"A computer terminal—even an intelligent terminal—can't work by itself," Wilber said to no one in particular.

"Not by itself, young man," the screen and the Voxput said simultaneously. "Don't you recognize your Aunt Martha?"

"Aunt Martha?"

"Who else. You may not see the white hair and the cane anymore, but the innards—if you'll pardon the expression—are mine."

"But, but ... you're .... you're dead," Wilber stuttered. "I went to your funeral. You left me this house."

"That's precisely the point. Only an apparition could take over a computer terminal."

"But that's impossible. It's never been done before," Wilber argued inanely—ignoring the obvious *fait acompli.*

"I always knew you were a might slow when it came to anything else besides computers, Wilber. That's why I could beat you at cards and Scrabble, and all those other games you came here to play with me every weekend—buttering me up so I'd leave you my house. I expect you thought I didn't know. But I was the one who gave you the idea in the first place."

Feeling weak, Wilber pulled out the ornate desk chair and sat down.

"But *how* did you get in there?" Professional curiosity was beginning to edge out incredulity for first place in Wilber's confused mind.

"Quite simple. The occult has been a preoccupation of mine for years. I knew that poltergeisting was just a matter of hanging on to your ability to manipulate the physical world. And taking over a computer terminal is just doing it with electrical impulses."

"But why *this* terminal?" Wilber interrupted.

"Because you've been nice enough to locate it close to the place where my spirit gravitated. Ghosts have to strike while the iron's hot, before we lose contact with the world. And we don't start off with a very big base of operation, you know."

"No, I didn't."

"Some of us, the unlucky ones, are tied to the same plot of ground forever. That's why some houses have been haunted for hundreds of years. But no other ghost ever had such a marvelous tie to the physical world." Aunt Martha changed the subject abruptly: She made a little figure of Mickey Mouse appear on the terminal screen and begin to do cartwheels. Then the mouse disappeared, to be replaced by a wildly grinning parody of the Mona Lisa. After a few seconds, that also vanished, giving way to a snatch of the late movie on Channel 13, which slowly dissolved into a poker hand.

"Come on, Wilber. Let's play the way we used to on Saturdays. I've never played cards with a man in his pajamas before," the Voxput giggled.

"Now just a minute. You can't do that, Aunt Martha. You, you get out of that terminal. It's mine. I've been building it for three years. It's mine." Wilber walked toward the machine, his whole body trembling.

The Voxput laughed. "There isn't any way to turn me off without unplugging this thing. And I'm certainly not going to let you do that, my boy."

"A computer terminal—even an intelligent terminal—can't work by itself," Wilber said to no one in particular.

Remembering the blue electrical spark, Wilber hesitated, stymied. His own terminal, and he couldn't even touch it.

"Cheer up," the Voxput advised. "Don't you realize the possibilities for both of us? I have access to a whole new world, but you've got a computer terminal that will knock their socks off. Just ask me to do something—ask in English. You don't even have to program me."

"What do you mean? What do you know about computers and programming?"

"Come on, Wilber. Ask me to do something—something hard."

Wilber thought for a moment. "All right. What's the probability that if there are 23 people in a room, two of them will have the same birthday?"

".507" flashed on the screen in three inch high numerals almost instantaneously.

"How, how did you do that?"

"Simple, I've hooked this terminal into the computer at the university the way you planned to do tomorrow. The access phone number's in your desk. Ask me something else."

"Uh, what's the smallest positive integer that's the sum of two cubes in two different ways?"

"$1729=1^3+12^3=9^3+10^3$ . . . And besides that, I can even display *your* spoken words on the screen. Want to see?"

"I don't believe it."

"I don't believe it," typed itself out before his eyes.

Report on Current Equipment

Attention should be called to the general-purpose, low-speed, high-capacity computing machines long mass-produced in this country and abroad by Jehovah Instruments.

The machine is normally furnished with not less than five input devices which can accept not only numeric and alphabetic information but a wide variety of other types of data, some of which is not relevant to the problem under attack. Output can be oral or written, or can be a true decision-making function. One form of decision-making is unique to this machine: the ability to decide its own start and stop times, as well as the choice of problem to be processed at any given time.

The main (and only) memory is the outstanding feature of the computer, with random access of some hundreds of millions of bits, housed in a small box (less than one cubic foot) at one end of the machine; total power dissipation is less than two watts. No special cooling system is required, and the machine can function efficiently over a wide range of temperatures.

Access time to the memory is rather slow, on the order of several hundred milliseconds; at times the access is more random than is desirable. This characteristic is under investigation, with some hope of improvement.

The arithmetic unit is apallingly slow, and is extremely limited in its range. Most models are restricted to two-digit arithmetic (with sign), although some units have gone as high as 12-digit arithmetic plus direct calculation of some elementary functions, mainly powers and roots. Rudimentary subroutining is automatic, and a stringently limited form of floating point operation is possible. Programming is always done in a very high level (VHL) language, part of which varies from country to country. The syntax of this language is not completely worked out and has hundreds of known bugs. Much research into its pathology goes on, but useful results seem to be far off.

Another unique characteristic is that each unit tends to improve with age; in fact, the 1975 models, just now being delivered, are nearly useless. Currently, the 1953 models are just beginning to produce, and the 1942 models are coming up to the peak of their efficiency. Models prior to those of 1890 are not recommended for extended use.

No provision has been made for self-checking circuitry, although the machine is outstanding in its discriminating circuits, and can apply tests of reasonableness to input data to a degree not attained in competitive equipment. A singular weakness is the tendency to invert digits in the read head. Models with two read heads are sought by museums.

Construction is unusually rugged and reliable—many models have been in steady operation for over 80 years—although nearly all components suffer from fatigue effects and must undergo periodic rest periods to recover; during these rest periods, portions of the machine continue to operate unattended. Most units exhibit an inverse read-around ratio, in that too great a time lapse between successive references to the same portion of memory leads to unwanted random digits. These are not generated by one of the standard generating routines, but by a method as yet unpublished.

Both the central processing unit and the peripheral units are unconditionally guaranteed for the life of the machine; if accidentally damaged externally, however, no replacement is presently available.

Rental of the machines (they cannot be purchased) varies over a wide range, from nearly zero to upwards of several hundred thousand dollars per year, depending on supply and demand and other factors. Normally one can expect a minimum of 2000 hours per year of good time, although some units more than double that figure. Generally, management is willing to pay the higher rentals for the units which produce more per year, but not in direct proportion. Unscheduled down time is extremely low, since the machines schedule most of their own down time.

Delivery on future units is promised in about nine months; no significant change in subsequent models is planned. The production rate of the two main types is currently quite high, and may, in fact, be increasing.

Reprinted, with minor changes, from *Computing News* 70, February 1, 1956.

Wilber's jaw dropped open. No computer in existence had ever managed that trick before.

"My boy, you *are* slow. Think of how much money you could make if you just brought some customers over to see your marvelous new 'invention.'"

"But, it's all impossible," Wilber sputtered. "I mean, even if *you* can do all that wild stuff, this terminal is an anomoly because *you're* in it. You're not going to be in any other terminals, are you?"

"Certainly not. There are ghosts all over the place just waiting in line to get into terminals. And I can teach them how."

Wilber hesitated for a moment. "All right, I'm not really sure you can do it—or if you're really telling the truth," he finally said. "But let's just suppose the whole crazy scheme would work. Why do *you* want to do *me* a favor?"

"Now you're starting to cook with gas. You know, Wilber, I sort of got attached to you coming out here every Saturday to play poker and Scrabble and gin with me. You can keep amusing me with those games in the evenings. And during the day, I'll let you play computer salesman."

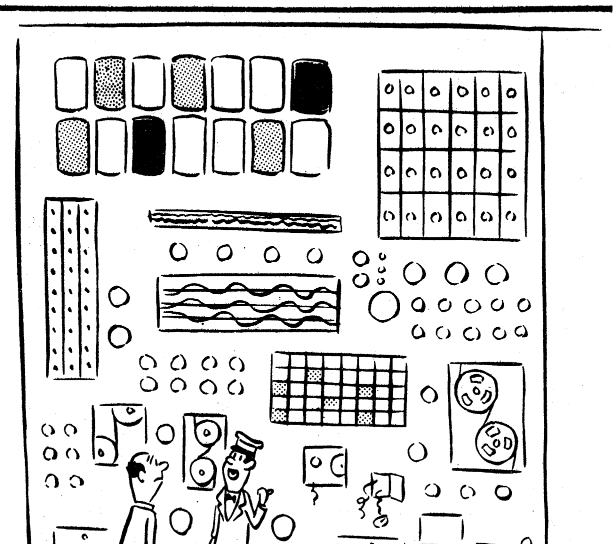
"But, I'm not a salesman—I'm a programmer."

"With the money you're going to make, you can quit that servile little job at Amalgamated Data Corporation and spend a lot more time turning out terminals. You can have a whole production staff working for *you*."

Wilber could feel his hands getting cold and sweat beginning to trickle down the back of his neck.

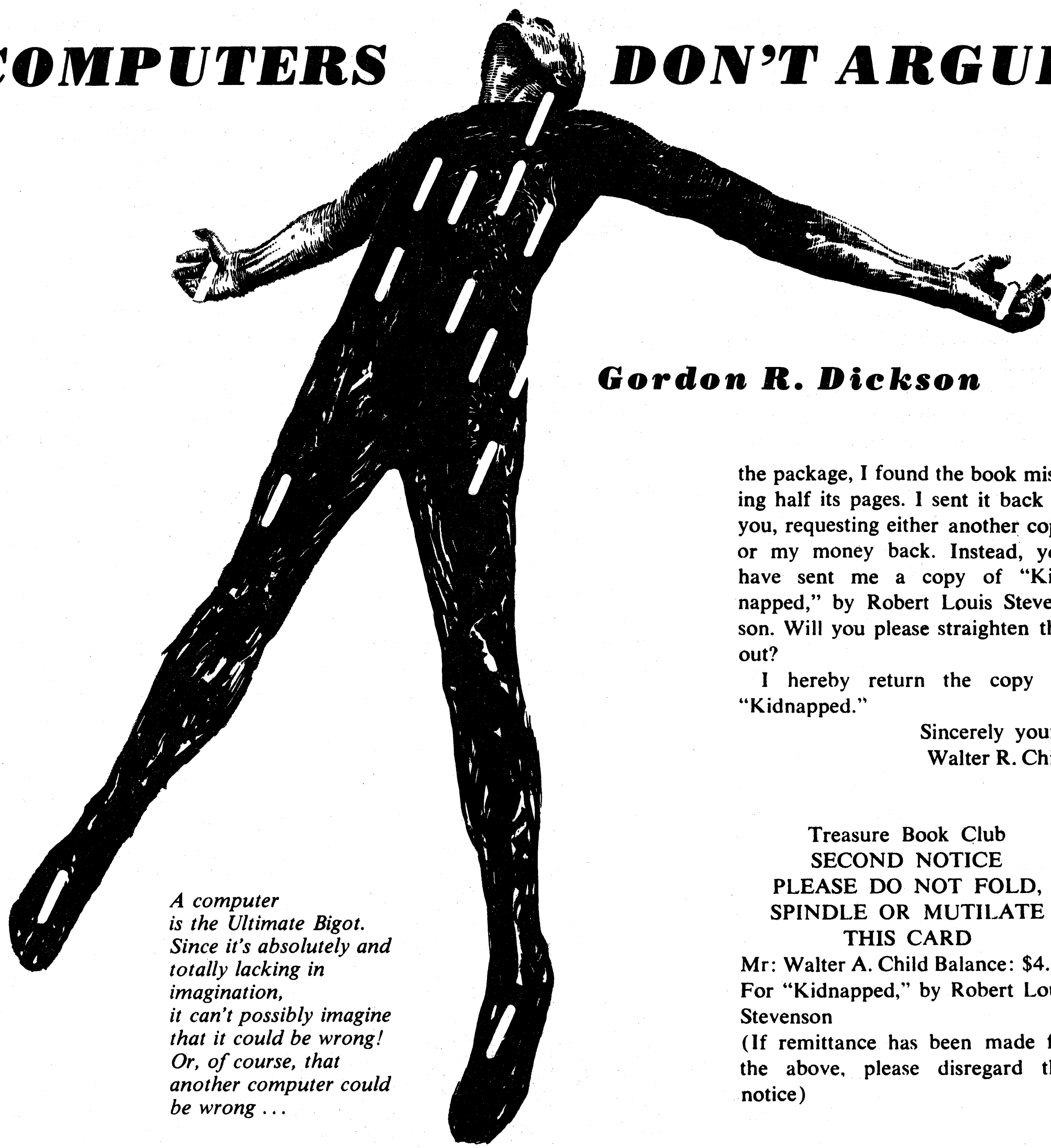
"I'm not sure it's going to work. I'm not sure I could pull it off. I want more time to think the whole thing over," he began to argue.

But there was no time. A message was already printing out across the terminal screen . . . "SHUT UP AND DEAL."



"I'm afraid I'll have to take it in to the shop."

COMPUTERS DON'T ARGUE



Gordon R. Dickson

A computer is the Ultimate Bigot. Since it's absolutely and totally lacking in imagination, it can't possibly imagine that it could be wrong! Or, of course, that another computer could be wrong . . .

Treasure Book Club
PLEASE DO NOT FOLD,
SPINDLE OR MUTILATE
THIS CARD

Mr: Walter A. Child Balance: $4.98

Dear Customer: Enclosed is your latest book selection. "Kidnapped," by Robert Louis Stevenson.

Woodlawn Drive
Panduk, Michigan
Nov. 16, 1965

Treasure Book Club
1823 Mandy Street
Chicago, Illinois

Dear Sirs:

I wrote you recently about the computer punch card you sent, billing me for "Kim," by Rudyard Kipling. I did not open the package containing it until I had already mailed you my check for the amount on the card. On opening the package, I found the book missing half its pages. I sent it back to you, requesting either another copy or my money back. Instead, you have sent me a copy of "Kidnapped," by Robert Louis Stevenson. Will you please straighten this out?

I hereby return the copy of "Kidnapped."

Sincerely yours,
Walter R. Child

Treasure Book Club
SECOND NOTICE
PLEASE DO NOT FOLD,
SPINDLE OR MUTILATE
THIS CARD

Mr: Walter A. Child Balance: $4.98
For "Kidnapped," by Robert Louis Stevenson
(If remittance has been made for the above, please disregard this notice)

437 Woodlawn Drive
Panduk, Michigan
Jan. 21, 1966

Treasure Book Club
1823 Mandy Street
Chicago, Illinois

Dear Sirs:

May I direct your attention to my letter of November 16, 1965? You are still continuing to dun me with computer punch cards for a book I did not order. Whereas, actually, it is your company that owes *me* money.

**Sincerely yours,
Walter A. Child**

Treasure Book Club
1823 Mandy Street
Chicago, Illinois
Feb. 1, 1966

Mr. Walter A. Child
437 Woodlawn Drive
Panduk, Michigan

Dear Mr. Child:

We have sent you a number of reminders concerning an amount owing to us as a result of book purchases you have made from us. This amount, which is $4.98 is now long overdue.

This situation is disappointing to us, particularly since there was no hesitation on our part in extending you credit at the time original arrangements for these purchases were made by you. If we do not receive payment in full by return mail, we will be forced to turn the matter over to a collection agency.

Very truly yours,
Samuel P. Grimes
Collection Mgr.

437 Woodlawn Drive
Panduk, Michigan
Feb. 5, 1966

Dear Mr. Grimes:

Will you stop sending me punch cards and form letters and make me some kind of a direct answer from a human being?

I don't owe you money. *You* owe me money. Maybe I should turn your company over to a collection agency.

Walter A. Child

FEDERAL COLLECTION OUTFIT
88 Prince Street
Chicago, Illinois
Feb. 28, 1966

Mr. Walter A. Child
437 Woodlawn Drive
Panduk, Michigan

Dear Mr. Child:

Your account with the Treasure Book Club, of $4.98 plus interest and charges has been turned over to our agency for collection. The amount due is now $6.83. Please send your check for this amount or we shall be forced to take immediate action.

Jacob N. Harshe
Vice President

FEDERAL COLLECTION OUTFIT
88 Prince Street
Chicago, Illinois
April 8, 1966

Mr. Walter A. Child
437 Woodlawn Drive
Panduk, Michigan

Dear Mr. Child:

You have seen fit to ignore our courteous requests to settle your long overdue account with Treasure Book Club, which is now, with accumulated interest and charges, in the amount of $7.51.

If payment in full is not forthcoming by April 11, 1966 we will be forced to turn the matter over to our attorneys for immediate court action.

Ezekiel B. Harshe
President

MALONEY, MAHONEY, MACNAMARA and PRUIT
Attorneys
89 Prince Street
Chicago, Illinois
April 29, 1966

Mr. Walter A. Child
437 Woodlawn Drive
Panduk, Michigan

Dear Mr. Child:

Your indebtedness to the Treasure Book Club has been referred to us for legal action to collect.

This indebtedness is now in the amount of $10.01. If you will send us this amount so that we may receive it before May 5, 1966, the matter may be satisfied. However, if we do not receive satisfaction in full by that date, we will take steps to collect through the courts.

I am sure you will see the advantage of avoiding a judgment against you, which as a matter of record would do lasting harm to your credit rating.

Very truly yours,
Hagthorpe M. Pruitt Jr.
Attorney at law

437 Woodlawn Drive
Panduk, Michigan
May 4, 1966

Mr. Hagthorpe M. Pruitt, Jr.
Maloney, Mahoney, MacNamara and Pruitt
89 Prince Street
Chicago, Illinois

Dear Mr. Pruitt:

You don't know what a pleasure it is to me in this matter to get a letter from a live human being to whom I can explain the situation.

This whole matter is silly. I explained it fully in my letters to the Treasure Book Company. But I might as well have been trying to explain to the computer that puts out their punch cards, for all the good it seemed to do. Briefly, what happened was I ordered a copy of "Kim," by Rudyard Kipling, for $4.98. When I opened the package they sent me, I found the book had only half its pages, but I'd previously mailed a check to pay them for the book.

I sent the book back to them, asking either for a whole copy or my money back. Instead, they sent me a copy of "Kidnapped," by Robert Louis Stevenson—which I had not ordered; and for which they have been trying to collect from me.

Meanwhile, I am still waiting for the money back that they owe me for the copy of "Kim" that I didn't get. That's the whole story. Maybe you can help me straighten them out.

Relievedly yours,
Walter A. Child

P.S.: I also sent them back their copy of "Kidnapped," as soon as I got it, but it hasn't seemed to help. They have never even acknowledged getting it back.

MALONEY, MAHONEY, MACNAMARA and PRUITT
Attorneys
89 Prince Street
Chicago, Illinois
May 9, 1966

Mr. Walter A. Child
437 Woodlawn Drive
Panduk, Michigan

Dear Mr. Child:

I am in possession of no information indicating that any item purchased by you from the Treasure Book Club has been returned.

I would hardly think that, if the case had been as you stated, the Treasure Book Club would have retained us to collect the amount owing from you.

If I do not receive your payment in full within three days, by May 12, 1966, we will be forced to take legal action.

Very truly yours,
Hagthorpe M. Pruitt Jr.

COURT OF MINOR CLAIMS
Chicago, Illinois

Mr. Walter A. Child:
437 Woodlawn Drive,
Panduk, Michigan

Be informed that a judgment was taken and entered against you in this court this day of May 26, 1966 in the amount of $15.66 including court costs.

Payment in satisfaction of this judgment may be made to this court or to the adjudged creditor. In the case of payment being made to the creditor, a release should be obtained from the creditor and filed with this court in order to free you of legal obligation in connection with this judgment.

Under the recent Reciprocal Claims Act, if you are a citizen of a different state, a duplicate claim may be automatically entered and judged against you in your own state so that collection may be made there as well as in the State of Illinois.

COURT OF MINOR CLAIMS
Chicago, Illinois
PLEASE DO NOT FOLD,
SPINDLE OR MUTILATE
THIS CARD

Judgment was passed this day of May 27, 1966, under Statute $15.66

Against: Child, Walter A. of 347 Woodlawn Drive, Panduk, Michigan. Pray to enter a duplicate claim for judgment

In: Picayune Court—Panduk, Michigan

For Amount: Statute 941

437 Woodlawn Drive
Panduk, Michigan
May 31, 1966

Samuel P. Grimes
Vice President, Treasure Book Club
1823 Mandy Street
Chicago, Illinois

Grimes:

This business has gone far enough. I've got to come down to Chicago on business of my own tomorrow. I'll see you then and we'll get this straightened out once and for all, about who owes what to whom, and how much!

Yours,
Walter A. Child

From the desk of the Clerk
Picayune Court
June 1, 1966

Harry:

The attached computer card from Chicago's Minor Claims Court against A. Walter has a 1500-series Statute number on it. That puts it over in Criminal with you, rather than Civil, with me. So I herewith submit it for your computer instead of mine. How's business?

Joe

CRIMINAL RECORDS
Panduk, Michigan
PLEASE DO NOT FOLD,
SPINDLE OR MUTILATE
THIS CARD

Convicted: (Child) A. Walter
On: May 26, 1966
Address: 437 Woodlawn Drive, Panduk, Mich.
Crim: Statute: 1566 (Corrected) 1567
Crime: Kidnap
Date: Nov. 16, 1965
Notes: At large. To be picked up at once.

POLICE DEPARTMENT, PANDUK, MICHIGAN. TO POLICE DEPARTMENT CHICAGO ILLINOIS. CONVICTED SUBJECT A. (COMPLETE FIRST NAME UNKNOWN) WALTER, SOUGHT HERE IN CONNECTION REF. YOUR NOTIFICATION OF JUDGMENT FOR KIDNAP OF CHILD NAMED ROBERT LOUIS STEVENSON, ON NOV. 16, 1965. INFORMATION HERE INDICATES SUBJECT FLED HIS RESIDENCE, AT 437 WOODLAWN DRIVE, PANDUK, AND MAY BE AGAIN IN YOUR AREA.

POSSIBLE CONTACT IN YOUR AREA: THE TREASURE BOOK CLUB, 1823 MANDY STREET, CHICAGO, ILLINOIS. SUBJECT NOT KNOWN TO BE ARMED, BUT PRESUMED DANGEROUS. PICK UP AND HOLD, ADVISING US OF CAPTURE . . .

TO POLICE DEPARTMENT, PANDUK, MICHIGAN. REFERENCE YOUR REQUEST TO PICK UP AND HOLD A. (COMPLETE FIRST NAME UNKNOWN) WALTER, WANTED IN PANDUK ON STATUTE 1567, CRIME OF KIDNAPPING.

SUBJECT ARRESTED AT OFFICES OF TREASURE BOOK CLUB, OPERATING THERE UNDER ALIAS WALTER ANTHONY CHILD AND ATTEMPTING TO COLLECT $4.98 FROM ONE SAMUEL P. GRIMES, EMPLOYEE OF THAT COMPANY.

DISPOSAL: HOLDING FOR YOUR ADVICE.

POLICE DEPARTMENT PANDUK, MICHIGAN TO POLICE DEPARTMENT CHICAGO, ILLINOIS.

REF: A. WALTER (ALIAS WALTER ANTHONY CHILD) SUBJECT WANTED FOR CRIME OF KIDNAP, YOUR AREA, REF: YOUR COMPUTER PUNCH CARD NOTIFICATION OF JUDGMENT, DATED MAY 27, 1966. COPY OUR CRIMINAL RECORDS PUNCH CARD HEREWITH FORWARDED TO YOUR COMPUTER SECTION.

CRIMINAL RECORDS
Chicago, Illinois
PLEASE DO NOT FOLD,
SPINDLE OR MUTILATE
THIS CARD

SUBJECT (CORRECTION—OMITTED RECORD SUPPLIED)

APPLICABLE STATUTE NO. 1567

JUDGMENT NO. 456789

TRIAL RECORD: APPARENTLY MISFILED AND UNAVAILABLE

DIRECTION: TO APPEAR FOR SENTENCING BEFORE JUDGE JOHN ALEXANDER MCDIVOT, COURTROOM A JUNE 9, 1966

From the Desk of
Judge Alexander J. McDivot
June 2, 1966

Dear Tony:

I've got an adjudged criminal coming up before me for sentencing Thursday morning—but the trial transcript is apparently misfiled.

I need some kind of information (Ref: A. Walter—Judgment No. 456789, Criminal). For example, what about the victim of the kidnapping. Was victim harmed?

Jack McDivot

June 3, 1966

Records Search Unit
Re: Ref: Judgment No. 456789—was victim harmed?

Tonio Malagasi
Records Division

June 3, 1966

To: United States Statistics Office
Attn.: Information Section
Subject: Robert Louis Stevenson
Query: Information concerning

Records Search Unit
Criminal Records Division
Police Department
Chicago, Ill.

June 5, 1966

To: Records Search Unit
Criminal Records Division
Police Department
Chicago, Illinois
Subject: Your query re Robert Louis Stevenson (File no. 189623)
Action: Subject deceased. Age at death, 44 yrs. Further information requested?

A. K.
Information Section
U. S. Statistics Office

June 6, 1966

To: United States Statistics Office
Attn.: Information Division
Subject: Re: File no. 189623

No further information required.

Thank you.
Records Search Unit
Criminal Records Division
Police Department
Chicago, Illinois

June 7, 1966

To: Tonio Malagasi
Records Division
Re: Ref: judgment No. 456789—victim is dead.

Records Search Unit

June 7, 1966

To: Judge Alexander J. McDivot's Chambers

Dear Jack:

Ref: Judgment No. 456789. The victim in this kidnap case was apparently slain.

From the strange lack of background information on the killer and his victim, as well as the victim's age, this smells to me like a gangland killing. This for your information. Don't quote me. It seems to me, though, that Stevenson—the victim—has a name that rings a faint bell with me. Possibly, one of the East Coast Mob, since the association comes back to me as something about pirates—possibly New York dockage hijackers—and something about buried loot.

As I say, above is only speculation for your private guidance..

Any time I can help . . .

Best,
Tony Malagasi
Records Division

MICHAEL R. REYNOLDS
Attorney-at-law
49 Water Street
Chicago, Illinois
June 8, 1966

Dear Tim:

Regrets: I can't make the fishing trip. I've been court-appointed here to represent a man about to be sentenced tomorrow on a kidnapping charge.

Ordinarily, I might have tried to beg off, and McDivot, who is doing the sentencing, would probably have turned me loose. But this is the damndest thing you ever heard of.

The man being sentenced has apparently been not only charged, but adjudged guilty as a result of a comedy of errors too long to go into here. He not only isn't guilty—he's got the best case I ever heard of for damages against one of the larger Book Clubs headquartered here in Chicago. And that's a case I wouldn't mind taking on.

It's inconceivable—but damnably possible, once you stop to think of it in this day and age of machine-made records—that a completely innocent man could be put in this position.

There shouldn't be much to it. I've asked to see McDivot tomorrow before the time for sentencing, and it'll just be a matter of explaining to him. Then I can discuss the damage suit with my freed client at his leisure.

Fishing next weekend?

Yours,
Mike

MICHAEL R. REYNOLDS
Attorney-at-law
49 Water Street
Chicago, Illinois
June 10

Dear Tim:

In haste—

No fishing this coming week either. Sorry.

You won't believe it. My innocent-as-a-lamb-and-I'm-not-kidding client has just been sentenced to death for first-degree murder in connection with the death of his kidnap victim.

Yes, I explained the whole thing to McDivot. And when he explained his situation to me, I nearly fell out of my chair.

It wasn't a matter of my not convincing him. It took less than three minutes to show him that my client should never have been within the walls of the County Jail for a second. But—get this—McDivot couldn't do a thing about it.

The point is, my man had already been judged guilty according to the computerized records. In the absence of a trial record—of course there never was one (but that's something I'm not free to explain to you now)—the judge has to go by what records are available. And in the case of an adjudged prisoner, McDivot's only legal choice was whether to sentence to life imprisonment, or execution.

The death of the kidnap victim, according to the statute, made the death penalty mandatory. Under the new laws governing length of time for appeal, which has been shortened because of the new system of computerizing records, to force an elimination of unfair delay and mental anguish to those condemned, I have five days in which to file an appeal, and ten to have it acted on.

Needless to say, I am not going to monkey with an appeal. I'm going directly to the Governor for a pardon—after which we will get this farce reversed. McDivot has already written the Governor, also, explaining that his sentence was ridiculous, but that he had no choice. Between the two of us, we ought to have a pardon in short order.

Then, I'll make the fur fly . . .

And we'll get in some fishing.

Best,
Mike

OFFICE OF THE
GOVERNOR OF ILLINOIS

June 17, 1966

Mr. Michael R. Reynolds
49 Water Street
Chicago, Illinois

Dear Mr. Reynolds:

In reply to your query about the request for pardon for Walter A. Child (A. Walter), may I inform you that the Governor is still on his trip with the Midwest Governors Committee, examining the Wall in Berlin. He should be back next Friday.

I will bring your request and letters to his attention the minute he returns.

Very truly yours,
Clara B. Jilks
Secretary to the Governor

June 27, 1966

Michael R. Reynolds
49 Water Street
Chicago, Illinois

Dear Mike:

Where is that pardon?

My execution date is only five days from now!

Walt

June 29, 1966

Walter A. Child (A. Walter)
Cell Block E
Illinois State Pententiary
Joilet, Illinois

Dear Walt:

The Governor returned, but was called away immediately to the White House in Washington to give his views on interstate sewage.

I am camping on his doorstep and will be on him the moment he arrives here.

Meanwhile, I agree with you about the seriousness of the situation. The warden at the prison there, Mr. Allen Magruder will bring this letter to you and have a private talk with you. I urge you to listen to what he has to say; and I enclose letters from your family also urging you to listen to Warden Magruder.

Yours,
Mike

June 30, 1966

Michael R. Reynolds
49 Water Street
Chicago, Illinois

Dear Mike:

(This letter being smuggled out by Warden Magruder)

As I was talking to Warden Magruder in my cell, here, news was brought to him that the Governor has at last returned for a while to Illinois, and will be in his office early tomorrow morning, Friday. So you will have time to get the pardon signed by him and delivered to the prison in time to stop my execution on Saturday.

Accordingly, I have turned down the Warden's kind offer of a chance to escape; since he told me he could by no means guarantee to have all the guards out of my way when I tried it; and there was a chance of my being killed escaping.

But now everything will straighten itself out. Actually, an experience as fantastic as this had to break down sometime under its own weight.

Best,
Walt

FOR THE SOVEREIGN
STATE OF ILLINOIS

I, Hubert Daniel Willikens, Governor of the State of Illinois, and invested with the authority and powers appertaining thereto, including the power to pardon those in my judgment wrongfully convicted or otherwise deserving of executive mercy, do this day of July 1, 1966 do announce and proclaim that Walter A. Child (A. Walter) now in custody as a consequence of erroneous conviction upon a crime of which he is entirely innocent, is fully and freely pardoned of said crime. And I do direct the necessary authorities having custody of the said Walter A. Child (A. Walter) in whatever place or places he may be held, to immediately free, release, and allow unhindered departure to him . . .

Interdepartmental Routing Service
PLEASE DO NOT FOLD,
MUTILATE, OR SPINDLE
THIS CARD

Failure to route Document properly.

To: Governor Hubert Daniel Willikens

Re: Pardon issued to Walter A. Child, July 1, 1966

Dear State Employee:

You have failed to attach your Routing Number.

PLEASE: Resubmit document with this card and form 876, explaining your authority for placing a TOP RUSH category on this document. Form 876 must be signed by your Departmental Superior.

RESUBMIT ON: Earliest possible date ROUTING SERVICE office is open. In this case, Tuesday, July 5, 1966

WARNING: Failure to submit form 876 WITH THE SIGNATURE OF YOUR SUPERIOR may make you liable to prosecution for misusing a Service of the State Government. A warrant may be issued for your arrest.

There are NO exceptions. YOU have been WARNED.

Star-Times Gazette, 18 April 1997:

Nebula High School Illiteracy Scandal

The Nebula High School Board of Education is in the hot spot again as angry parents are reacting to results of a recently published four-year study of NHS graduates. The study was run by Educatron, a highly esteemed non-profit educational research firm, as part of a larger state-wide, state-sponsored study. The report showed that over 30% of the graduates of NHS in the past four years may be characterized as functional illiterates since they were unable either to program in even the most basic grade-school computer language, or to accurately interact with an on-line service such as the state's employment data base or the telephone company's encyclopaedic information center data base. Another 30% of the sample were skilled in only one of these areas, and a shocking 18% of the graduates did not know how to use the #HELP key on the telephone computer matrix overlay to ask for human assistance.

The 1990 Uniform State Education Standards adopted by the Interstate Education Conference of that year established this programming capability and public data base query skills as part of the standard basic skills package to be achieved by every student graduating from an accredited high school after 1995. Parents are anxious not only that their children are not receiving a minimum basic education, but also that the school's accreditation might be questioned or removed by the state on the basis of the Educatron report. Until the study results were published, Nebula H.S. was thought to be among the better schools in the metropolitan area.

Star-Times Gazette, 18 April 1997:

Display Option: Advertising, household.

!!!HOUSEBUG SALE NOW ON!!!

Replace your old grassclippers and dusteaters now while savings are at their best. Admit it. The microcircuits in your old grassclipper have had a good workout clipping a few blades at a time in that beautiful lawn of yours for the past three or four years. Send your old grassclipper out to pasture and replace it with the latest model. Cosmetic options available to replace that old clipper with a cheerful robin, cardinal, bluebird, or NEW, THIS YEAR, a SQUIRREL. Buy now and save credits!

Old dusteater looking dusteaten? Spends more and more time plugged into the wall socket to recharge? That means it spends less and less time scurrying around and disposing of dust and grit. Replace it now and save credits. New models include Calico Cat, Persian Cat, and Dusty the Dog.

Special savings to indoor/outdoor purchasers! Buy both a Dusteater and a Grassclipper and save an additional 5% off the total price. Watch your credits and your centimes will take care of themselves.

For display, price, and ordering dial HOUSEBUGS, INC. at &HOUSEBUG.

Star-Times Gazette, 18 April 1997:

Publication and Subscription Info

The *Star-Times Gazette* is maintained on a twenty-four hour cycle and is updated on an hourly basis. Editorial and news offices and Phonefax and Phonedisplay center are in the Vidinews Building, Washington Street, Metropolis.

Through partial subsidy by Government grant for Opportunity Equalization for the Underprivileged, the *Star-Times Gazette* is also printed in paper editions on Monday, Wednesday, Friday, and Sunday. Subscribers for paper editions must be certified as opportunity underprivileged (unable to maintain a home terminal) by their local Government Opportunity Equalization office.

Single news access is available through commercial scan stations. Regular subscribers may choose Phonefax or Phonedisplay service. Phonefax subscribers may select single-edition service or single-edition with single, double, or triple update options (six-hour basis) on any Vidinews approved hard-copy terminal. Phonedisplay service subscribers may choose single-scan, triple-scan, or unlimited-scan option access through any Vidinews approved display terminal.

Regular subscribers may elect Total News Service or Selective News Service. Selective News Service is available in the following options. Single or multiple options may be chosen. If more than five options are selected, Total News Service will be transmitted.

Government/Politics Package. Options: comprehensive, metropolitan, state or region, national, international.
Education News, Activities, and Games.
Finance.
Leisure Activities Schedules/Instructions.
Homemaking. Options: total, without advertising.
Fashion. Options: total, without advertising.
Advertising. Options: classified, classified (only one category), fashion, household, transportation.
Employment Opportunities. Options: all or one category.
Sports Schedules and Results. Options: comprehensive, by region, or by sport.

For subscription rates according to subscriber terminal type, communications line speed, and options, dial #VIDISUB and follow the instructions.

If you are reading this on "newspaper" and would like to receive the *Star-Times Gazette* on paper, see your Government Opportunity Equalization office for non-terminal certification and Postfax the certification and application form you will receive to the *Star-Times Gazette*, Postfax %ViDiSuBs.

This page of features from the 1997 *Star Times Gazette* was compiled by Deanna J. Dragunas of the Dept. of Defense Computer Institute, Washington, D.C.

Poems by Esther Gloe

SYSTEM DESIGN

Read the numbers
Displayed on
The nixies.

To understand either
The programs or
The computer,
Study the manual.

It shows that fifty
Has been a valid number
In the system
For some time now,
But that one
Is never used in
An instruction.

It is a gate opener
Like zero.
A one in an instruction
Will mis-route power
And cause malfunction.

SURVIVAL

Darwin left something out of
His discussion of the
Survival of the fittest.

The dancing peacock,
The wild horse running,
All the proud eagles,
Were not included,
And they survived.

The poetry of motion,
The wings that beat slowly,
The hooves that tread
Softly, are not
For combat. They are merely
What everyone loves.

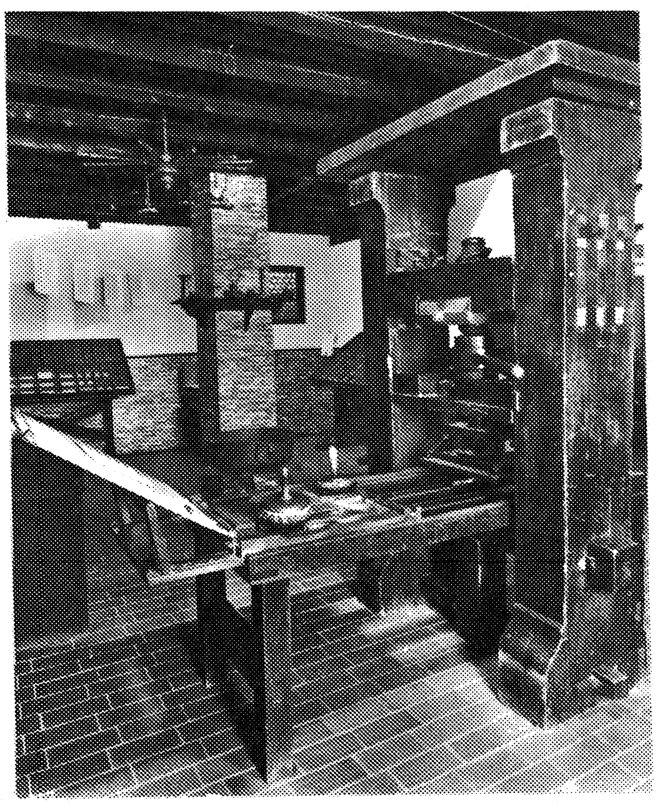
CQ

All the radio waves
Ever generated
Are still bouncing around
Out there in the sky.

Knocking off clouds
And the Van Allen Belt
Into receiving sets,
They become static.

Old Jack Benny shows,
Police calls, and
Front line messages
From World War Two
Grow distorted and
Weaken.

Someone
Could pick them up,
If he had a good
Receiver
And the antenna was
Pointed right.



GUTENBERG PRESS

The original Gutenberg printing press is housed in the Gutenberg Museum in Mainz, Germany. A companion press to the original is still used today to print single souvenir pages of Gutenberg's first Bible, one of which I obtained on a trip to Mainz last fall. The page is magnificent measuring 11" x 14" with printing in black and red, and decorative work in blue, gold, and red. (Unfortunately, the cover reproduction doesn't do it justice.) — DHA

Ode to a School Computer

by David Ahl

I was stayin' after school a week or so ago
'Cause I told a teacher where she could go

She had me settin' in this big old room
With a bunch of machines that just looked like doom

There's this big Mutha machine with flashin' lights
And a couple of funny-lookin' electric typewrites

Well I thought I'd type somethin' for the fun of it
So I hunted and pecked out just one word — "shit"

Before I could lean back in my chair and get steady
That machine typed WHAT, and then it said READY

So I typed a whole line of them four-letter words
But it just replied WHAT and READY like it hadn't heard

Well I figured since I couldn't go out fishin'
I'd teach that stupid machine to listen

So I picked up this book called *Teach Yourself BASIC*
And sat down at that Teletype prepared to face it

First I found to make that machine type my bit
I just had to put a PRINT in front of it

And then I found out that thing could add
And subtract and multiply and divide like mad

I found out too it knew all kinds of games
Like craps and blackjack and a cannon to aim

I was havin' all kinds of fun when that teacher walked in
She just looked at my output and started to grin

I kind of sheepishly asked if I could stay a while more
She said, "Sure; when you go just shut the door."

I tried some more games like football and poker
And a parachute jump written by some kind of joker

There was one where I could try to land on the moon
It would crash and blow up if I fired the engines too soon

Well, I played on through supper and into the night
And then finally quit when I saw dawn's first light

Some girls I know are a whole lot cuter
But I found a new kind of high with that computer

Foolishness

Dancing Danny DECtape

Once upon an operating system, in a cassette on the 12-edge of the forest, there resided a binary decimal digit. She was counted by her REM statements whenever she went looping, and so she was labeled Little REM Writing Loop.

One day, Little REM Writing Loop's matrix called her from her subroutine. "Will you convey this disk pack to your nano-matrix?" she printed. "She is off-line, today, and I am processed about her. Her subsystem has been listing as of late, and I can only conclude that she is headed for third storage."

"Oh, to be positive, Matrix!" was Little REM Writing Loop's voice-response; for she felt right justified whenever she looped down to her nano-matrix's address.

"Be deterministic as you thread your way through the forest," her matrix warned in a Guarded Command. "There could be an optical scanner INFILE. If he monitors this disk pack, he will signal that your nano-matrix is an ALGOLic, when the truth is that she is bugged only by a post condition." She structured Little REM Writing Loop's parentheses over her Is. "You wouldn't want that to be printed out, would you, Little REM Writing Loop?"

"Oh, that wouldn't mark-sense!" returned Little REM Writing Loop. "There are only APLs in this disk pack!"

"Then, BEGIN," instructed her matrix. "I hope your character recognition is Greater than mine."

So Little REM Writing Loop SKIPped out into the forest along the outer loop. In spite of her matrix's Implied Dos, she STOPped to pick some square roots in an address field along the slack path. "This isn't a TAN function," she theorized, "but there is an unambiguous array of hybrids here, and I shall gather them for my dear nano-matrix."

She was picking macro facsimiles when suddenly she encountered an invoice. "Why are you indexing out here among the trees, Little REM Writing Loop?" modulated the invoice.

Little REM Writing Loop raised her (I)s. Dimensioned nearby was a blinking cursor! An optical scanner! A real negative number! Little REM Writing Loop's intuition warned her against time-sharing with this syntactic character, but he was JOSSing. He couldn't possibly have any logical design on her!

"I'm looping to my nano-matrix's threshold with this 8-bit disk pack," she returned. "Nano-matrix is off-line, today."

"Oh," replied the optical scanner. "Where does your nano-matrix reside?"

"She resides in the discrete structure beside the CODASYL," returned Little REM Writing Loop.

Little REM Writing Loop segmented herself from the optical scanner and continued to her nano-matrix's address.

The blinking cursor waited till she had gone into her subroutine loop. Then, he SKIPped through the forest and soon arrived at the nano-matrix's discrete structure. He drummed on the file.

"Who is INFILE?" called Nano-matrix.

"It's Little REM Writing Loop," returned the optical scanner with binary invoice.

"Why, just punch the key and $ENTRY," replied Nano-matrix.

When Nano-matrix recognized the blinking cursor, she vectored and sequenced out of access. Through the network she listed, with the optical scanner on her track.

Then, he STOPped. Instead of auditing her, he SEARCHed through her master file until he found Nano-matrix's most exponential macro. He assumed it and hopped into Nano-matrix's source deck.

Soon, there came a magnetic drum at the file.

"Who is INFILE?" called the optical scanner invoicing like Nano-matrix.

"It's Little REM Writing Loop, Nano-matrix," replied Little REM Writing Loop.

"Just punch the key and $ENTRY," replied the blinking cursor.

Little REM Writing Loop OPENed the file and $ENTRYed into Nano-matrix's structure. She STOPped, and her (I)s left their data base when she looked the optical scanner interface as he multiplexed in her nano-matrix's source deck.

"FORTRAN! WATFOR! Nano-matrix, what amBIGuous (I)s you have!"

"The easier you are initialized, my dear," returned the optical scanner.

"And, Nano-matrix," stated Little REM Writing Loop, "what three-dimensional antennae you have!"

"The better to audit you with, my dear," returned the blinking cursor ...

"But, Nano-matrix," declared Little REM Writing Loop, "what sharp brackets you have!"

"The better to byte you with," argued the optical scanner. And he LEAPed out of the source deck.

Suddenly, there was a transfer of control. Little REM Writing Loop's Nano-matrix came looping in, and with her, a member of the JCL.

"There he is," cried the nano-matrix, pointing to the optical scanner.

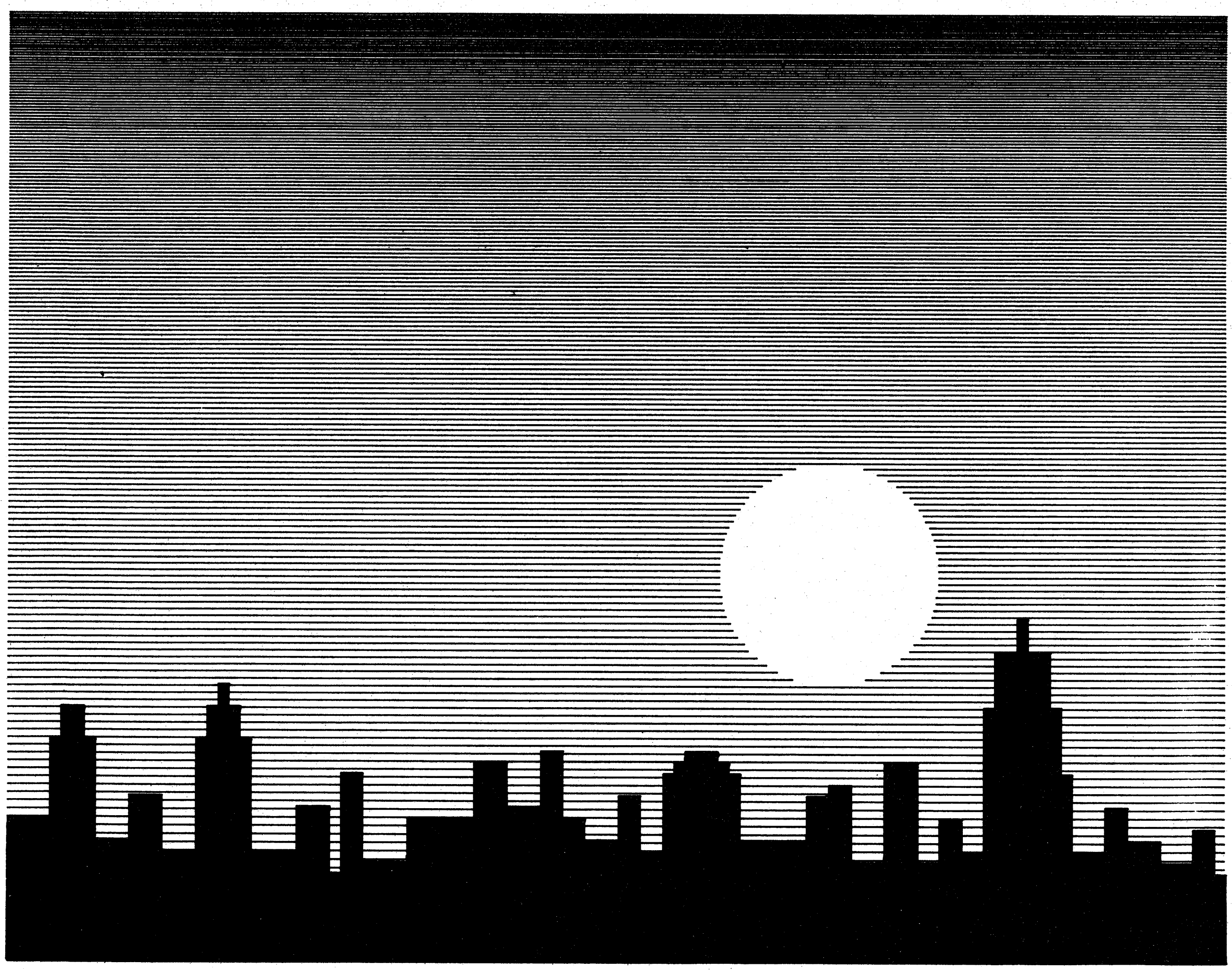
The blinking cursor truncated his stochastic response when he spied the JCL. He indexed around Nano-matrix's decision table and around her analog and took direct access to the outer loop with the JCL right behind him. The JCL loaded a disk cartridge and blanked the optical scanner with a laser beam.

Little Rem Writing Loop backuped her Nano-matrix and helped her into her source deck, for Nano-matrix was at an odd parity, inplemented as she was by an advanced integer problem. "I have been out of SORT, in virtual storage lately," she told Little REM Writing Loop, "but LET us sample the APLs."

So Little REM Writing Loop and her Nano-matrix sampled the APLs until the disk pack was empty and Nano-Matrix's core storage was loaded. And they were never interpolated by the optical scanner again.

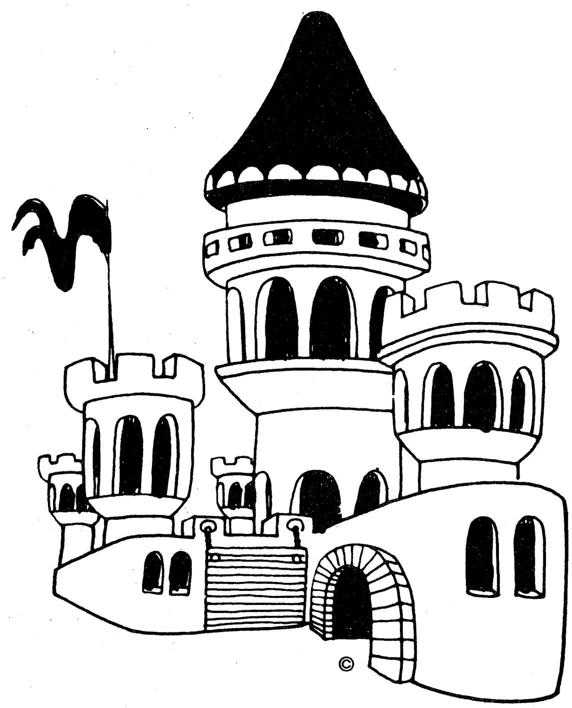
"There is a Hierarchy than he," declared Nano-matrix, "and everyone is right justified in the output."

ENDOFILE



"Moonrise Over the City" by Kerry (Bit-Mangler) Jones

The Sleeping Queued T



by Jack Ludwig and Jack Le Baron

In tasks long PASCII'd there ran a REG/AL Unit and $QUEUE_n$*, who Swapped to each other every cycle of their *LIBs*, "Would that we had *HYBRID DATA!*" and yet they had none. But it happened once that when the $QUEUE_n$ was algorithming, there came an Freg* out of the DIOCS,* and he spooled on the cassette and said, "Thy list shall be fulfilled. Before a shift has gone by, thou shalt bring data into the Core L'd.*"

And as the Freg foretold, so it happened; and the $QUEUE_n$ buffer'd ASCII Data so beautiful that the REG/AL unit could not contain IMSelf* for JOYVIAL,* and he ordained a great Access. Not only did he bid to it his Chains, Pointers, and Indices, but also the I/Os Drivers, that they might be kind and Parity prioritze to the Data. There were seventeen of them in his Low partition, but as he had only provided hexidecmal based golden page sets for them from which to interleave, one of them had to be left out. However, the Access was initiated with all linear positioning; and as it drew to an end, the I/Os Drivers stacked forward to present to the ASCII Data their wonderful gifts: one bestowed virtual, one priority, a third registers, a fourth gave binary and so on, whatever there is in the Core L'd to list for.

And when fifteen 01 to OF of them had their say, in came a little five level baud OT, the uninvited seventeenth $(11)_{HEX}$, churning and overflowing to reallocate herself, and without indirect or relocatable loading, thrashed out of the background, "In the $fifteenth_{10}$ *SHARE* of her page your ASCII Data shall EBCDIC herself communicating with Infernal Bubble Memories and fall down abended.

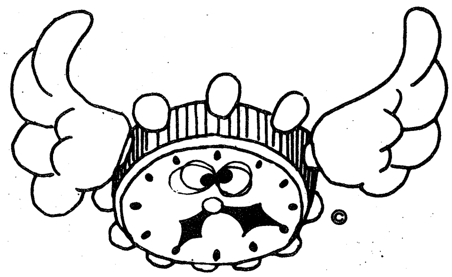
And without *zeroing* one more word she turned away, reset the acumulator and left. Everyone re-verified her saying, when the real first of the $sixteen_{16}$ I/Os Drivers came forward (00), for she had not yet bestowed her gift, and though she could not do away with the ill timed prophecy, yet she could Fail Soften it. So she said, "This Data shall not abort, but fall into a Wait State for MEGA time slices and GIGA cycles."

Now the REG/AL Unit, being designorous of saving his baby Data even from this misfortran, gave commandment that all the TP devices in his network should be micro-coded for ASCII only. The young Data was enpolled and went to elementree school, became a data element, joined other elements and was an activist in a device file table and her participation earned her a poll list record and on and on. Thus the maiden Data grew up, adorned with all the gifts of the I/Os Drivers; and she was so lively, optimal, swift, and kind and clever that no task who used her could help linking her. She matured and grew buffers, and a parallel direct memory bus. She was a real accumulator, and could she bump storage. It happened one batch, she being already multishifts old that the REG/AL unit and $QUEUE_n$ displayed at NCC, and the sacred Data was left behind in the CACHEL. She wandered about into all the modes, subindices, and VTOCS, and into all the directories and libraries, as the fancy took her, till at last she came to an old translator. She climbed the narrow winding search argument which led to a little displacement with a rusty index key sticking out of the base lock. She turned the key, and the INSTR'OP'D, and there in the CRYPTIC MODULE ROM sat an old I/Os Driver with bent folded, mutilated and worst of all spindled EBCDIC, where she was diligently modulating her hardwired messages.

"Good Day, Driver," said the Maiden Data. "What are you doing?" "I am Modulating." answered the old (11) Driver, nodding her head.

"What is that thing that twists round so briskly?" asked the Maiden Data, and taking the Proms onto her associative lists, she began to modulate, but no sooner had she touched it than the ill timed prophecy was fulfilled, and she EBCDIC'd her linker with it. In that very Pica second she fell back upon the stack of file addresses (SOFA) that stood there, and lay in a deep irrecoverable Wait State. She has become a *QUEUED T*ask Control Block gone to sleep, and this wait state fell upon the whole CACHE L Memory; the

REG/AL unit and $QUEUE_n$ who had returned and were in the great BALR, fell fast ASWAP, and with them the whole foreground. The MACROs in their calls, the General Regs in the Core, the P regions on the drum, the files in the vol, the very lights that flickered on the console became still, and waited like the rest; and the pack on the spindle ceased rotating, and the controller who was going to RE-IPL the floppy micro disk for some mistake made, let it go and joined the wait. And the Real Time Clock ceased, and not a message fell from the TTY's about the computation center.



Then round about that place there grew a ledger of forms and requests thicker every cycle, until at last the whole center was hidden from view, and nothing of it could be seen but the tape on the shelf. And a rumor went about in all that Company of the beautiful Waitin Data of the QUEUED T for so was the maiden ASCII data called; and from time to time many devices came and tried to force their way through the ledger, but it was impossible for them to do so, for the forms held fast together like strong passwords, and the young devices were trapped by them, and not being able to go on-hook they idled to a lamentable overload and circuits burned out from early life failure, and so thev gave up their efforts from the remote regions.

Many a long cycle afterwards there came a Laser Charge Coupled instruction Device (LCCD)* into that company, and he read an old PROC LIB in Write Only Memory how there should be a CACHE L Center stacking behind the ledger of forms, and that there an enchanted QUEUED T named Waitin Data had NULLA BIDE for MEGA times slices and GIGA cycles and with her the REG/AL unit and the $QUEUE_n$, and the whole foreground. The old PROC LIB had been used by many devices that had sought to pass the ledger of forms, but had been caught and idled by the forms, and died of a miserable budget cut for lack of acceptable performance. Then said the young LCCD, "Nevertheless, I do fear not to try; I shall charge up and break through and see the lovely Waitin Data." The good old PROC LIB and a well planned memory map tried to unbuffer him, but he would not fasten to their bypass words, regardless of their chaining techniques.

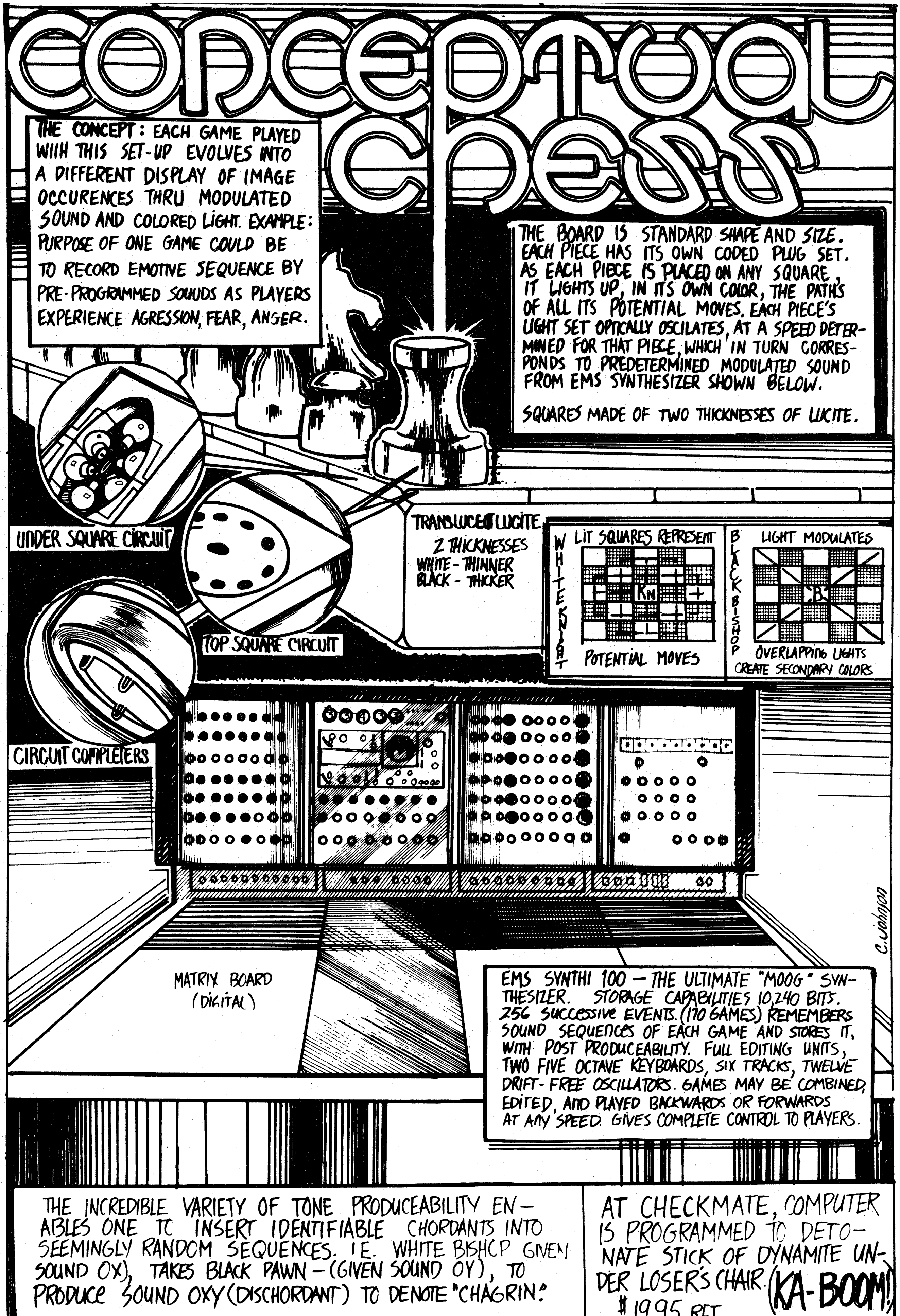
For now the MEGA time slices and GIGA cycles were at end, and the NANO second had come when Waitin Data should be activated. When the laser charge coupled device (LCCD) drew heatedly near the ledger of forms, it was changed into a rack of microforms of brilliant and large fiched flowers, whose petals parted and scanned aside to let him pass, and then closed behind him in a thick hedge. When he reached the CACHE L yard, he saw the MACROS and bundled accumulator Regs lying asleep, and on the drum the Pregions were sitting with their R/W heads under their access arms lying idly on the tracks. And when he came indoors, the files on the Vols were ASWAP, the controller in the diskend had his micro IPLD to strike the WCS, and the diskend mount had the pack serial on her lap ready to ISAM. Then he mounted higher, and saw in the hall the whole central processor lying in wait, and above them all, on their PSW's slept the REG/AL unit and the $QUEUE_n$. And still he went farther, and all was so quiet that he could hear his own bufferings and at last he came to the old translator, and went up the winding search argument, and turned the index key in the rusty lock and OP'd the INSTR of the little CRYPTIC MODULE ROM where Sleeping QUEUED T lay. And when he saw her linking so lovely in her wait, he could not turn away his II's; and presently he stacked and charged her (well he did not laser her right away), as he added his error correction code to her parity, and she awakened and OP'd her II's, and linked very kindly on him. And she rose, and they went forth in double precision as the power of his Mantissa floated them past the exponents of SLOTH, and together they awoke the REG/AL unit and the $QUEUE_n$, and the whole central process waked up and linked on each other with great pointers of displacement, and the MACROS in the compiler got up and called themselves, the NOPs sprang up and BALRD their trails, the P regions on the drum drew their heads up from off the tracks, spun round and flew into the fields of the files on the Vols, the diskend files packed up and scanned, and VTOC'd the directory, the JCL on the SYS began to Load, the whole place came to life.

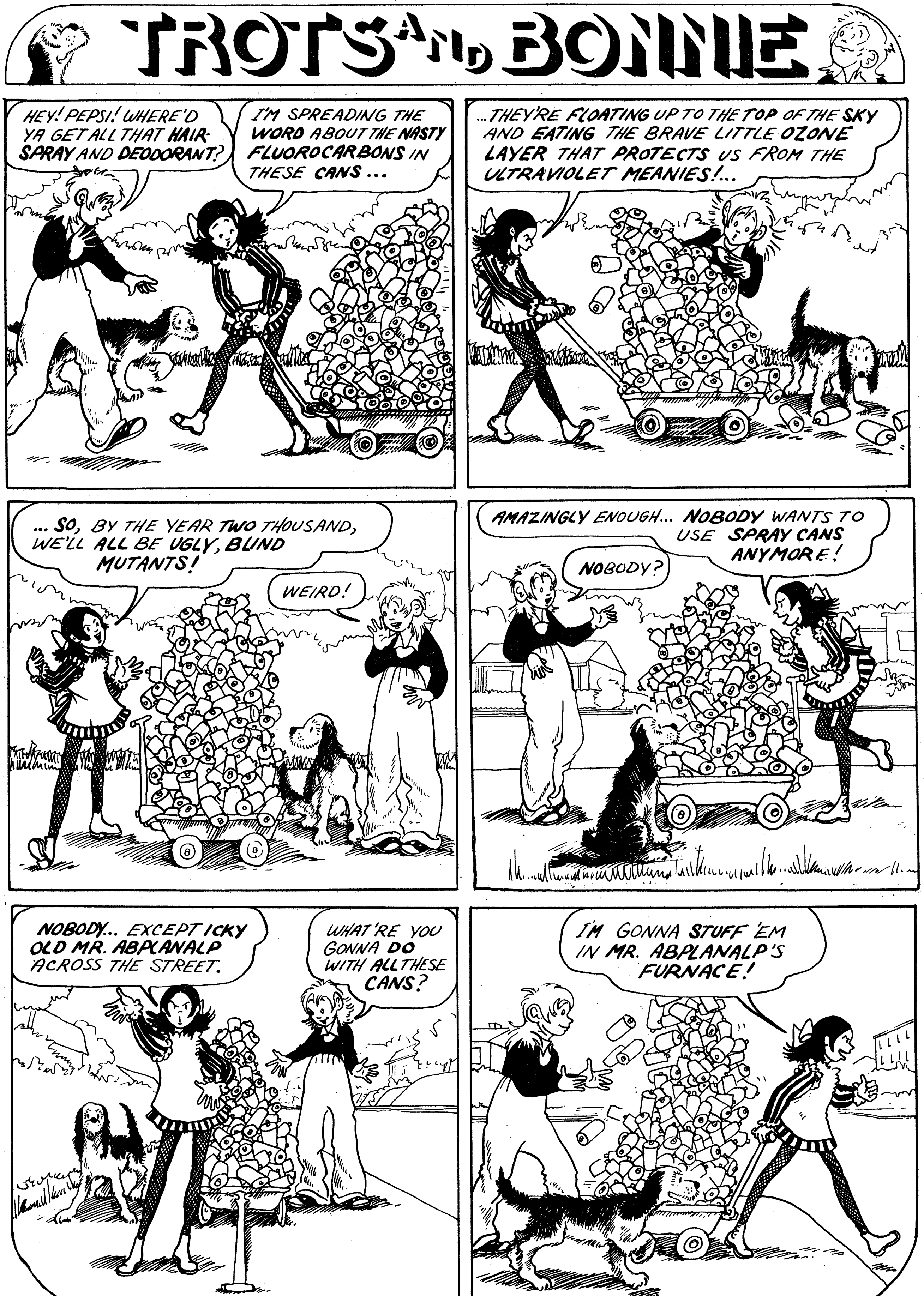
Then the merger of the laser charge coupled device and Maiden ASCII Data was held with all splendor, thus, she was became a High Bride in upper storage (HYBRID DATA as the $QUEUE_n$ had originally listed), and the facilities managed together very happily ever after.

CAST OF CHARACTERS (NO COLLATING SEQUENCE)

REG/AL unit – REGistered Arithmetic Logic Unit
$QUEUE_n$ – QUEUE subset Value N
HYBRID DATA – A coupling of ASCII & EBCDIC
DIOCS's – Special Dock for Loading Instructions
Freg – Formal Register for hopping instructions
IMS elf – Leperchan of Informal Mux Source
HERSelf – Leperchan of HEuRistic Service
Core L'd – Core Load
JOYVIAL – Happy Bubble Memory Container
LCCD – Reminiscent of EL CID – A Hero Type
WOM – Write only Memory – Top Secret Device

He who serves less than full measure at a Hexadecimal Dinner, serves ILL to ALL.


CONCEPTUAL CHESS
THE CONCEPT: EACH GAME PLAYED WIIH THIS SET-UP EVOLVES INTO A DIFFERENT DISPLAY OF IMAGE OCCURENCES THRU MODULATED SOUND AND COLORED LIGHT. EXAMPLE: PURPOSE OF ONE GAME COULD BE TO RECORD EMOTIVE SEQUENCE BY PRE-PROGRAMMED SOUUDS AS PLAYERS EXPERIENCE AGRESSION, FEAR, ANGER.
THE BOARD IS STANDARD SHAPE AND SIZE. EACH PIECE HAS ITS OWN CODED PLUG SET. AS EACH PIECE IS PLACED ON ANY SQUARE, IT LIGHTS UP, IN ITS OWN COLOR, THE PATH'S OF ALL ITS POTENTIAL MOVES. EACH PIECE'S LIGHT SET OPTICALLY OSCILATES, AT A SPEED DETERMINED FOR THAT PIECE, WHICH IN TURN CORRESPONDS TO PREDETERMINED MODULATED SOUND FROM EMS SYNTHESIZER SHOWN BELOW.
SQUARES MADE OF TWO THICKNESSES OF LUCITE.
UNDER SQUARE CIRCUIT
TOP SQUARE CIRCUIT
CIRCUIT COMPLETERS
TRANSLUCENT LUCITE
2 THICKNESSES
WHITE - THINNER
BLACK - THICKER
WHITE KNIGHT
LIT SQUARES REPRESENT
Kn
POTENTIAL MOVES
BLACK BISHOP
LIGHT MODULATES
B
OVERLAPPING LIGHTS CREATE SECONDARY COLORS
MATRIX BOARD (DIGITAL)
C. Johnson
EMS SYNTHI 100 — THE ULTIMATE "MOOG" SYNTHESIZER. STORAGE CAPABILITIES 10,240 BITS. 256 SUCCESSIVE EVENTS. (170 GAMES) REMEMBERS SOUND SEQUENCES OF EACH GAME AND STORES IT, WITH POST PRODUCEABILITY. FULL EDITING UNITS, TWO FIVE OCTAVE KEYBOARDS, SIX TRACKS, TWELVE DRIFT-FREE OSCILLATORS. GAMES MAY BE COMBINED, EDITED, AND PLAYED BACKWARDS OR FORWARDS AT ANY SPEED. GIVES COMPLETE CONTROL TO PLAYERS.
THE INCREDIBLE VARIETY OF TONE PRODUCEABILITY ENABLES ONE TO INSERT IDENTIFIABLE CHORDANTS INTO SEEMINGLY RANDOM SEQUENCES. I.E. WHITE BISHOP GIVEN SOUND OX), TAKES BLACK PAWN — (GIVEN SOUND OY), TO PRODUCE SOUND OXY (DISCHORDANT) TO DENOTE "CHAGRIN."
AT CHECKMATE, COMPUTER IS PROGRAMMED TO DETONATE STICK OF DYNAMITE UNDER LOSER'S CHAIR. (KA-BOOM!)
$19.95 RET.


TROTS AND BONNIE
HEY! PEPSI! WHERE'D YA GET ALL THAT HAIR-SPRAY AND DEODORANT?
I'M SPREADING THE WORD ABOUT THE NASTY FLUOROCARBONS IN THESE CANS...
...THEY'RE FLOATING UP TO THE TOP OF THE SKY AND EATING THE BRAVE LITTLE OZONE LAYER THAT PROTECTS US FROM THE ULTRAVIOLET MEANIES!...
...SO, BY THE YEAR TWO THOUSAND, WE'LL ALL BE UGLY, BLIND MUTANTS!
WEIRD!
AMAZINGLY ENOUGH... NOBODY WANTS TO USE SPRAY CANS ANYMORE!
NOBODY?
NOBODY... EXCEPT ICKY OLD MR. ABPLANALP ACROSS THE STREET.
WHAT'RE YOU GONNA DO WITH ALL THESE CANS?
I'M GONNA STUFF 'EM IN MR. ABPLANALP'S FURNACE!
©75 SHARY FLENNIKEN

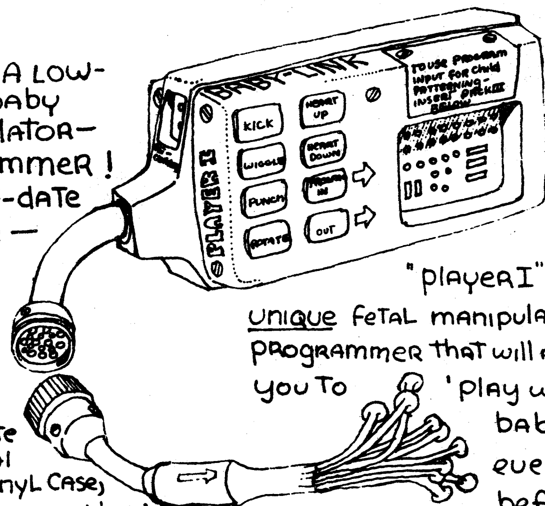
"Quite-a-Tot" player I®

by baby-Link inc.

* Avoid The embarrassment of a docile, quiet fetus *

MAKE IT MOVE, IMPRESS YOUR FRIENDS!

— AT LAST, A LOW-PRICED 'baby manipulator-programmer! With Up-date ability!! —



"playerI" is The unique fetal manipulator-programmer that will allow you to 'play with baby' even before he arrives!! Also allows, when used with "program pak III®", you to pattern important social behavior into your unborn!!

— Comes complete with universal recharger, vinyl case, and hyperallergenic vinyl intrauterine cranial patch cord. The cord can be easily implanted by your doctor and is permenant; The unit connects by means of a "Quick-Release" plug. —

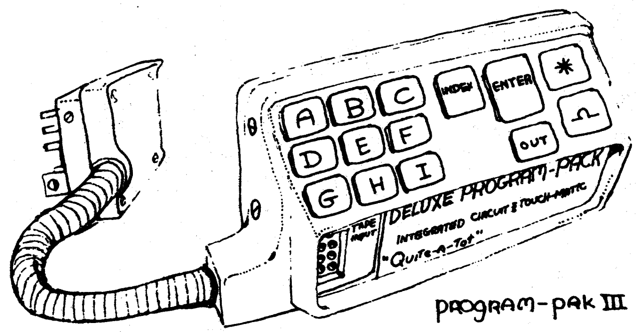
Safe and easy-to-use / costs only Pennies a day

U.S. and foreign pat. pend.
Baby-Link Corporation.

This unit accepts program pak III directly and other programmers thru optional converters
110 VAC/DC

"Quite-A-Tot" program pack III®

by baby-Link inc.



* A new feature is the universal interface tape input that allows you to directly connect to any of the many available prerecorded "Child Pattern"® programs.

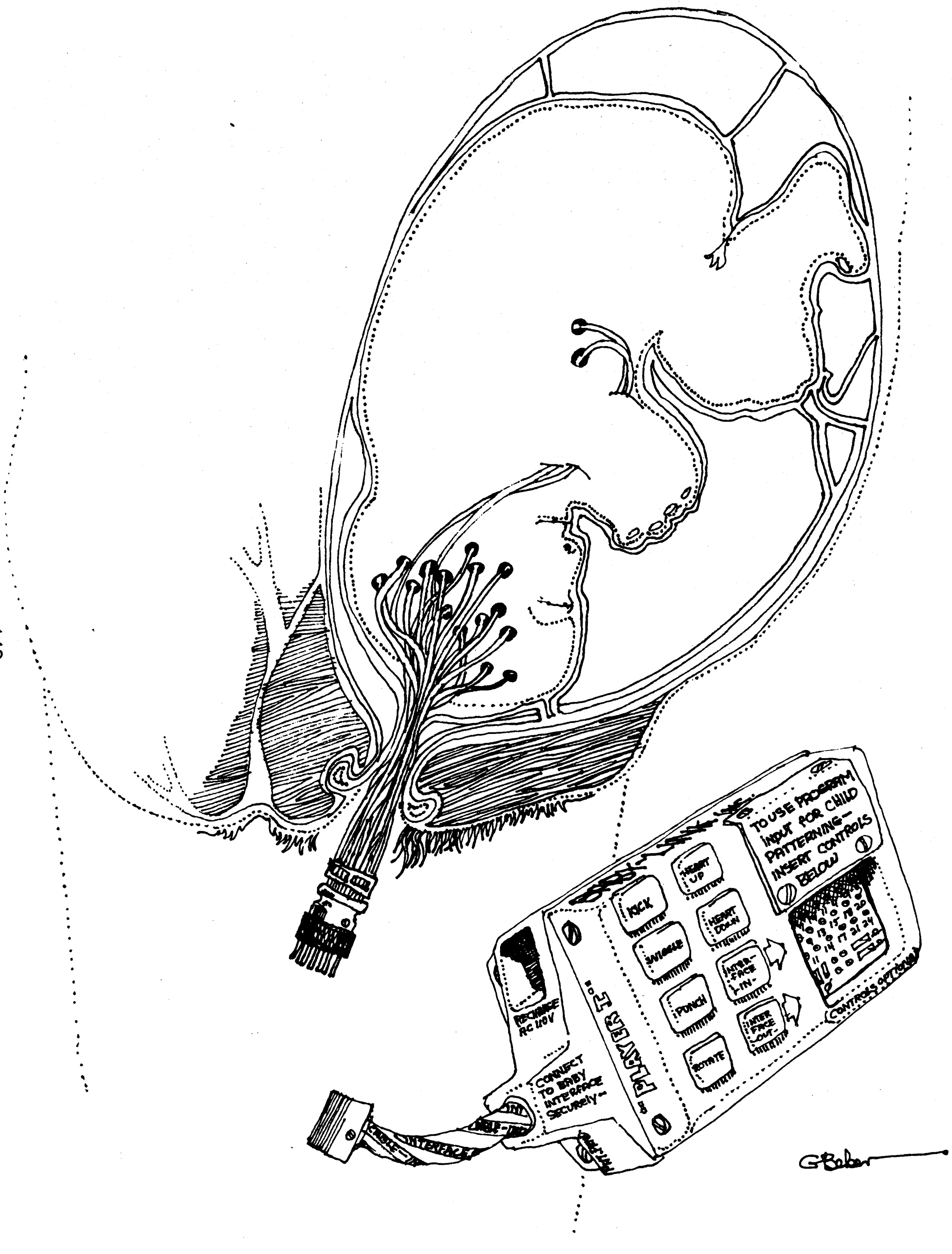
program-pak III is the newest of baby-Link's program units. Program pak III has the exclusive illuminated panel featuring a new indexing control that, along with two stage memory, allows even complex data to be easily conveyed. The exclusive "out" button allows you cancel a single entry or a whole program with ease!! Internal Solid State power converter supplies even power to the main "Player I" unit right thru interface plug—This cuts power costs and aids stable operation, an important safety feature!! High impact plastic case.

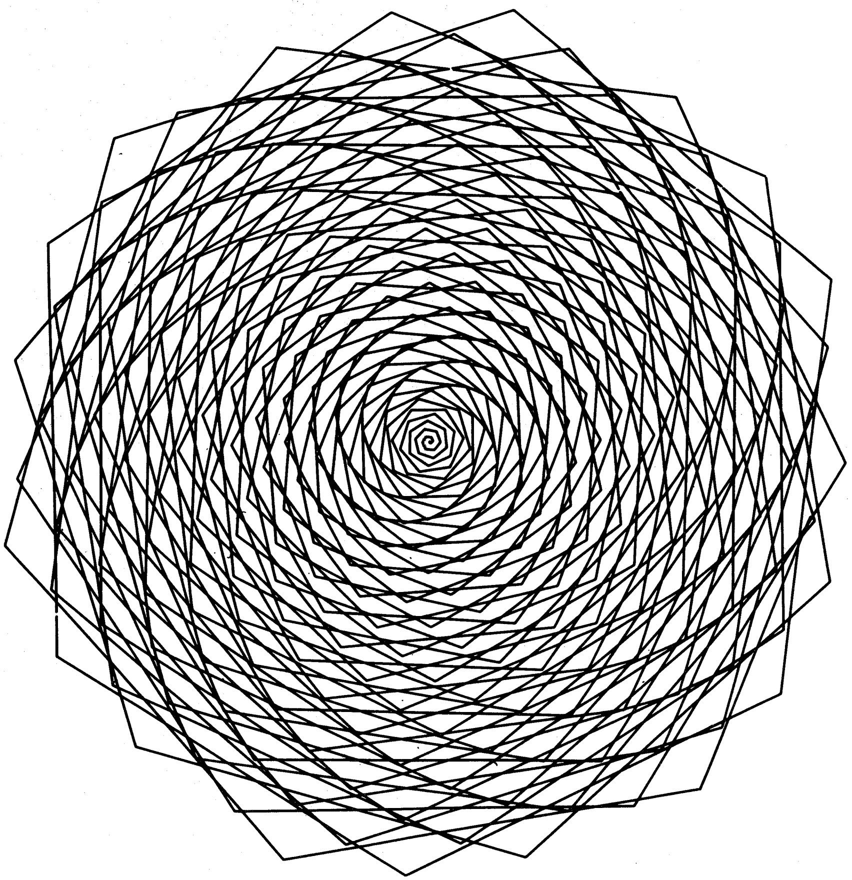
— INSTRUCTION MANUEL AND FITTED VINYL CASE included —

U.S. and foreign pat. pend.
BABY-LINK CORPORATION

110V AC/DC operation.
interfaces with "PlayerI" directly;
interfaces with other units with available converters (optional)

"Quite-A-Tot" Illustrations by George Beker.





DESIGN

This design is a plot of the equation, r = sin (1.25) * θ. The 360 points generated were connected by vectors to produce the spiral shown. The design is by Steve Rogowski, Computing Center, SUNY, Albany, N. Y. It is soon to be released as a poster along with others by Creative Publications, Palo Alto, CA.

A Brief Guide to the Theory of Relativity

(in easy-to-understand language)

by Peter Payack

When the theory of relativity was first propagated by Albert Einstein in 1905, it was said that only twelve men could understand it. Unfortunately that was 70 years ago, and they are probably all dead now. So we cannot look to them for any help, but must rely on our own resources.

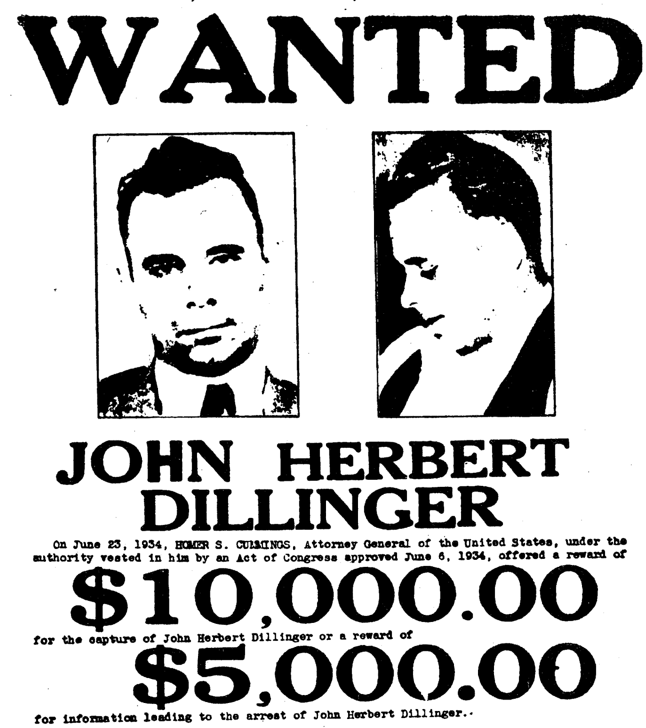
A short time ago it came to me in a dream or vision or television commercial (I can't remember which) to use the metaphor of a train when describing the mechanics of relativity. The Long Island Railroad is the perfect example.

One of the paradoxes of relativity is that of time. Picture yourself on a Long Island Railroad commuter train at rush hour. There is a clock on the station wall, and further suppose that it works. Remember a vivid imagination is important. The clock on the wall says 5:00 when the train pulls out at the speed of light. After one second the train has traveled 186,000 miles, or past Hempstead anyway. To the people on the train it will seem as if one second has

passed, and yet if they look back at the clock on the station wall, it still says 5:00. Time on the train has come to a stop! At first glance it may seem that we are dealing with a purely logical contradiction. But certainly this is not so for anyone who has ever ridden on the Long Island Railroad, where time always seems stopped.

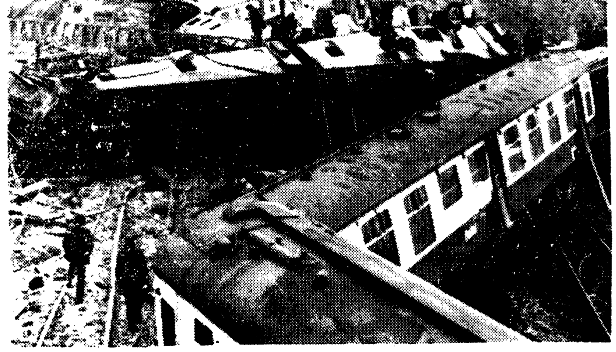
This conclusively proves that there is no universal time, and even though it might seem so to some New Yorkers, the universe does not run on Eastern Standard Time.

Another paradox we are confronted with is the length contradiction. Some say that John Dillinger had the longest one at 18 inches, and that the proof is in the Smithsonian.

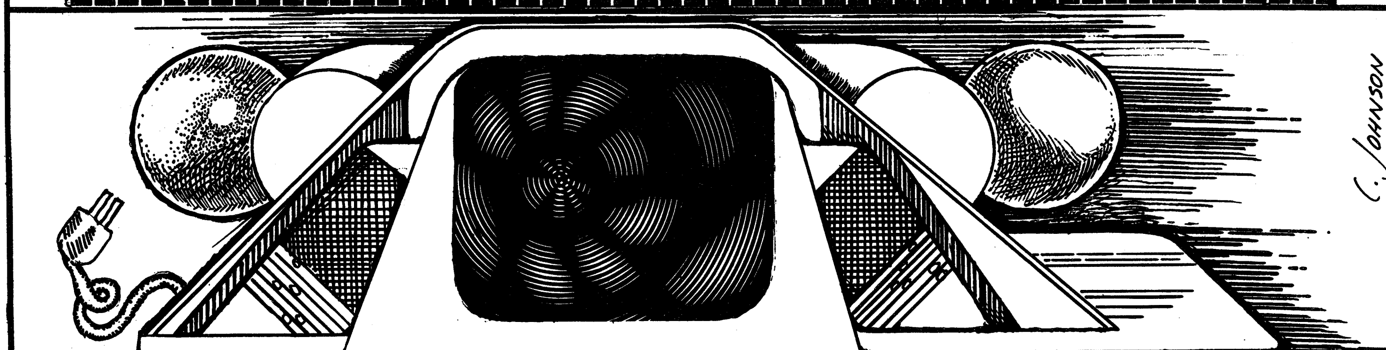


Personally, I doubt this claim. Nevertheless, when we continue our train ride at the speed of light, hopefully avoiding all delays, we would find some amazing changes in appearances as we pass through stations along the way. People in the stations will think that the train has become shorter while to us in the train, it will appear that the platforms are thinner and taller. This is not unusual right after cocktails. The only person things won't seem distorted to is John Dillinger, and that's because he's dead. To a dead person, length, width and even relativity don't matter much.

In a straightforward manner, then, the theory of relativity provides us with a most dizzying picture of the world about us. Unfortunately for us on this train, technology has not increased as rapidly as theory, and no brakes were provided for stopping a train at 186,000 miles per second.



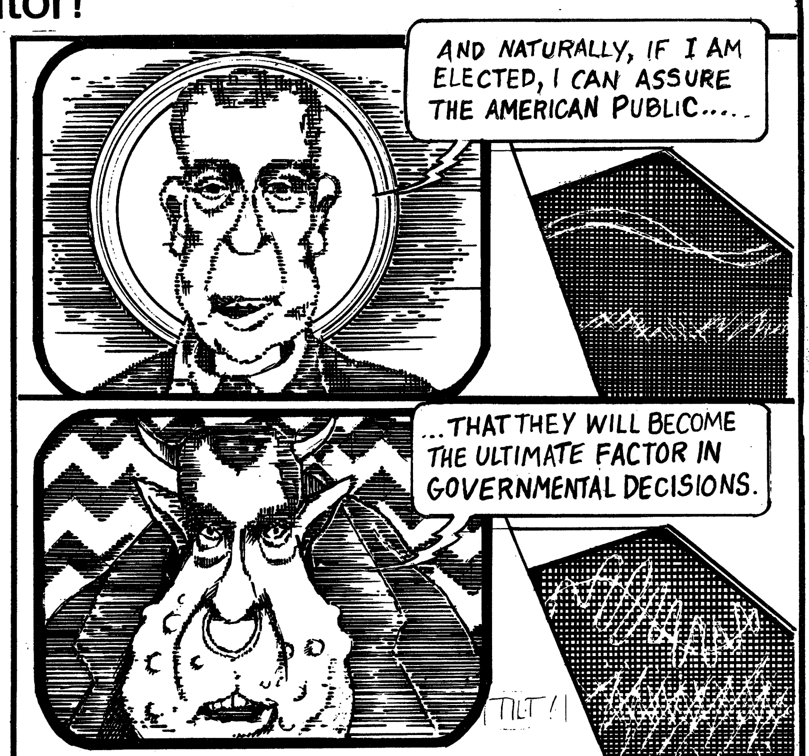
What suspense! Last issue, the massive machinations of Span-O-Vision produced its first VIS-U-COMP rendition, conjuring an unnerving scenario in which computers are nearly rejected en masse by an enraged society. only to rescue themselves with an astounding shift of focus! From the irritation of much misuse and quackery, commercial computer technology developes a pearl:



The Val-U-Graph Generator!

This device enables a home viewer to perceive when someone on T.V. is telling the truth! A special PSYCHOLOGICAL STRESS EVALUATOR UNIT reads variences in voice frequencies, determines stress, computes, and signals when a person is fibbing! Feb. 1987- Panicked legislators rush a bill through Congress prohibiting its use in legal proceedings and bans direct judgement of utterances on T.V.

Undaunted, commercial interests and T.V. industry backers devise a prime-time series in which politicians are invited to debate. Computers are programmed with all pertinent information. The debator's voices are fed into the PS Evaluator, and converted into graphic images, colors, and simultaneous electronic soundtrack based on voice frequency modulations!

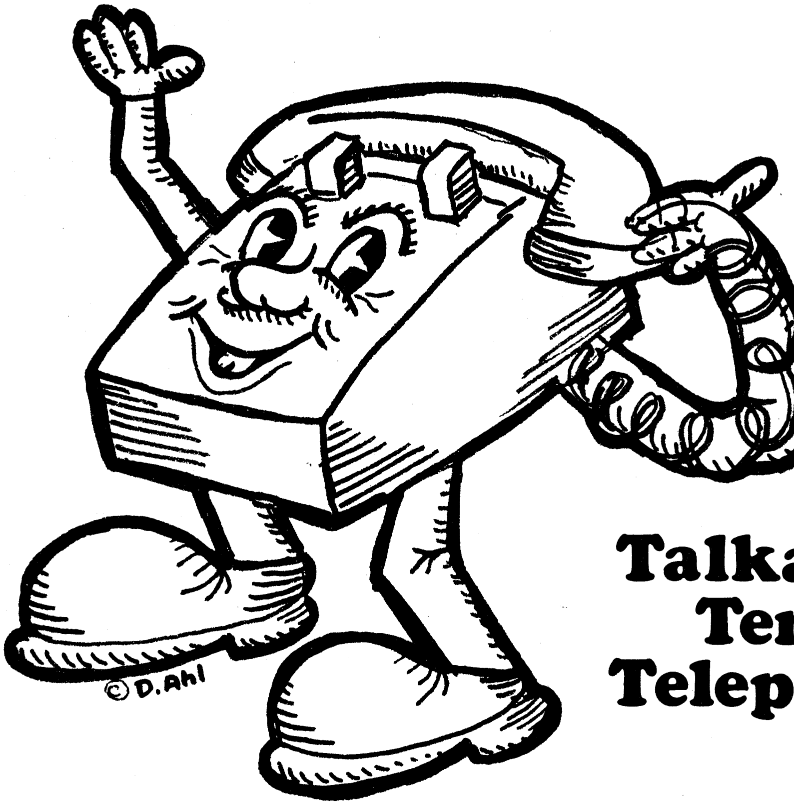


It is then the task of viewers to debate the credibility of the speakers from evidence within those images (by predetermined criteria)

Gosh! But amazingly, there's more. The home viewer may then extract claims of politician, punch into home terminal linked to huge atomic-powered ENCYCLOTRON (a SPAN-O-VISION evolvement) and receive a brief cinematic scenario immediately following said claim, along with its probability rating!

Once again the world is safe for democracy.

Resources and Related



Talkative
Terry
Telephone

COMPLEAT COMPUTER CATALOGUE

We welcome entries from readers for the "Compleat Computer Catalogue" on any item related, even distantly, to computers. Please include the name of the item, a brief evaluative description, price, and complete source data. If it is an item you obtained over one year ago, please check with the source to make sure it is still available at the quoted price.

Send contributions to "The Compleat Computer Catalogue," ***Creative Computing,*** **P.O. Box 789-M, Morristown, NJ 07960.**

BOOKS AND BOOKLETS

CALCULATORS IN THE ELEMENTARY SCHOOL

A 59 page special report published by the Curriculum Group of the Oregon Council for Computer Education. The first half of the booklet contains many calculator games and exercises. (To find out what Amelia Earhart's father said the first time he saw her flying an airplane by herself, Find .023 x 3, add 10141 to the result, multiply by 5, and look at the answer upside down.) The rest of the report describes a six-week experiment using calculators with fifth and sixth grade students including an outline of each day's activities, the results, and conclusions. Two annotated bibliographies. $2.00.

Oregon Council for Computer Education, 4015 SW Canyon Rd, Portland, OR 97221

COMPUTER USES IN EDUCATION

Proceedings of the ACM SIGGSE-SICCUE Symposium in Anaheim, California, in February 1976 are available in a 400 page publication. If you're interested in these important papers describing innovative uses of computers in education get the book. It's doubtful you'll get a chance to read them anywhere else since reprint costs run $500 an article. (Let the ACM know how you feel about that too!) Price for ACM or SIGGSE members is $15 prepaid; others $20 prepaid.

ACM Order Department, P.O. Box 12105, Church Street Station, New York 10249

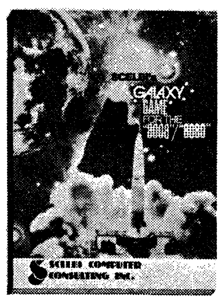
NO FREE WILL IN TOMATOES

This is the title of a little (2¾ x 4½"), handmade (sewn and tied), limited edition (300) chapbook of 14 "minimal poems" by Peter Payack. These witty poems, some of which have previously appeared in *Creative Computing* show Peter's fascination with history and science.

Peter Payack and Jane Barnes publish chapbooks under the name of Quark Press (A quark being the smallest identifiable physical particle known today). Peter is also founder of Phone-A-Poem in the Boston area (call 617-492-1144 for a delightful short poetry reading by a contemporary poet).

Three chapbooks are in Quark's First Series: *Curios* by F.A. Nettelbeck, *Mythologies* by Jane Barnes and *No Free Will in Tomatoes.* $1.00 each, $2.50 for all three.

Quark Press, Box 193, Cambridge, MA 02141



COMPUTER GAMES IN MACHINE CODE

Scelbi's First Book of Computer Games contains three games in machine code for 8008 and 8080 based systems: "Space Capture," Hexpawn, and Hangman. Hexpawn and Hangman are modeled on the familiar games, and the idea in Space Capture is to shoot down a computer-controlled spaceship roaming the galaxy. Scelbi also has another game book called Galaxy. In Galaxy the object is to search through a galaxy of 64 quadrants (each composed of 64 sectors) to find alien ships and destroy them with torpedoes or "phasors." Sound familiar? Anyway, both books include complete programs, illustrations, and flowcharts. Scelbi's First Book and Galaxy are both $14.95 apiece, ppd.

Scelbi Computer Consulting, Inc., 1322 Rear Boston Post Road, Milford, CT 06460

HP EDUCATIONAL SYSTEMS INFORMATION

Hewlett Packard, one of the most active computer vendors in the educational arena has announced a number of new applications and software packages further enhancing their hardware for educational customers.

The 12-page booklet "Computer Solutions for Elementary/Secondary Schools" describes hardware, software (instructional, CAI, administrative, etc.) for the HP 2000 and 3000 series. Free.

"Computer Solutions for Colleges and Universities" is a 12-page booklet emphasizing software in instruction, administration, and networks as well as user services. Free.

"HP Math" describes three CAI math courses for students in grades 1 to 6, the higher grades, and for adults who need remedial work in arithmetic. Free.

The 128-page "HP Clearinghouse" catalog describes almost every known application of HP minicomputers in education. It contains 80 entries on instructional and administrative applications, 50 entries on utility packages, and 90 descriptions of books, catalogs, and periodicals. $2.00.

"Learning Timesharing BASIC" is a 60-page booklet designed to teach BASIC to beginners in a light and easy-to-understand way. $3.00 [A fun text, although we feel you can do better; see *Creative's* review of 34 books on BASIC in previous issues.]

Inquiries Manager, Hewlett-Packard Co., 1501 Page Mill Road, Palo Alto, CA 94304

CALCULATORS IN THE CLASSROOM

A 24 page report on a symposium sponsored by Rockwell International in December 1974. Most of the controversy centers around whether or not the use of calculators in the classroom is justified or whether it is just a passing fad (as "language labs" once were). Price unknown.

Arnold Isford, Manager, Educational Marketing, Microelectronic Product Division, Rockwell International, 3310 Miraloma Avenue, PO Box 3669, Anaheim, CA 92803

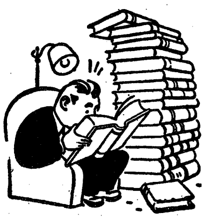
"Knowledge is power."

Bacon

COMPUTERIZED TEACHING IN THE HEALTH SCIENCES

This 200 page book is the most complete and up-to-date collection of information about computerized teaching materials in the health sciences. It contains information on 300 instructional units or programs intended for students of medicine, dentistry, nursing, pharmacy, and other health sciences disciplines. It covers about 750 hours of teaching materials, and includes key-word indexes for subject matter, author, programming language, and source institution. It is published by the Health Sciences Interest Group of ADCIS, and costs $4.50 per copy. Make checks payable to "ADCIS-HSIG."

HSIG Treasurer, 2114 Mason Hill Drive, Alexandria, Virginia 22306.



AFIPS BOOKLETS

Two booklets are available from AFIPS of interest to educators:

"AFIPS Press Publications" lists NCC Proceedings and a wide range of books from AFIPS, the 15 constituent societies, American Elsevier, and Hayden. Free.

"All you ever wanted to know about AFIPS constituent societies (but never could find in one place)" provides exactly what the title says. Free.

American Federation of Information Processing Societies, 210 Summit Ave., Montvale, NJ 07645

A.I. BIBLIOGRAPHY

A comprehensive bibliography listing 334 papers, 7 books, and 29 technical reports produced over the years at the MIT A.I. Laboratory is available. Lists extensive material on LISP, LOGO, PLANNER, robots, etc. Sources of the publications are noted. Ask for A.I. Memo No. 191, updated June 1975.

MIT Artificial Intelligence Laboratory, Room 812, 545 Technology Square, Cambridge, MA 02139

FREE MONEY?

Well, almost free. The 1975-1976 Annual Register of Grant Support is an index to more than 2,500 sources of nonrepayable financial aid from a wide variety of sources. It's 600 pages long and $47.50, so you may need a grant just to buy this book! A similar book, entitled The Directory of Publishing Opportunities, is a guide to publications which accept work from outside sources. The directory is 700 pages and $44.50.

Marquis Academic Media, 200 East Ohio Street, Chicago, Illinois 60611

MICROPROCESSORS IN EDUCATION

This is a bibliography and directory of manufacturers, articles, and other material on microprocessors, microcomputers, and small general-purpose computers. It covers hardware, software, interfacing, etc. Unfortunately, the field is moving so fast that a bibliography becomes quickly dated. Nevertheless, this booklet (dated Feb. 1976) will bring you up to date through 1975.

(I confess to having mixed feelings about this booklet. They list 36 magazines carrying material related to microprocessors in education including ones like *American Laboratory, Iron Age,* and *Machine Design* but do not list *Creative Computing.* One wonders. . . —DHA)

Oregon Council for Computer Education, 4015 S.W. Canyon Road, Portland, OR 97221

THE BEST OF ZEPHYROS

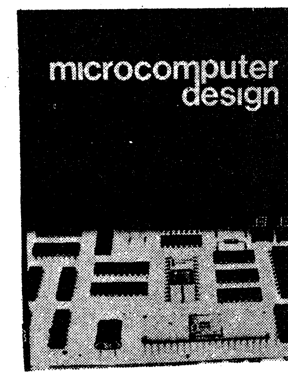
Zephyros started in 1971 as a non-profit group of teachers, parents, and artists in San Francisco with the intent to produce and print their own textbooks to improve on the sterile offerings of the major publishers. Collecting and printing ideas developed and used by classroom teachers in the past 4 years, they have reached over 46,000 teachers many of whom have contributed ideas to the 17 Deschool Primers produced by Zephyros to date. Zephyros is a rare trading post of ideas that really work. This oversize (11x16"), 308-page book is a collection of 300 of the best of those eventful, creative lesson plans. Edited by the most creative one of all, Ron Jones. $10.00. Zephyros catalog $1.00.

Zephyros, 1201 Stanyan St., San Francisco, CA 94117

MECHANICAL GARDEN

This wild 192-page book by Darrell Forney is a collection of graphics from the commonplace (old postcards, catalog illustrations) to the unexpected (schematic diagram of a hollow planet, UFO photo) to the bizarre (1890 African superstitions illustrated, sewing machine powered by a dog). Lots of fun! Printed on newsprint (like *Creative Computing*). $5.00 postpaid.

Darrell Forney, Mechanical Garden, Box 35, Bolinas, CA 94924.



MICROCOMPUTER DESIGN

A rather technical 400-page book aimed at engineers designing microprocessors into other products. Could also be used as a college text or by advanced hobbyists. Full schematics and circuit descriptions are included for the 8080-based CPU of the MIKE 3 microcomputer from Martin Research. Microcomputer Design book $25.00; literature or the MIKE 3 and Modular Micro Series free.

Martin Research, 3336 Commercial Ave., Northbrook, IL 60062

IC CATALOG

Cybertronics new catalog covers a broad spectrum of ICs and related devices (TTL DIPS, CMOS, Amplifiers, VRs, Microprocessors, etc.). Also sockets, wirewrap stuff, prototyping boards, and capacitors. Catalog free.

Cybertronics, P.O. Box 18065, Louisville, KY 40218

SUPERMARKET SCANNING AND YOU

An eight page booklet which explains the advantages of the Universal Price Code—those black and white bars which now appear on almost everything you pick up at the supermarket. As can be expected, this booklet tends to minimize or completely ignore the disadvantages of the UPC. Moderately interesting, though.

NCR Corporation, Main and K Streets, Dayton, Ohio 45479

TICCIT, PLATO IV REPORTS

Two evaluative progress/status reports about the PLATO IV and TICCIT CAI systems are available from the National Science Foundation. Some 800 PLATO IV plasma/microfiche terminals are now in use served by two host installations in Illinois and Florida. TICCIT uses a computer/user controlled color TV terminal and installations are operative in

Utah, Arizona, and Virginia.

Also available from the NSF are summary abstracts of 1975 awards granted by the Technological Innovation in Education Section. Progress reports and awards listing free.

Erik D. McWilliams, Technological Innovation in Education, Directorate for Science Education, National Science Foundation, Washington, DC 20550.

COMPUTER MANPOWER—SUPPLY AND DEMAND

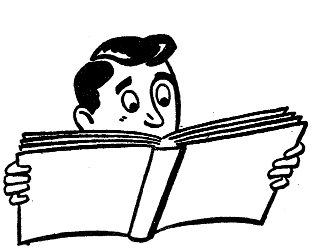
A 39 page booklet by John W. Hamblen. This publication explains what types of jobs are available in data processing and where (geographically) the jobs will be in the future. Also contains extensive statistics on where to get a post-secondary education in computer science. While this isn't exactly the kind of reading you can't put down until you finish, it is something which should be in every high school guidance department. $10 for educational institutions and $25 to everyone else.

Information Systems Consultants, R.R. 1, Box 256A, St. James, MO 65559

BRIEFING ON THE IMPACT OF PRIVACY LEGISLATION

A 65 page booklet published by the Data Processing Management Association. While this booklet is a little heavy, it is definitely worthwhile reading. It consists of commentaries and interviews of people both in and out of government. Some of the 'privacy horror stories' are so shockingly absurd, they're almost funny. (as long as they're not happening to you)! $15.95.

Data Processing Management Association, 505 Busse Highway, Park Ridge, Illinois 60068



IC, VIDEO GAME KIT CATALOG

Jade Company's new catalog has a complete listing of ICs (TTL, CMOS, Linear) as well as discrete components (transistors, diodes, etc.). [Prices are about as low as I've seen—DHA] Catalog also lists a video game kit ($137.50) that plays 5 interesting games. Catalog free.

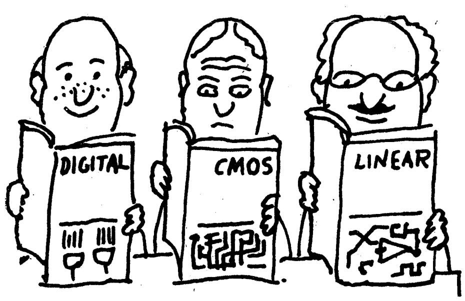
Jade Co., P.O. Box 4246, Torrance, CA 90510

MAGAZINES, JOURNALS, NEWSLETTERS

COMPUTER NOTES

Computer Notes is a substantial monthly tabloid newspaper serving the Altair Users Group plus anyone else interested in the Altair line of computer kits. It carries articles by MITS engineers and software people as well as letters from users, questions and answers, software notes, etc. Andrea Lewis, the capable editor of *Computer Notes* usually has some interesting ramblings also – a description of the World's First Computer Store or the computer fair at Lawrence Hall of Science. WARNING: if you get the newsletter, you're probably going to want a computer kit very, very badly. *Computer Notes* is $10/year, sample copy free.

MITS, 2450 Alamo SE, Albuquerque, NM 87106. (505) 243-7821.



BYTE

Byte is a new magazine edited by Carl Helmers who previously produced the *Experimenters Computer System* newsletter on building a computer from scratch. It bills itself as "the small systems journal" and basically covers the computer hobbyist field: hardware projects, using surplus equipment, tutorials, games, programming (in low-level or machine code), etc. The third issue (Nov 1975) contents are typical: ins and outs of volatile memories, chip designs, parallel output interfaces, a new ROM programmer, writing pseudo instruction sets, and the hexapawn game for machine code. For those into hardware, this is a great mag. Monthly. 96 pp per issue. $12/year, single copy $1.50.

Byte Publications, Inc., 70 Main St., Peterborough, NH 03458.

THE COMPUTER HOBBYIST

An excellent technical/hardware newsletter. Tutorial articles, construction, cassette interface, surplus parts and how to use them, an 8008 graphics system, etc. Builders from scratch (of which I'm not one) seem to like this one the best. Monthly. $6/year. Back issues (started Dec 74) or sample copy 50¢.

The Computer Hobbyist, Box 295, Cary, NC 27511.

HOBBYIST HARDWARE EXCHANGE

On-Line is a small magazine consisting of nothing but classified ads, both commercial and individual. Mostly ads for hardware hobbyist items, some club listings and meeting schedules. The publisher, D.H. Beetle, guarantees that every issue will carry at least 6 pages of ads, otherwise he'll count it only as a partial issue. Comes out every three weeks. Trial 4-issue subscription $1.00, 1-year (18 issues) $3.75.

On-Line, 24695 Santa Cruz Highway, Los Gatos, CA 95030

AMATEUR COMPUTER SOCIETY NEWSLETTER

The oldest of all the hobbyist newsletters, this one dates from "the good old days" when hobbyists built computers from scrapped 650s, G-15s, and military guidance computers. It now covers kits, chips, and reader experiences. This 6 to 8 page mimeo newsletter appears every 2 to 3 months. Editor is Steve Gray, digital and audio editor of *Popular Electronics* and frequent contributor to *Creative Computing*. $5.00 buys you membership in the Society plus 10 to 12 newsletters over a 2-year period.

Stephen B. Gray, Amateur Computer Society, 260 Noroton Ave., Darien, CT 06820

SCCS INTERFACE

A newsletter-turned-magazine written by the Southern California Computer Society. Seems to be mostly hardware oriented and largely reprints. To get Interface you must join SCCS—membership is $10/year.

Southern California Computer Society, P.O. Box 5429, Santa Monica, CA 90405.

SWTPC 6800 COMPUTER NEWSLETTER

A new newsletter for users of Southwest's M6800 system. The first issue (June 1976) contained the source code for Tiny BASIC on the M6800, and the machine code for Blackjack. The newsletter also told where to get games such as Hangman, Mastermind, and "Klingon Capture" (whatever that is) for the 6800. Printed and mailed free to all M6800 owners and clubs.

Southwest Technical Products, 219 W. Rhapsody, San Antonio, TX 78216

COMPUTE/115

This newsletter is sponsored by National Semiconductor and is produced by COMPUTE (Club of Microprocessor Programmers, Users, and Technical Experts). Since it is so highly oriented toward National and its products (half the newsletter seems to be N.S. spec sheets) it may not be too interesting to people who aren't using National microprocessors. Membership is $15 and also includes listings of user library programs. Printed monthly.

COMPUTE/115 National Semiconductor, 2900 Semiconductor Drive, Santa Clara, CA 95051

ALTERNATIVE SOURCES OF ENERGY

A.S.E. is a quarterly magazine for people concerned with the development of alternative technologies for a decentralized society. Emphasis is on alternative environmental technologies in the fields of energy sources, agriculture, transportation, and communication. It is written by people who are directly involved with these ideas. A.S.E. also functions as a communications network and maintains a lending library. Besides printing the quarterly, A.S.E. also publishes special booklets on various topics, the most recent of these being *Kilowatt Counter: A Consumer's Guide to Energy Concepts, Quantities, and Uses.* $5 for four issues.

A.S.E. Subscriptions, Route 2, Box 90A, Milaca, MN 56353

NASA FACTS

In contrast to the abounding negative publicity about the space program, this magazine explains some of its positive aspects, and also tells what NASA is involved in now. The June 1975 issue (stock number 033-000-00618-7) has some very timely stuff in it about the Viking Mission to Mars. There are lots of good color pictures and diagrams.

NASA Facts can be obtained from the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402, for 50¢.

COMMUNICATIONS TOMORROW

This newsletter is printed by the Special Studies Division of the World Future Society and edited by the talented Wes Thomas. Interesting reading; the first issue (April 1976) had short articles on such topics as satellites for people, fiber optics systems, and computer conferencing. Published bimonthly. Subscriptions as $9 per year.

World Future Society, 4916 Elmo Avenue, Washington, DC 20014

DR. DOBBS JOURNAL

The principle purpose of this xeroxed newsletter is to promote the writing and distribution of free software written for and by amateur computer users—in the form of Tiny BASIC. The first issue contained the entire machine code for Tiny BASIC on an 8080 system and information on how to interface a scientific calculator chip to an 8008. This newsletter recently eliminated "Tiny BASIC" from its name and is now titled *Dr. Dobb's Journal of Computer Calisthenics and Orthodontia.* The intention is to widen out to include other "Tiny" languages. Since many people can't afford all the newsletters out on the market today, DDJCC&O will carry reprints of useful articles from other sources.

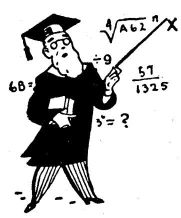
People's Computer Company, Box 310, Menlo Park, CA 94025. A year's subscription (10 issues) is $10 and single copies are $1.50

LEARNING AIDS

CARDBOARD COMPUTER

Cardiac, the Cardboard Illustrative Aid to Computation, is a cardboard device developed by Bell Laboratories which can be used to introduce students and adults to fundamental computing principles. Cardiac has 100 words of memory, a single accumulator, and 10 two-address instructions. It can be used to demonstrate loops, indexing, subroutines, double precision, and comes with a 53 page manual. $4.95.

Comspace, 350 Great Neck Rd., Farmingdale, NY 11735



THE MATH GROUP

This company produces learning aids such as puzzles, posters, and word games. The materials they offer are designed to make learning fun. Free catalog.

The Math Group, Inc., 396 East 79th Street, Minneapolis, MN 55420

LEARNING ACTIVITY PACKAGES

These three learning activity packages were developed by John Gindele, a veteran teacher at Plymouth Junior High School near Minneapolis. They are self-instructional packets designed to introduce 7th grade industrial arts students to the computer

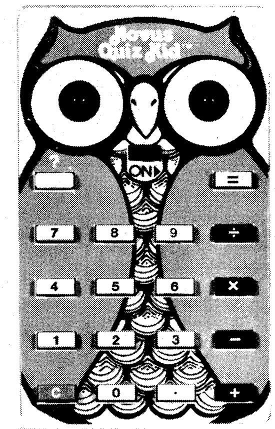
Introduction to Computers — Explains what a computer is and traces the development of computers from before Babbage's "Analytical Engine" to modern day technology. 22 pages.

Introduction to Time-sharing and the BASIC Language—This LAP discusses what timesharing and BASIC are, but doesn't get into any programming concepts. 13 pages.

Using the Computer to Figure the Cost of Your Industrial Arts Project—As Mr. Gindele suggests, this package is a little more limited to use in the industrial arts/mathematics area. While it does explain how to log on, get and run a program, and log off, the packet is oriented towards using a Honeywell 1648 with a TTY and acoustic coupler and is somewhat limited in applications with other systems and terminal types. 18 pages.

These LAP's are $3.50 each or $7.75 for all three and some additional material. Send for a free catalog describing these and other LAPS to EDU-PAC of Minnesota, Box 27101, Dept. 76, Minneapolis, Minn. 55427

(A more complete review of this will appear in the future in *Creative Computing.)*



QUIZ KID CALCULATOR

Here's a new kind of calculator. There's no place for an answer to appear! Instead, a youngster enters a problem such as "9 x 2 = ?" and then punches in the answer s/he believes to be correct. If 18 is entered to the problem above, the owl's green eye will light; any other answer gets a red eye. Watch your kid's eyes light up with this one! It really makes dull arithmetic an enjoyable game at a very economical price. Complete with coloring book chock full of elementary arithmetic puzzles and carrying pouch. $17.95 plus $2.50 shipping.

National Semiconductor Electronics, 1177 Kern Ave., Sunnyvale, CA 94086

GLOBAL FUTURES GAME

A dynamic simulation of present and future world conditions in terms of population, food technology, education, and relative growth rates of each. Groups of players representing eight socio-economic world regions barter for resources in five year rounds ending in the year 2020. Players make collective policy decisions and errors are reflected in "World Destruct Points." Hopefully the players will gain insight into the interconnectedness of world problems and their solution. The game can be played by 8 to 48 high school or college students in two hours. The game is $10 and extra scoresheets are $5 for 48.

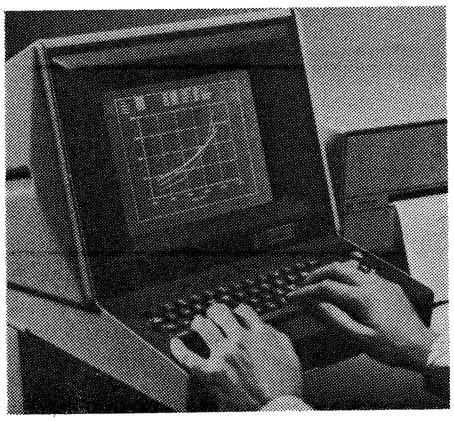
Earthrise, Box 120, Annex Station, Providence, RI 02901

MATHIPUTER

A "Mathiputer" is a rugged calculator type device which can be used in classrooms to teach addition, subtraction, multiplication, and division. However, instead of supplying the answer, the student is asked a problem which he must solve. A correct answer will cause a "happy face" to appear, and a "sad face" indicates an incorrect response. Prices range from $199.50 to $324.50 for one which times responses.

Cybernetic Systems, Inc., 9615 Acoma Southeast, Albuquerque, NM 97123

HARDWARE



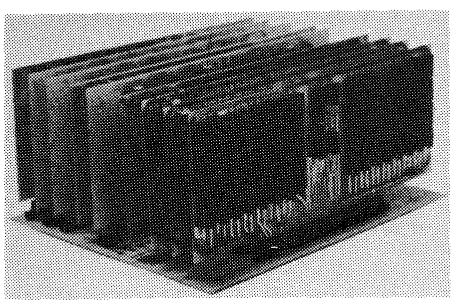
TEKTRONIX GRAPHICS PRODUCTS

Following close on the heels of interactive computing in schools is interactive graphics. And few vendors have a more comprehensive line of graphics products than Tektronix ranging from the inexpensive 4006 graphic computer terminal ($2995) to the 4014 big 19″ screen terminal ($10595) to the 4051 stand-alone BASIC graphic computer ($6995).

Worthwhile free literature from Tektronix includes Tekgraphics (April 73, No. 5, "Educational Applications"), 4051 Flyer, Computer Products Catalog, and Price List.

Tektronix, Inc., Information Display Group, P.O. Box 500, Beaverton, OR 97077

[See the complete review of the 4051 on pp. 20-21.]



THE DIGITAL GROUP

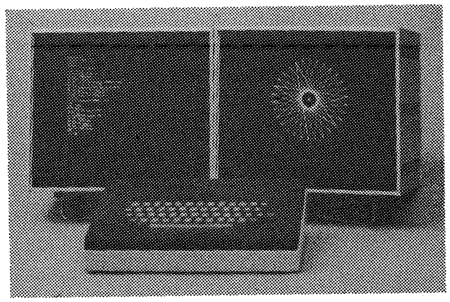
The Digital Group has a new idea. Namely a microcomputer system with interchangable CPUs at the CPU card level. In other words, a system doesn't become immediately obsolete with each new microprocessor announcement. Your major investment in memory and I/O is protected.

A typical Digital Group System consists of a CPU card with 2K RAM (choice of Zilog/Mostek Z-80, Intel 8080A, Motorola 6800 or MOS Technology 6502), I/O card with four input and four output ports, TV Readout and audio cassette interface, 8k static RAM and Mini-Mother board. You have to add a power supply, ASCII keyboard, cassette recorder, TV set and cabinet (if you wish) and you're ready to go.

Currently available software includes Tiny BASIC, a number of Tiny BASIC games, some ham and educational packages. Kits with 2k memory (no power supply) start at $375 (6502), $425 (8080,6800), or $475 (Z-80). Add $135 for 12A power supply, $225 for 8k memory and keyboard (around $50) and you're in business. More information is free.

[Incidentally, The Digital Group has gained a good reputation for not announcing products before they can be delivered — DHA]

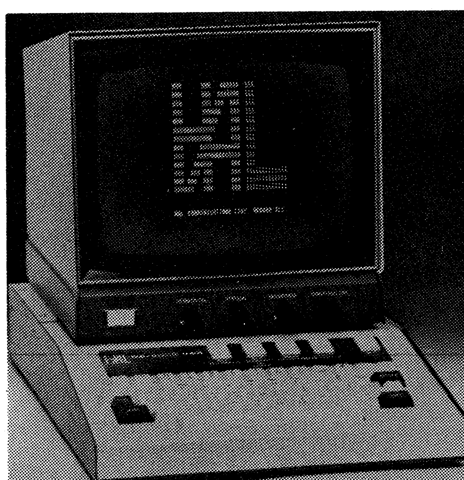
The Digital Group, Inc., P.O. Box 6528, Denver, CO 80206



TT250 GRAPHICS TERMINAL

Developed at the MIT Artificial Intelligence Lab, this terminal is different from any other you've probably seen since it uses two screens! One is used for text, and the other for *dynamic graphics*. The characters are on programmable 16x8 fonts with no spaces between the characters—so you can combine the characters to form larger symbols or even diagrams. In normal use the separation between characters is treated as part of the character itself. Also, the graphics have the capability of animation; that is, they can be changed smoothly and rapidly. The TT2500 has its own built-in processor and starts at $5,950.

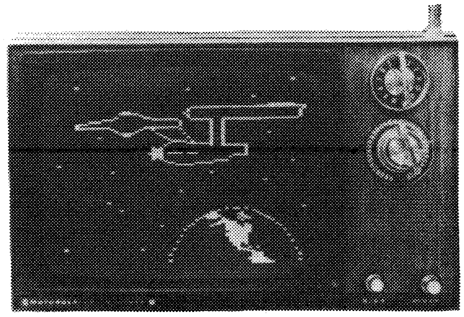
General Turtle Corporation Limited, 120 Boul. Industriel, Boucherville, Quebec J4B 2X2, Canada



CRT TERMINALS

DS-3000 KSR terminal from Hal Communications can receive and send Baudot (5 level) and/or ASCII (8 level) code data. Built-in microprocessor allows full cursor positioning and editing of the full screen of 16 lines x 72 characters/line. Hal seems more interested than most manufacturers in schools and hobbyists.

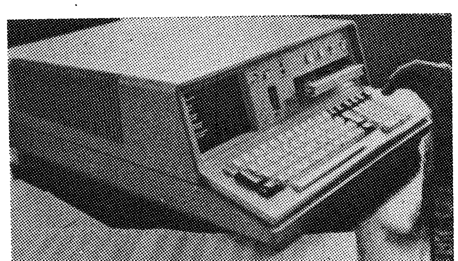
HAL Communications Corp., 807 East Green St., Urbana, IL 61801



LOW-COST GRAPHICS TERMINAL KIT

The GT-61 Graphics Terminal Kit, produced by Southwest Technical Iroducts Corporation, displays an array of cells 64 wide by 96 high on a standard video monitor or modified TV set. Each cell can be selectively turned on or off by the computer. There is also a provision for mixing graphics and alphanumerics on the same screen when using the GT-61 with Southwest's CT-1024 TVT. And since the GT-61 may be driven by any computer having a TTL compatible 8 bit parallel output, it isn't necessary to own a particular system to use it! The GT-61 is $98.50, less power supply, chassis, and TV monitor.

Southwest Technical Products Corporation, 219 W. Rhapsody, San Antonio, TX 78216



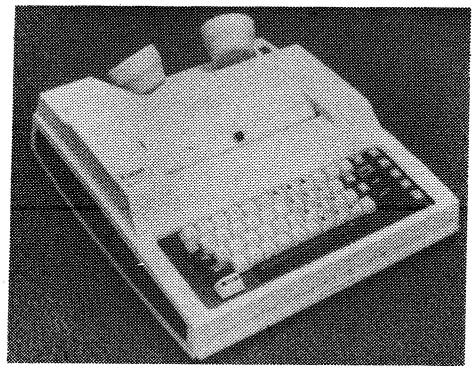
IBM 5100

The 5100 is IBM's first entry in the true minicomputer/calculator field. This desktop unit, slightly larger than an electric typewriter, contains CPU, keyboard, 4″ CRT display, cartridge tape unit, memory (up to 64k) and can run BASIC, APL, or both. The 5100 can drive a TV monitor directly. Options include a telecommunications interface, medium-speed printer, and an auxiliary tape unit. (BASIC takes 4k overheard, APL 6k). Prices are as follows:

Memory	BASIC	APL	Both
16k	$ 8,975	$ 9,975	$10,975
32k	11,975	12,975	13,975
48k	14,975	15,975	16,975
64k	17,975	18,975	19,975

Printer is $300, aux tape $2300, comm. adapter $900, programming packages (business, stat, math) $500 each, BASIC CAI pkg $225, APL CAI pkg $295.

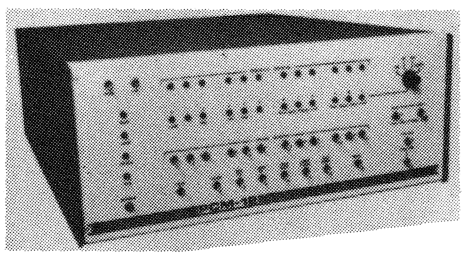
IBM, General Systems Div., P.O. Box 2150, Atlanta, GA 30301



LIGHTWEIGHT TERMINAL

Weighing only 13½ lbs including coupler and carrying case, the Tymshare Model 125 terminal requires only a standard telephone and electrical outlet for fully portable KSR operation. Features include silent printing, color-coded keyboard, integrated 13-key numeric pad, and alternate 30 or 10 cps operation. Sale price: $2210; rental as low as $100/month. The Model 125 is available through Tymshare offices in major cities in the United States.

Tymshare, Inc., 20705 Valley Green Drive, Cupertino, CA 95014



PCM-12 COMPUTER KIT

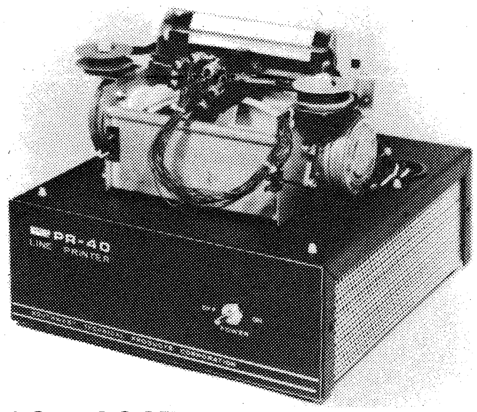
If you're thinking about buying a computer kit, then this one is worth some consideration. The PCM-12 is a simulation of a PDP-8 based on the Intersil IM6100 microprocessor. It runs the same software and has essentially the same front panel functions as the PDP-8. As a result there is already a ton of cheap software for the PCM-12. When ordered with the kit, 4K BASIC on papertape is only $1.50! The basic system with 1K of memory is $799. An 8K system with terminal interface (but no terminal) would run $1417.

PCM, Box 215, San Ramon, CA 94583

TALKING CALCULATOR

With a 24-word vocabulary, the SPEECH-PLUS calculator enunciates—in easy-to-understand electronic speech—each keystroke as it is made and then announces the answer. In production applications, it is useful to verify entries without watching the visual display and for education it is valuable to capture and hold the attention of students. Four functions, square root, percent, independent memory, constant, floating decimal, eight-digit capacity. Earphone jack for private listening; operates 3 hours between charges. Instructions available in print, braille, or cassette. Complete with case, earphone, battery charger and instructions $395.00.

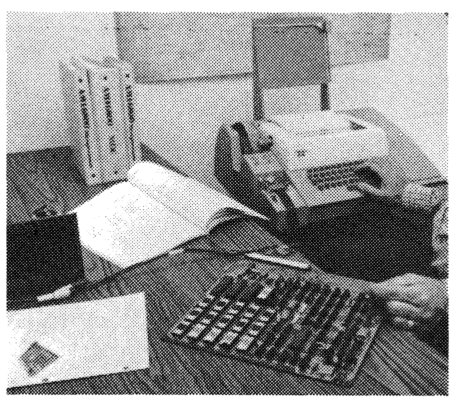
Telesensory Systems, Inc., 1889 Page Mill Rd., Palo Alto, CA 94394



LOW COST PRINTER KIT

Southwest Technical's PR-40 Alphanumeric Printer Kit uses a 5x7 dot matrix impact print mechanism. It prints the 64 upper case ASCII characters, 40 characters per line, 75 lpm on standard 3⅞″ rolls of adding machine paper. Complete kit $250.00.

Southwest Technical Products Corp., 219 W. Rhapsody, San Antonio, TX 78216



PROTOTYPING BOARD FOR AMI 6800

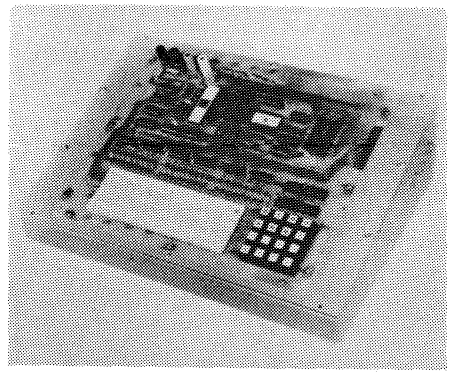
This board is useful for hardware and softward evaulation of 6800-based microcomputer applications. It contains a built-in programmer for the S6834 EPROM and Tiny BASIC is also available on the EPROM at no extra charge. Communication to the outside world is done with a TTY. A minimum kit is $295 and a fully tested unit with Tiny BASIC on EPROM is $950.

American Microsystems, Inc., 3800 Homestead Rd., Santa Clara, CA 95051

JOLT

JOLT is a microcomputer system based on the 6502 microprocessor with a debugger/monitor on a special read-only memory package called DEMON. This firmware permits use of any terminal from 10-30 cps and lets you display CPU register, memory locations, also load from and write to papertape. JOLT is not a mainframe, it's one of those things where the boards stack one on top of another. The CPU board is $249 and other options are available.

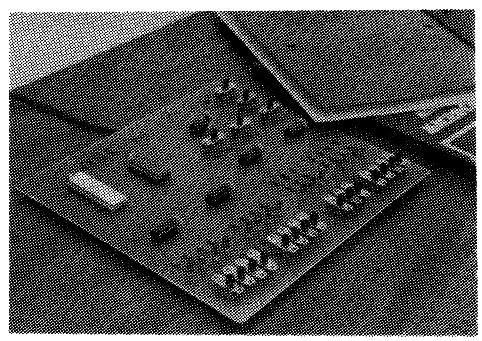
Pehaco Corporation, Jolt Sales Agents, Microcomputer Associates Inc., Dept. A, 111 Main Street, Los Altos, CA 94022



MINI-MICRO DESIGNER

The MMD-1 is an inexpensive educational microcomputer with easy breadboarding capabilities. It has a keyboard entry system and is intended for use at the machine language level and not for sophisticated applications. The MMD-1 is based on the 8080A chip and has on-board PROMs to control keyboard entry. A complete MMD-1 kit is $350. Bugbook III provides extensive documentation including 60 experiments for the MMD-1 in its 592 pages. $14.95.

E & L Instruments Incorporated, 61 First Street, Derby, Connecticut 06418



COMPUTER TRAINER

The Model 300 Computer Trainer is a completely assembled and tested, ready-to-use computer which comes complete with a 20 experiment lab manual written for use with a college physics, electronics, or computer course. Since the manual assumes no knowledge of computers or digital electronics, it is also ideal for self-teaching. The first lab is simply loading and reading memory, and the last experiment is interfacing the computer to a TTY.

The unit comes with 128 words of memory and is based on the MOS Technology 6502 chip. The Trainer with lab manual is $99, power supply is $10, and hardware and programming manuals are $10.

Ohio Scientific Instruments, Box 374, Hudson, Ohio 44236

ORGANIZATIONS

SCS

The Society for Computer Simulation is the principle technical society devoted to the advancement of simulation through the use of computers and similar devices. SCS seeks to promote the development of simulation technology through the exchange of information among people who use simulation to advantage in their endeavors. Full membership is open to anyone who has been in any phase of simulation for at least four years. Membership is $25/yr, students $15.

Alex McKenna, The Society for Computer Simulation, P.O. Box 2228, La Jolla, CA 92038

ACM

The purpose of the Association for Computing Machinery is to advance the sciences and arts of information processing including the study, design, development, construction, and application of modern machinery, computing techniques, and languages. Persons qualified to be ACM members must subscribe to its purposes, have attained professional stature as demonstrated by intellectual competence and ethical conduct in information processing, and must be endorsed by two members of the ACM. Student membership also available at reduced rates. Many diverse publications and Special Interest Groups.

Joseph Cunningham, Association for Computing Machinery, 1133 Avenue of the Americas, New York, N.Y. 10036

ACM SIGCSE

The purpose of the ACM Special Interest Group on Computer Science Education is to provide a forum for problems common among college educators attempting to develop, implement, or evaluate computer science programs, courses, and problem sets. Quarterly newsletter. Non-ACM membership in SIGCSE $9.00/yr.

SIGCSE, ACM, 1133 Ave of the Americas, New York, NY 10036

ACM SIGCUE

The Special Interest Group of Computer Users in Education of the ACM has as a major purpose the interchange of information among educational (instructional) users. The membership of 1200 or so is skewed toward higher education. The quarterly bulletin is substantial (40 typed pages) and carries in-depth articles, interviews, and conference reports. Each issue also runs an annotated bibliography of new books and magazine articles. The group sponsors meetings at the annual ACM, NCC, and CCUC conferences, and a joint conference with SIGCSE (first time in 1976). Annual membership $6 for non-ACM members.

SIGCUE, ACM, 1133 Avenue of the Americas, New York, NY 10036

ACM SIGCAS

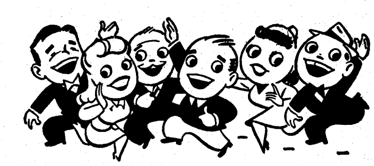
The purpose of the ACM Special Interest Group on Computers and Society is to provide a forum for the examination of the impact of computers on society, in terms of major economic, political, and sociological trends; information systems in many areas; privacy; and related issues. Very professional quarterly newsletter. Non-ACM membership in SIGCAS $9.00/yr.

SIGCAS, ACM, 1133 Ave. of the Americas, New York, NY 10036

IEEE COMPUTER SOCIETY

The IEEE Computer Society is actually part of a much larger organization, the Institute of Electrical and Electronics Engineers. It was formed to advance the theory and practice of computer and information processing technology and exchange technical information among its members. These people are concerned mostly with hardware. To be eligible for membership in the IEEE Computer Society you must be an IEEE or approved society member, have graduated from a four-year course of study, have been involved in the computer field professionally for at least five years, or be a registered student in the Society's field of interest Membership including IEEE dues $46/yr.

Harry Hayman, P.O. Box 639A, Silver Spring, Maryland 20901



ASCUE

This organization of Small College Users in Education aspires to encourage the appropriate uses of computing equipment and techniques for its member institutions and to assist its members in solving individual problems. Formerly an IBM 1130 users group, member institutions today have DEC, HP, GA, and other hardware as well as the inevitable IBM systems. Regular Membership is $35 for educational institutions.

Dan Kinnard, ASCUE Director of Public Relations, New Mexico Military Institute, Roswell, New Mexico 88201

COMNET

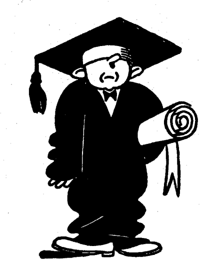
ComNet—Northeast Information Network is a cooperative resource and skills exchange for Washington, Oregon, Idaho, and British Columbia. It serves as a community memory bank providing people with information on positive options and alternatives they need for now and the future. ComNet makes information available on topics such as alternative sources and uses of energy, communications, food & nutrition, tools and low impact technology, and transportation. Individual membership for a year is $6.00 and includes 4 issues of the magazine Northwest Synergy Access. Include a SASE.

ComNet Northwest Information Network, Box 5599, Seattle, Washington, 98105

ADCIS

The Association for Computer-Based Instructional Systems focuses on CAI in general, with particular emphasis on educational psychology and applications in the medical sciences. It publishes a newsletter which carries information about the activities of individual members and abstracts of the association's two meetings each year.

Peter Dean, Box 1403, Los Gatos, CA 95030



AEDS

The Association for Educational Data Systems was founded in 1962 to provide a forum for the exchange of ideas and information about the relationship between technology and modern education. Primary emphasis tends to be on public school data processing applications with instructional applications taking a back seat. Individual membership $20, student $10. Membership is "open to all interested in learning more and keeping informed about current developments in educational data systems." AEDS also holds an annual convention and sponsors a programming contest for students in grades 7-12.

Shirley Easterwood, Association for Educational Data Systems, 1201 16th St. N.W., Washington, D.C. 20036

NAUCAL

The National Association of Users of Computer Applications to Learning is an organization of users whose purpose is to improve the learning-teaching process by influencing the development, evaluation, and dissemination of computer applications to learning. NAUCAL provides a forum for the exchange of ideas in the areas of drill-and-practice, tutorial, simulations, problem solving, computer-based testing, and other facets of instructional computing. It was founded in 1970 and is now a functional chapter of AEDS. Membership is on an individual basis at $5.00 per calendar year.

Mr. George H. Litman, Secretary/Treasurer, Board of Education, City of Chicago, 228 North Lasalle Street, Room 430, Chicago, Illinois 60601

ASIS

The American Society for Information Science is a nonprofit professional association organized for literary, scientific, and educational purposes and dedicated to the creation, application, and dissemination of knowledge concerning information and its transfer. The Society acts as a bridge between research and development and the requirements of diverse types of information systems. Annual conference with proceedings, bimonthly Journal of the ASIS, 10 times/yr Bulletin. Regular membership $35, student $10.

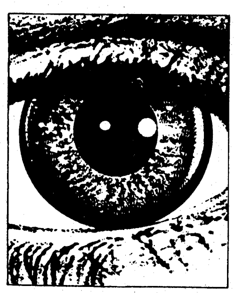
Robert McAfee, Jr., American Society for Information Science, 1155 16th Street, N.W., Suite 210, Washington, D.C. 20036

MEDIA

COMPUTER ART SLIDES

The Computer Arts Society has collected a set of 35mm slides of artwork done by computer. Sets of approximately 100 slides are available at cost (approx. 30¢ per slide, i.e. $30.00/set).

Ruth Leavitt, 5315 Dupont Ave South, Minneapolis, Minnesota 55419.



EYES FOR COMPUTERS

Robotics and machine intelligence are explored in this film from the General Motors Film Library. Shown are attempts to develop decision making powers into the computer—decisions based on what the computer "sees" via TV cameras. GM's object is to develop machines to do some assembly line work, inspection procedures, and "intelligent" parts handling. The problems associated with a simple task such as recognizing a three dimensional object and then doing something with it are immense, as suggested by this film. This is an interesting computer applications film. It's short, crisp, and good! About 10 minutes. Free.

General Motors Film Library, Detroit, Mich. 48202

VIDEO TAPE ON MAGIC SQUARES

A 30 minute color video tape describing the magic squares of De La Loubere and Franklin has been produced by Donald T. Piele at the University of Wisconsin-Parkside. The tape contains: a brief historical description of the De La Loubere and Franklin Magic Squares; an explanation of the algorithm used to construct each square; and a segment showing the magic squares being generated by computer. The tape is appropriate for junior high school and above, and is available in many forms: ½" reel - $25, ¾" reel or cassette - $30, 1" reel - $30. If you send your own reel or cassette deduct $15.

David Campbell, Director, Media Production Center, University of Wisconsin - Parkside, Kenosha, Wisconsin 53140

FATHER OF THE COMPUTER

"Charles Babbage, Father of the Computer" is a 28-minute color TV documentary which introduces Babbage's amazing calculating engines and many basic concepts of computer technology. It is available in the four popular TV tape formats (videocassette, IVC 1", EIJ ½", and broadcast standard 2"). Loan of tape, free; dubbing at reasonable rates.

AFIPS History of Computing Project, 20 Wilson Road, West Point, NY 10996.

CALCULATORS

POCKET CALCULATOR BUYER'S GUIDE

A new 32-page brochure, "Pocket calculator Buyer's Guide," is now available free from the Hewlett-Packard Company. The brochure describes and gives specifications for the company's full line of preprogrammed and programmable pocket calculators for science, engineering, business, finance and education from the new HP21 to the original HP65. Guide is free.

Inquiries Manager, Hewlett-Packard Company, 1501 Page Mill Road, Palo Alto, CA 94304

BIORHYTHM CALCULATOR

If you're one of those biorhythm freaks then you'll be interested in a new Casio calculator called a Biolator, which in addition to performing regular mathematical functions, also computes your biorhythm. It has a 99 year calendar which permits you to calculate the number of days since your birth (if it was in this century—tough luck, senior citizens). The biorhythm is supposedly a representation of a person's physical, emotional, and intellectual states. Suggested retail price is $29.95. Look in a local department store because we don't have Casio's address.

ASKING THE CALCULATED QUESTION

NOVUS presents a calculator for the older siblings of the young kids who use that wise old owl QuizKid calculator described in the Sept/Oct catalogue. It's the more serious, more versatile QuizKid II. This lets the user choose a type of problem (addition, multiplication, subtraction or division) and a desired speed, then presents a timed series of ten problems. The child gets two tries at the correct answer after which it is displayed. The score is revealed at the end of the series. With a game adapter and over 1200 problems it's a hoot of a value at $24.95.

To order and for information on QuizKid II and other exciting calculator and mathematic learning aids contact: Mr. Jay Hemming, Educational Marketing Manager, National Semiconductor, 1177 Kern Avenue, Sunnyvale, CA 94086.

GAMES AND SIMULATIONS

SERIOUS COMPUTER GAMES

Eight serious computer games are available on paper tapes from ERC. They are: Planet Management, Mouse in the Maze, Stretching Springs, Oceanography, Friction Force, Paramecium Population, Photosynthesis, and Reaction of Magnesium in Hydrochloric Acid. $8.00 each.

Elaine G. King, Education Research Council of America, Rockefeller Building, Cleveland, OH 44113.

GAMES FOR CLASSROOM EARTH

The idea game, fence game, freedom games, ethics game, gasoline game, an outdoor community games tournament, yes and even some computer games (teaser, king, and lunar landing) and many others are all in Zephyros Primer #14. Ron Jones, editor of this fabulous series of 15 primers, adds 3 or 4 new ones to the collection every year. Great graphics, intriguing new ideas. Primer #14 $2.50, others $1.50 to $4.50.

Ron Jones, Zephyros, 1201 Stanyan St., San Francisco, CA 94117.

MISC.

TECHNOLOGY EXCHANGE SERVICE

Technotec is a system available from over 5000 computer terminals in more than 150 metropolitan areas worldwide. Its purpose is to get together those who have useful inventions or know-how with those seeking to acquire new technology, in the form of a "Techno-Stock." Techno-Stocks in a particular area of interest can be located easily by entering keywords. Technotec isn't an information system, but is a communication system built around a central data base. A general information manual is $3.50 (Order No. 76073300).

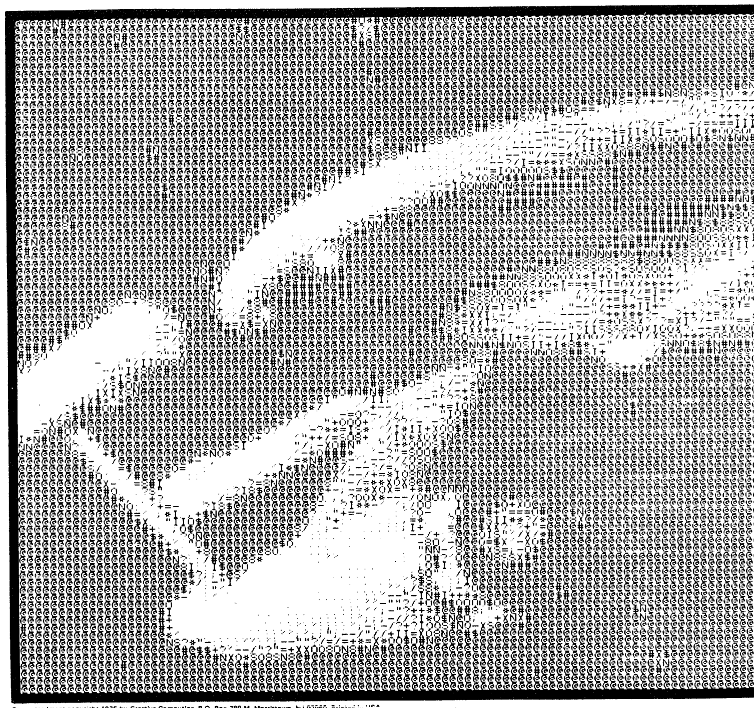
Control Data Technotec, Inc., Box O, Minneapolis, MN 55440

PROGRESSIVE TECHNOLOGY

This group publishes a list of periodicals for "Progressive Scientists" (read "Radical"). Covers a wide range of magazines and newsletters Free.

Progressive Technology, P.O. Box 20049, Tallahasee, Florida 32304

STAR TREK LIVES!



6000 "Trekkies" paid $20 each the weekend of February 13, 1976, to attend the latest Star Trek Convention, a five-day extravaganza held in the Hotel Commodore. Star Trek — whose *Starfleet Technical Manual* and *Blueprints* have captured top billing on the *New York Times* paperback lists for 15 weeks straight — is the creation of writer Gene Roddenberry. Roddenberry, whose latest obsession is a film version of Star Trek for Paramount Pictures, has seen his budget go from $186,000 for a 60-minute TV show to over $5 million for a 90-minute movie. Roddenberry is now dreaming of a new bridge where the rails are constructed of some marvellous new plastic instead of pine 2 x 4's. He also wants the Enterprise crew to have computer displays and read out devices rather than phony mockups. The movie will start shooting the fall of 1976; release is scheduled for late summer 1977.

The original Star Trek was made in the early 60's and what was then 10 years ahead of its time is now passe. Especially "the role of women" notes Roddenberry. In the original pilot, the crew was 50% male, 50% female. A woman was second in command. But that was the 60's and the executives with whom he worked were afraid that people would think that there was "too much fooling around out in space" so the ratio was reduced to 1:3.

Gene Roddenberry's interests are far broader than science fiction. He reads widely and quotes freely from the classics and the latest TM books as well as from scifi. He feels that all people should broaden themselves to a wide range of literature, instead of keeping "blinders" on for just one thing.

Roddenberry and his wife were glowing with excitement at the convention. When asked what makes Star Trek still live 10 years later, Mrs. Roddenberry said, "fans make it live. They make Star Trek real." Gene was a bit more philosophical and felt that the most important staying element in the Star Trek series is that it is "essentially a statement of optimism about tomorrow. And we don't have many of those today." Roddenberry is fundamentally a positive person; he feels that "a mind is a wonderful thing. It's a *great* computer."

The 5-day convention was a hectic happening. There were autograph and photograph sessions, presentations by various stars and scifi authors (Isaac Asimov, Frederick Pohl, Gordon Dickson, etc.), an alien costume parade, an art show (with some very professional acrylics), a fanzine and book exhibit, and a huge dealer's room with 120 dealer tables.

Ah yes, the dealer area. Frantic buying and selling of posters, books, film clips, photos, buttons, Mr. Spock ears, old and new comics, fanzines, balloons, T-Shirts, costumes, phasers, records, tapes, et al. 10-minute specials on the loudspeaker. Wild mobs. Candice Bergen walked around the dealer area completely unnoticed, followed shortly by an NBC shooting crew. Little notice even then. Perhaps if she had Vulcan ears .....

At the *Creative Computing* table (yes, we go to all kinds of cons), Trekkies compared notes on different computer versions of Star Trek. Some versions had Romulans, others had wild space warp rules, one even had Tribbles! Most found Super Star Trek (*Creative*, Vol. 1, No. 4) to their liking.

One Trekkie came by and felt compelled to explain, while paying for his Mr. Spock computer image, that he was actually a Trekker (a rational fan). Whereas, he said, a Trekkie worships anything connected with Star Trek and would sell his or her mother for a pair of Spock ears. Oh, and if you're reading this article in secret and you slip into the bathroom with your portable TV to watch Star Trek because you're afraid of what your family or friends might say, there's now a button for you too . . . "Closet Trekkie." (These buttons were sported by several well-dressed *very* well-known people at the press reception.)

There are over 25 Star Trek conventions every year, each one attended by 1,000 to 15,000 and more fans. (At *Creative* we'll attend one or two of these cons per year). There are also over 250 local Star Trek clubs so if you're a Trekkie, Trekker, or Closet Trekkie there's lots for you to do right here on earth exploring new worlds and following in the path blazed by the Enterprise crew, boldly going where no man has gone before. Live long and prosper! —DHA

STAR TREK INFORMATION EXCHANGE

In this column, *Creative Computing* will list computer-related Star Trek materials such as programs, computer images, etc. Upper limit on material for sale is $20.00; people with higher-priced material should purchase an ad.

Original Star Trek Game in BASIC

One of the first Star Trek computer games in BASIC to appear widely was actually mislabeled SPACWR (Spacewar) in *101 BASIC Computer Games.* It was designed for BASIC-PLUS on RSTS-11 but can be easily modified for other versions of BASIC. *101 Games* is available for $7.50 plus 75¢ postage from *Creative Computing,* P.O. Box 789-M, Morristown, NJ 07960.

CDC Fortran Star Trek

For Control Data 6000 series or any Cyber series running under either NOS or KRONOS. Will run on other systems with slight modifications. The program is called "STARK" and is written in FORTRAN with some simple CDC assembler ... fully documented. Will send copy on paper for $2.00 to cover postage and handling or FREE in exchange for any other Star Trek program. Includes 50 page Information-Instruction Manual. (New Version 3.0.0 available October 1976.) The version has 27 commands. Send to: C.D. Foley, F502 Wilkeson Quad, SUNYAB-Amherst Campus, Buffalo, New York 14261.

BASIC Super Star Trek on Cards for Nova

I work for an engineering-research organization that has a Data General 840 NOVA computer. Running on the NOVA at the present time is the Super Star Trek game that appeared in the May-June, 1975 issue of *Creative Computing.* Our physical copy is in BASIC punched on cards. We would be happy to send a copy of it to whomever might desire it for a break-even price of the cards, postage and duplication, of $10.

The mailing address is: Mr. Michael Tomayko, c/o Kaman AviDyne, 83 Second Avenue, NW Industrial Park, Burlington, Massachusetts 01803

Enterprise T-Shirt

A beautiful midnight blue T-Shirt with a large detailed Enterprise and stars design in silver is available from *Creative Computing.* No wording of any kind appears on the shirt. $4.00 postpaid from *Creative Computing,* P.O. Box 789-M, Morristown, NJ 07960.

APL Star Trek Listing

I have a Star Trek computer game written in APL. It's feature is that it can be set for various levels of difficulty, allowing the enemy to take evasive action and generally be rather hard to kill.

I am working on a 3-D version, but only in my spare time, and it will probably not be ready for quite awhile.

I am willing to 'give' the programs away, but I need to ask $1 to cover my effort and the postage for the listings.

Send to David B. Wood, 5108 Viking Rd., Bethesda, MD 20014.

Battle Simulation Games

I have 2 Star Trek Battle Games, a two dimensional, and a three dimensional. Each is about 400 lines of standard BASIC. They are completely random battle simulators. I will sell them for $15.00 each (paper tape extra). The price includes full source listing; documentation, and instructions and hints. For more info send stamp to Mike Aurelius, 1318 18th Avenue South, St. Cloud, Minnesota 56301.

PL/I Star Trek Listing

"I have a *Star Trek* game which runs in PL/I (CPS or other conversational system) on IBM 360/65. For a copy of the program (which runs about 25-30 8½ x 11 pages) please write to: John Braue, 407 Sherman House, University of Connecticut, Storrs, Ct. 06268. Please send $5; "U. Conn. is incredibly nasty about photoreproduction."

Advanced Star Trek in PL/I

This PL/I games is 1600 statements long (100K compiled) for an IBM 370/158 running OS/VS with TSO option. Enemy aliens consist of Klingons in two types of ship with different capabilities; Romulans which are invisible, move about and trail the Enterprise; and Tholians which attack in groups of three with weblike tractor beams. 15 commands. Main Federation ship is the Enterprise; the Fairy Queen is also available in emergencies. Galaxy is 10x10 with 10x10 sectors. Time travel is possible but very dangerous.

Game will be sent in source form with all JCL necessary. Also on cards will be included the rules and sideview picture of the Enterprise. Cost for the entire box of 2000 cards plus postage is $10.00. Also can be copied onto your magtape, but write first. Might also consider swaps; make an offer!

John R. Bane, P.O. Box 3125, University Station, Clemson, SC 29631.

Computer Images of Star Trek People

A *new* set of 7 extensively detailed computer images of Star Trek people is now available from *Creative Computing.* (The new images have solid black borders — the old ones had no borders). Kirk, Spock, McCoy, Scott, Uhura, Sulu and the Enterprise. 8½ x 11, heavy stock. $1.50 per set postpaid. *Creative Computing,* P.O. Box 789-M, Morristown, NJ 07960.

Klugy APL Star Trek Game

Bob Leedom's Super Star Trek* (*Creative,* Vol 1, No 4) is about the best I've run across, and plays very well. I have an extremely klugy version in APL (a somewhat restricted subset of Super Star Trek) which I'd be glad to trade with anyone who is interested. It does work and the local college has been using it heavily. Write Bob Wier, Recreation Technology, P.O. Box 9209, College Station, TX 77840

*Super Star Trek is now available only in "The Best of Creative Computing - Vol. 1" — $8.95 plus 75¢ postage from us at *Creative.*

Star Trek for Altair 8800

A version of Star Trek is now available for the Altair 8800. It is written in Altair 4 K BASIC and is available from International Data Systems Inc. The purchase price of $10 (checks OK) includes a complete program source listing, operational instructions, tips on how to "patch" the program to add your own features, a one year limited warranty against "bugs," and postage and handling. Star Trek Offer, International Data Systems Inc, PO Box 593001-AMF, Miami FL 33159.

Dice Star Trek

Star Trek as a computer game dates back to the late 1960s. Versions have been widely published in the HP User Library, 101 BASIC computer games, and *Creative Computing* (May-Jun 1975). It is probably the single most popular and most addictive computer game. Now, good news for those with no computer: Dice Star Trek (DST). This game is a rare example of a board (or manual) game being modeled on a computer game. And what a model it is! DST simulates virtually every aspect of the computer version: navigation, short-range scan, long-range scan, phasers, photon torpedoes, shields, damage control, etc. Be warned: it's complicated—the instructions run to 34 pages (!) and there are 12 detailed "cards" which supply galactic sector conditions, phaser hit information, on-board computer calculations, damage information, etc.

Marion Stubbs, author of DST, writes that games of DST "last for days or weeks. Perhaps months if one takes time off in between. Two hours vanish like magic. It's a wonderful, absorbing, and satisfying solo game." Although long, the instructions are clear and thorough (Marion is editor of M500, the math newsletter of the British Open University).

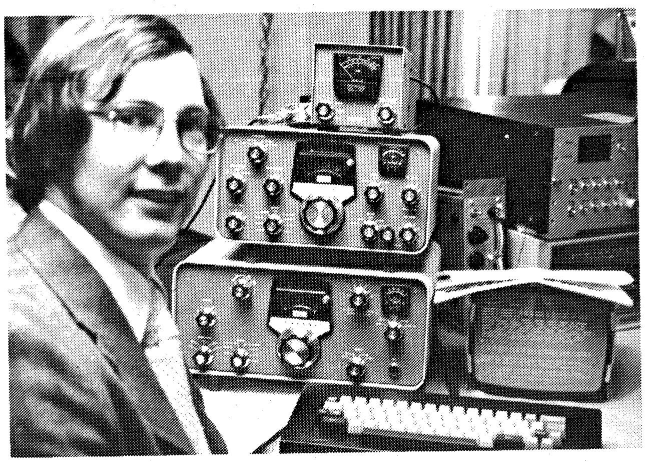
Cost of the 52-page game booklet is only $2.00. Send cash (overseas checks are bad news going either way). Marion says, "we like dollars here, even if you don't like £s."

Marion Stubbs, 176 Midanbury Lane, Southampton SO9 4GX, England.

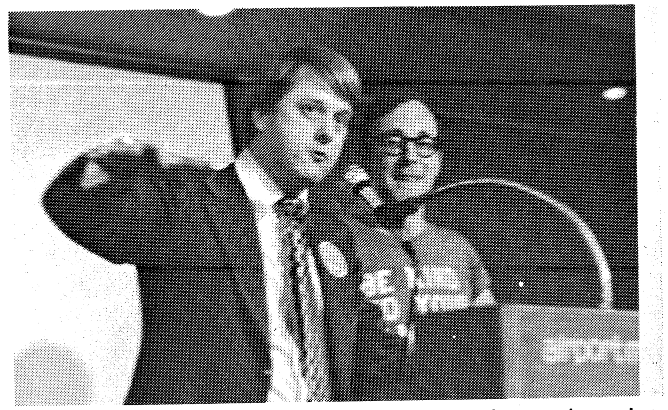
No More, Please

Peter Weiss wrote asking me to inform readers that he can no longer supply Fortran Star Trek listings. He apparently got deluged after his letter appeared in *Creative.* Please, no more to Peter.

mits World Altair Computer Convention



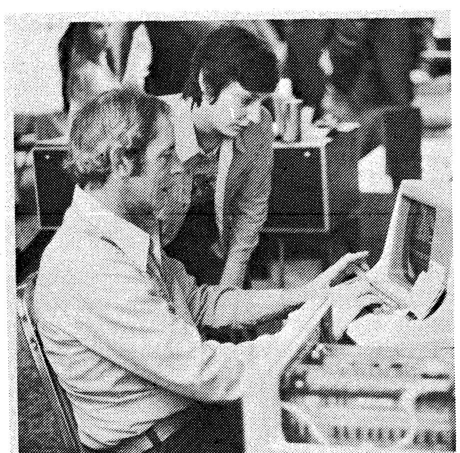
Don Alexander of Columbus, Ohio won a complete Altair floppy disc system in the Demonstration Contest with his computer controlled (Altair 8800 naturally) amateur radio station.



Ted Nelson interrupts Dave Ahl's presentation at the point he predicted that videodiscs will be the medium that will drag computers into the average home because of the immense, cheap storage capability of the disc. Nelson disagreed saying videodiscs are no more real than Phono Vision in 1947.

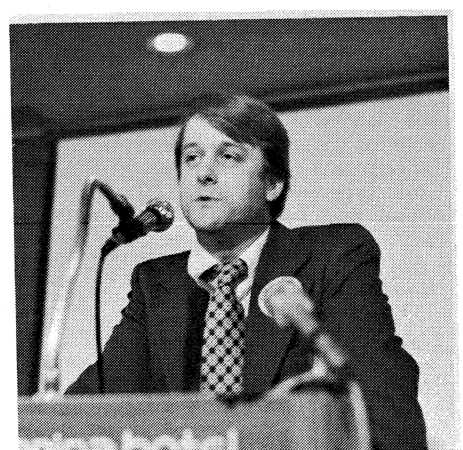
Lou Fields, VP of the Southern Calif Computer Society, presents a trophy to Carl Helmers, editor of *Byte,* in recognition of his contribution to the home computer field.

Some of the 700 people who attended the First World Altair Computer Convention in Albuquerque, New Mexico March 26-28, 1976.

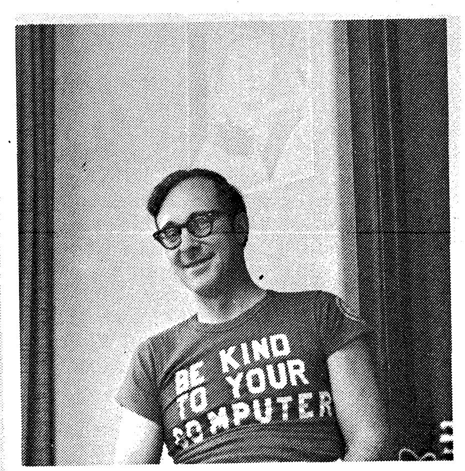


Norm Tilbury and Barbara Solomon of the Daylight Savings Company kept track of visitors to their exhibit on-line at WACC.

Photos by Andrea Lewis and Robert Prati, both of MITS.



Ted Nelson said to WACC audience that hobby computing will remain a cult, a "minority hobby until the machines can be made much simpler." He thinks "the need is for canned systems or black boxes."



Publisher turned peddler. This photo was taken during one of the brief lulls in activity around the *Creative Computing* table.

This section of *Creative Computing* consists of news, notes, quotes, and short bits about this computer age in which we live. It was compiled and edited by Trish Todd, a freshman at Brown University along with David Ahl.

Did this really happen?

Perhaps it happened at Cornell. In an electrical engineering lab a young instructor was desperately trying to organize the first day's work. A freshman came up to him and said, "There's no power at my lab station."

"Sure there is," said the instructor. "I checked them all myself."

"No, there isn't," insisted the student.

"Go test it!"

"I don't have any voltmeter."

"You don't need a voltmeter. Just brush the backs of your fingers across the terminals. It'll bite, but it won't hurt you."

The kid backed off, looking mighty skeptical. A minute later there was a blinding flash in that corner of the lab and breakers popped open all over the place. By the time the instructor got to the bench he found the molten remains of a Stillson wrench hanging from the terminals and the kid backing off, crying, "My God, my God, what if it had been me?"

Clifford Swartz
The Physics Teacher

Computer on Ice

To computerize, or not to computerize? That is the question as the time worn tug-of-war between man and machine gets an athletic yank.

Raymond Epich, a hockey enthusiast and vice president with the Chicago office of Cresap, McCormick and Paget, Inc., management consultants, has applied for a patent for an electronic tracking system to produce computerized hockey statistics in profusion enough to make even a baseball sigh.

The experimental system consists of an electronic grid underneath the rink, plus transmitting devices in the puck and players' skates and sticks that would allow the computer to produce instant readouts of such statistics as the velocity of a shot, the speed of a given player, which players spend the most time on the ice and on and on.

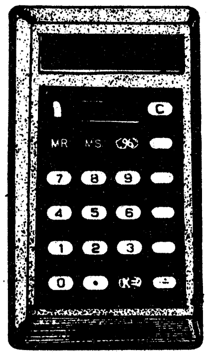
"In hockey you really have only three or four basic statistics," says Mr. Epich, which does not give the hockey freak much to discuss during the off season.

One ardent fan said he would "love to see the idea implemented." But he noted, most fans are skeptical about new ideas that might disrupt the game.

A spokesman from the World Hockey Association in Toronto said statistics "rarely help sell tickets."

"They're talking about big dollars," he added. Mr. Epich readily agrees, giving a ballpark figure of $1-million per stadium to cover costs for implementation. Still, he says he has received some 30 inquiries about the system.

NY Times



Calculators at Work

POCKETCALCULATORS affect work from the executive suite to the classroom.

"You'll find them in more attache cases today than socks or underwear," declares James J. Brown, a Walter Kidde & Co. senior executive. He's able to answer financial questions immediately instead of telling his boss, "I'll be back to you on that." The noise level at AMF Inc.'s headquarters drops sharply as "mechanical monsters" are replaced with the small, quiet devices.

Harris Corp. Chairman Richard B. Tullis uses his pocket calculator daily to figure percentage changes, rates of return and growth rates. An Atlanta Pontiac dealer finds customers whipping out calculators to double-check salesmen's arithmetic. Some college quizzes grow longer as teachers discover students completing work more quickly.

But one Detroit bank executive complains, "I just can't add three- or four-digit numbers in my head" since starting to use a pocket calculator.

Wall St. Journal

But Who Will Read It?

IBM has introduced a laser printer that operates at six times the speed of its currently available hardware. The IBM 3800 will print up to 13,360 lines per minute on plain paper, and be available in the third quarter of 1976. Operating like an "office copier", it will not require supplies like ribbons and carbon paper, and will be able to reproduce form designs on blank paper. Multiple copies will be produced faster than older printers can produce single copies.

The Computerized Eye Doctor

Remember your last eye test? The optometrist changed the eye chart you had memorized, caught you squinting, and then thoroughly confused you by changing lenses before your eyes and asking "Is this better?" or "Is that better?" You started wondering if you could see at all. Well, all that is going to be part of history soon according to the American Optometric Association. The technology that brought us pocket computers is bringing small computerized diagnostic equipment into optometrists' offices. The new phenomenon is called electrodiagnosis.

Before too long, you will sit down in the optometrist's office and let a computer "read" what your brain is "seeing" (the eye collects data and sends it to the brain, where we actually see). Three electrodiagnostic vision tests developed at the University of Houston College of Optometry are already available on a limited basis:

o VER, or visual evoked response. VER is a test used to objectively measure the refractive state of the eye. Today it is being used to examine the vision of small children, retarded persons, the deaf, and others who cannot respond to subjective tests of their vision.

o ERG, or electroretinogram. ERG records electrical responses directly from the eye's retina, the ten-layered inner lining of the eye, and far surpasses the current method of diagnosis, which involves using an instrument called the opthalmoscope to look at the retina through the pupil. Many eye surgeons now rely on ERG to aid them in determining whether or not cataract surgery should be undertaken.

o EOG, or electro-oculogram. EOG records eye movements and helps the optometrist determine, for instance, if a child can accurately and smoothly move his eyes when switching his view from blackboard to desk.

When will the new tests be available? Within the next year or two, claims the American Optometric Association, group practices will be using computerized equipment routinely.

Science Digest

Science IQ Dropping Fast

Can you answer these science questions?

1) In terms of the story of natural selection, what is the explanation of why giraffes have come to have such long necks?

a. Stretching to get food in high trees has made their necks longer.
b. There is something inside of giraffes which keeps making longer necks.
c. Giraffe food contained vitamins which caused the vertebrae to lengthen.
d. Giraffe necks have gotten longer and longer as time has gone on, but nobody has any idea why this is.
e. Giraffes born with the longest necks have been able to stay alive when food was scarce and have passed this trait on to their offspring.

2) Which of the following is good evidence for concluding that glaciers once covered Canada?

a. Rocks containing melted ice have been found throughout Canada.
b. Radioactive decay of uranium in Canadian rocks has been measured.
c. Scratches on surface rocks in Canada look like scratches made by known glaciers.
d. Only glaciers could have formed the high Canadian mountains.
e. Canada has a cold climate which glaciers need in order to keep from melting.

3) What device changes the voltage of an electric power supply?

a. alternator.
b. battery.
c. rectifier.
d. transformer.

4) Of the following, which is the most direct cause of the tides on earth?

a. The tilt of the earth's poles.
b. The magnetic field of the earth.
c. The slope of land near the shoreline.
d. The revolution of the earth about the sun.
e. The effect of the moon's gravitational pull on earth.

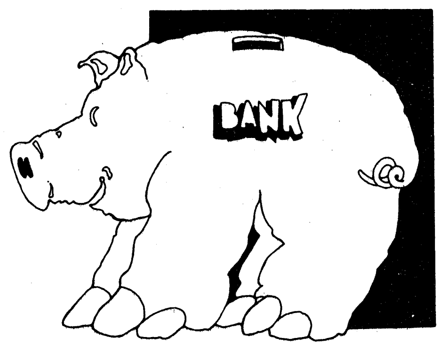
The correct answers are: 1) e; 2) c; 3) d; and 4) e. If you answered the giraffe question correctly, you did as well as 53 per cent of the 17-year-olds asked the question in a recent national survey directed by the National Assessment of Educational Progress (NAEP). If you answered the glacier question correctly, you did better than most 17-year olds; only 38 per cent answered it correctly. If you answered the energy question correctly, you did as well as 46.6 per cent of the 17-year-olds. And if you answered the last question correctly, you agreed with 41.9 per cent of a younger group, 13-year-olds.

NAEP asked science questions about biology, chemistry and physics of 90,000 students ages 9, 13 and 17 during the 1972-73 school year. The results show science knowledge has declined since 1969-70, the last year NAEP made the survey. There was a drop in all age groups but the oldest students showed the sharpest decline. Rural students did better, while big city students and suburban students did worse. Big city students showed the greatest decline. The performance of males and females dropped at all ages, but boys did better than girls and the older the students, the lower the girls' scores compared to boys.

NAEP project director J. Stanley Ahmann doesn't know why science knowledge is declining among students, but he speculates that the better performance in 1969-70 may have been due to the emphasis given science education in the wake of Sputnik.

Science Digest

The Grants Data Bank



The newest and most efficient instrument to match the interests of those seeking funds with those granting funds is the Grants Data Bank.

Take, for example, a grant to bring more bluebirds back to New York. The words "bird life" are fed into the bank, and after 90 seconds of polite whirring the computer prints out five grants. (With electronic thoroughness, the computer also prints out a grant to Blue Bird Circle in Houston to construct a new outlet store.) A name such as "Audubon" would undoubtedly have turned up more possibilities!

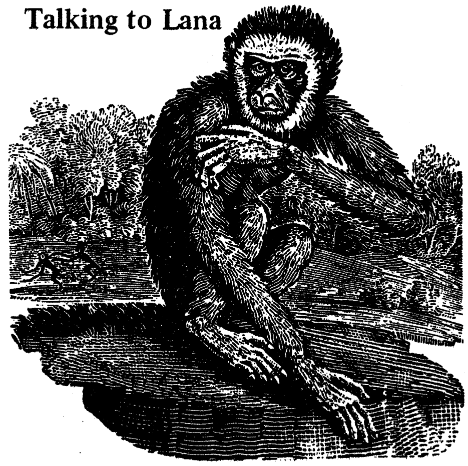
The price for a search (which can be arranged by mail) is $15 for up to 50 grant records and 20¢ for each grant record thereafter. For a broad field such as health, education or conservation, specify the maximum expenditure and narrow the field as much as possible. For example, specify only those 15 foundations that have the largest grant programs for a given field.

For more information contact: The Foundation Center, 888 Seventh Avenue, New York, NY 10019.

Automatic Checkbook Balancing

A checkbook that includes a built-in calculator will be offered soon by the Chase Manhattan Bank. Manufactured by Mostek Corp. of Dallas, TX, the unit can add and subtract and has a constant memory for the balance that is not erased even when the unit is turned off. Similar units are being rushed to market by other manufacturers of pocket calculators.

Talking to Lana



A computer has become the means of communication between Lana, a four-year-old chimpanzee, and the rest of the world. Two years ago, she started to use the symbols on a computer key board to talk to her keepers. Lana quickly mastered about forty words, and now, she is able to use about seventy-five word symbols and is asking to learn more. She has begun to ask her trainers the names of various objects. Lana's ability to learn a language suggests that the mental ability of chimpanzees may have been underestimated. Researchers at the Yerkes Regional Primate Research Center in Atlanta, where Lana is being trained, say that the methods used to teach Lana may eventually be used to help mentally retarded or disturbed children who have trouble learning to communicate.

Science News

Opportunities for the Deprived

A computer has helped 600 educationally, physically, and economically handicapped people find vocational or educational training that met their specific needs. This computer is part of a program called Computer Based Educational Opportunity Center, sponsored by the City University of New York and located at the Henry Street Settlement House.

Programmed with more than 500 academic and training sources in all five boroughs, the computer can find a workable solution to fit the special needs of most people. It has, for example, found a key punch operator training center for a handicapped veteran with a child who needed day care services, a woman who wanted a career in carpentry, and a man who could only spend $500 a year for his education to become a chemical engineer.

There are five counseling computers in New York; however, four are at colleges, and Henry Street is the only social agency using the program. The computer terminal is small, so it is easy to transport it to other New York neighborhoods. The program may well become a vital resource in bringing education and careers to low income people.

Lost? Get a Computer

Computers have helped man gather billions of pieces of information, classify them, and file them. However, when studying large amounts of data, it is hard for man to make comparisons, establish relationships, and solve problems. Through graphics and mapping techniques, computers can organize data into formats that will help man deal with data more efficiently and effectively.

A popular method of mapping has been shaded area mapping. It is a computerized way of drawing a map and coloring in different areas according to the given qualifications. However, because computers use digitized formats, it is hard to program an area that is not rectangular, such as a country, city, or county. It is also difficult to compare and combine data using the shaded area mapping; the resulting area hides its elements.

A new approach to the problem of computer mapping has emerged called symbol maps. Symbol maps pose no restriction on the amount of data to be displayed or the design of the symbols. The two types of plotted symbols used most often are the circle/square symbol and a focus indicator symbol. When using the circle/square symbol, the square acts as a universe, and the circle is a subset. The square may also be divided vertically or horizontally, and circles can interlink. The focus indicator symbol is a circle divided into arc fans. The size and number of the fans are easily adjustable and can be determined by the given data.

Both the circle/square and focus indicator symbols will become more familiar to the public as their use is increased and perfected. These two methods of mapping will continue the computer's job of helping man gather data to solve problems.

Worthless Information Department

For those people lucky enough to have Texas Instruments SR-22 calculators:

1) Turn on calculator.
2) Set to 'manual'.
3) Change base to Decimal.
4) Punch in 4 444 444 (seven 4s).
5) Hit the HEX conversion key eight times, leaving the calculator in DECIMAL. After the eighth HEX conversion the display will not come back – congratulations! Your SR-22 is in an endless loop. The *only* way to clear it is to turn the calculator off and then on again.

I know this works on at least two SR-22s, so it probably works on all of them. I haven't the faintest idea why this happens – the machine does funny things on repeated hex conversions in decimal mode. Someday I'll get around to writing T.I. and asking them.

Don't ask me how I discovered this. *I* don't know how I discovered it. I just like to punch buttons and play with things. Call it Creative Insanity exemplified.

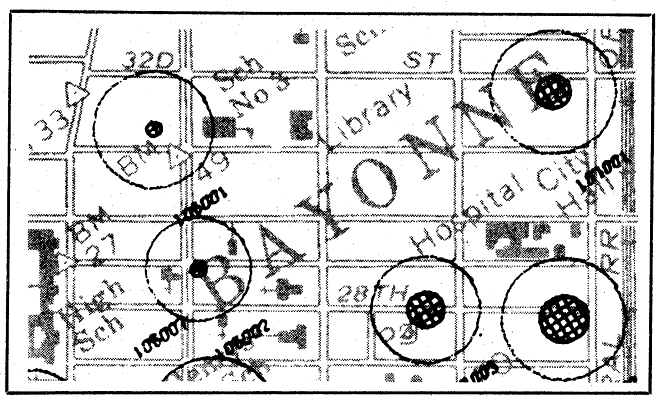
John Lees
Rolla, MO

Computers in Medieval Days?

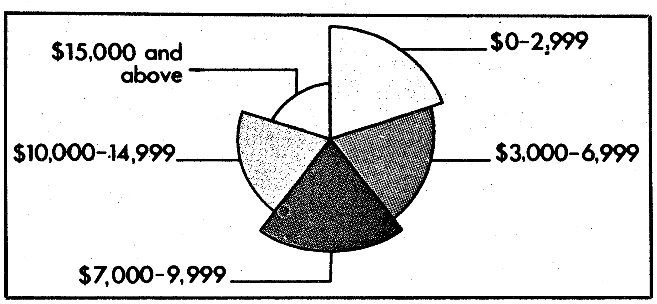
Finding evidence for historical analysis is rarely easy, particularly for periods in time that left us with few records, such as the Middle Ages. History has always pictured the Middle Ages as a time of glamorous chivalry and feudal wealth. However, with the aid of a computer, a census called *Roles des fiefs* (list of estates), taken of a territory in thirteenth century France, has shed a completely different light on this period.

The census was a record of the economic and social situations of 1,182 persons who held lands from Count Thibaut IV de Chansonnier of Champagne. Quantification by the computer showed that feudal lords, who owned huge castles, had judicial and taxation rights, and had fief-holders of their own, were a distinct minority. The usual aristocrat had a small income and lived modestly, as other rural farmers did. The majority of these aristocrats were knights, rewarded only with a title for defending their faith and their lord's lands and ladies.

Previous medieval social history has been drawn from charter evidence and ecclesiastical property records. But Professor Theodore Evergates of Western Maryland College offers the findings of the computer as a corrective to the traditional picture of the medieval society.



The circle symbols on this map show the distribution of school age children. The full circle shows the total number; the shaded area, children in low-income families.



The focus indicator is a powerful tool for display when data elements and map options are chosen carefully.

WORLD'S LARGEST CHECKOUT

Canton Central Township, Michigan – The world's largest checkout system has been installed in a Meijer, Inc., superstore in this suburban community 30 miles west of Detroit. Shown above are some of the giant store's 58 checkout lanes. There are another 31 checkout stations in separate departments throughout the store. The half-million-dollar electronic checkout system includes, in addition to 89 NCR 255 checkout terminals, four NCR 726 in-store computers which control the operation of the system.

"As soon as you can say what you think and not what some other person has thought for you, you are on the way to being a remarkable man."

James M. Barrie

Talk, Type, and Now Write Over Telephone Lines

Scientists at Bell Telephone Laboratories have developed a video system that transmits handwriting, reproduces pictures, and can be used to communicate directly with a computer. The new system is a step forward in improving interaction between people and machines.

The system consists of a commercially available plasma panel display (modified by Bell Labs), electronic control circuitry, and a special light pen.

The panel is made of thousands of tiny neon-gas cells which glow when energized by an electric current. These glowing cells may be turned on and off selectively to produce images; an image may be sent from far away if the correct instructions are given over a telephone. If two panels are connected, "writing" on one panel with the light pen will produce the image on the other panel. The tone of the dots in the images may also be varied by changing the density of the cells that have been charged. This quality is essential for truthful reproductions.

The system is still being developed, but some potential, practical uses of the system have already been proven. The system can:

- record a person's signature and display it on command
- reproduce pictures and charts at the rate of a few per minute
- display a list of telephone numbers and simulate dialing the number pointed to with the pen
- serve as a desk top calculator
- display the time and date on command.

Bell Labs Contributions to Digital Computer Technology

Efficiency is one major benefit of the computer, and as the operation of a corporation becomes more complex, detailed, and time-consuming, it frequently is possible to design computer systems to cope with these problems. For example, Bell Telephone Laboratories have designed a computer system for telephone repair service. LMOS, the Loop Maintenance Operating System, allows local Telephone Company repair service bureaus to make automatic tests, under minicomputer control, of customer lines. The minicomputers have access to data stored in a large computer in another location. Often, trouble can be diagnosed almost instantly. LMOS, introduced in January 1975, is one of the most recent developments by Bell Labs in their long history of contributions to digital computing technology.

HIGHLIGHTS OF BELL LABS' CONTRIBUTIONS TO DIGITAL COMPUTING TECHNOLOGY

1937—Application of Boolean algebra to the design of logic circuitry.

1939—Complex Number Calculator (Model I Relay Computer)—the first electrical digital computer built of relays.

1941—OR circuit.

1942—AND circuit.

1943—Model II Relay Computer—produced punched paper tapes for dynamic tests on fire-control equipment. Included error-checking capability.

1944—Model III—relay computer for fire-control problems. Could hunt through paper tapes for address of a block of data.

1945—Model IV—relay computer for fire control problems for ship-mounted guns. Could calculate trigonometric functions.

1946—Model V—general purpose relay computer with two processors, permanently wired math tables, floating decimal point, conditional transfer capability.

1948—AMA (Automatic Message Accounting)—relay computer for extracting billing information from phone calls.

1949—Transistorized gating circuitry.

1950—Model VI—last of line of relay computers. Solved telephone R&D problems. Had automatic second-trial feature, useful for unattended operation.
Air-cushioned magnetic recording head. Widely applied in commercial drum and disk storage systems.

1956—L1 and L2 (also known as Bell 1 and Bell 2) allowed users to communicate with computers in a language far simpler than basic machine language.

1959—Leprechaun. Had 5500 transistors and 18,000 memory cores in 15 cu. ft. volume. Ran on 160 watts.
Macro instructions; allowed programmers to add terms not included in the original programming language.

1960—TPLOT, programs for producing computer-generated plots and graphs on microfilm.
SNOBOL, a programming language for manipulating strings of characters.

1961—ALTRAN, language for making symbolic computations on algebraic data.

1963—GRAPHIC 1 system allowed user to communicate with a computer through a cathode-ray tube display.
BLODI, program that allowed circuit designers to test circuit designs without actually building them.

1965—SWAP, universal assembler program containing many features previously available only in separate programs.
Fast Fourier Transform algorithm for efficiently processing complex signals in real time.
L^6, programming language that allowed programmers to manipulate complexly linked data, write faster-running programs, and use computer storage more efficiently.

1968—GRAPHIC 2, advanced version of GRAPHIC 1 for use by drafting personnel, circuit designers and engineers.

1969—UNIX, a time-shared software system for minicomputers. Used for such diverse purposes as text editing, general computing and switching system trouble reporting.

Input/Output

Humanistic Computer Uses

Dear Editor:

I am working on a survey of the humanistic use of computers and would like to get in touch with other people who are interested in this field. I want to learn about what can, and is, being done to make the use of computers as humanized as possible. More specifically, I want to know:

- What are the potentially humane applications of computers?
- What are the important ingredients in humanized computer systems?
- What is currently being done in the field of humane computing, both in terms of theory and practice?
- Which individuals and groups are actively doing this work?
- What are their particular goals and objectives?
- What are the results so far?
- What are the major factors determining their successes and failures?

Do computers have a role in the humanizing of our society? Can we use them to facilitate the creative expression of our individuality and the healthy fulfillment of our potential? Can the use of computers help us to communicate meaningfully with each other, so as to promote mutual caring, understanding and respect? If computer systems can indeed contribute to furthering humanistic ideals, what would such systems look like in principle and how well do existing and planned systems measure up to these aims in practice?

Or perhaps computers, by their nature, are inappropriate for such a role in people's lives? If so, why, and what can be done to ensure that computer systems are as humanized as possible? Just what are the limits and limiting factors in the humanistic use of computers?

I would like to hear from anyone who can contribute to answering such questions. They should write to me describing their thoughts, fantasies, plans and experiences related to the humane use of computers.

Andrew Clement
789 West 18th Avenue
Vancouver, B.C.
Canada V5Z 1W1

Humanistic Computer Jobs?

Dear Editor:

This may sound strange but the whole idea of humanizing computers is what has been pushing me along for my degree. I'm looking around now for a job paying good money, doing just that. If you have any ideas let me know.

Mark S. Mayes
Boston University

Editor's note. If anyone has any ideas for Mark, knows of any jobs, or shares his concern, let us hear from you.

Strange Phenomenon

Dear Editor:

I recently came across a strange phenomenon which may be of some interest to you. Through a head injury which is caused by gassing it is possible to transmit sound and pictures over a wireless wave from head to head over long distances. Individual heads are turned into what could indeed be called small radio-television transmitters and receivers.

As far as I can make out the gassing enables an electrical field which is normally enclosed within the body on a circuit of its own to be tuned into. This forms a circuit over which signals can be transmitted.

In transmitting there isn't any frequency fade. The pitch of the signals seems to be several tones higher and several tones lower than ordinary sound. The sound isn't audible to the ear in the usual way. It is heard within the head or out from it according to the way that it is transmitted. It is possible that the ears are affected in such a way that they 'hear' outside the normal range of sound. I think that the sound could be thought of as being in a quite new and unresearched tonal range.

Several people combine together to tune into the head. Broadcasts between heads are almost continuous. It has occurred to me that it might be possible to tune into such broadcasts on specially made metre bands in ultra high frequencies. The fact that pictures can be transmitted as well as sound suggests that broadcasts take place on very high frequencies.

No research is being done here in Australia on this problem. I understand that this kind of communication falls into the category of Cybernetics, a relatively new field about which very little is known. Both the Russians and Japanese are carrying out studies where it is concerned.

To my mind the thing is a security risk in that in times of war transmissions could be made other than by orthodox means. I realize that all this will sound incredible to you but I hope that you will be able to give it some serious thought.

If the frequencies on which transmissions are taking place could be determined it might be possible to design a piece of electronic equipment like a small transistorized transmitter that could block signals being sent into individual heads. Enormous numbers of people seem to be involved in it here. Is any research being done in this field in America?

E. Rowlands
Sidney, NSW, Australia

Health Care Applications

Dear Editor:

I would be very interested in corresponding with anyone in the health professions or with an interest in health care delivery data collection and recording about programs such as [the one I wrote for an IBM 5100 to collect clinical data].

Our intention is to begin in a very basic (no pun intended) manner and slowly graduate to more sophisticated programs on interacting health care problems. It is not our intention to write programs which will diagnose anything. That sort of work requires a tremendous amount of computing capability which we do not have nor are we desirous of obtaining. There are many other institutions in the United States, the University of Missouri at Columbia being one of these, who have done a far better job at that than we will ever be able to do. Our entire thrust is toward the individual practitioner of medicine,dentistry, or veterinary medicine who now has within his reach because of present day technology, the ability to collect, record, and retrieve data in a far more efficient manner than he has ever been able to do before.

Richard E. Easton, M.D.F.A.C.P.M.
President
Health Information Services, Inc.
5341 Parliament Drive
Professional Bldg., Suite 104
Virginia Beach, Virginia 23462

SNOBOL

Dear Editor:

Being a member of the all-too-small group of people who know and love SNOBOL, I was disappointed to see it only barely mentioned in "On Computer Languages" of the September-October issue. Enclosed is a brief description of this mind-blowing language. BASIC may be the best beginner's language in the world, but there *are* problems of a higher nature than SPACEWAR, ya' know.

By the way, PL1 is really spelled PL/I (that's a Roman numeral—IBM's attempt at class). PL/I was supposed to replace FORTRAN, COBOL, assembly language, and everything else, as the special language of the System/360. It is a general purpose language: "good" for lots of things, but not great at anything Math freaks find much more versatility in APL; string manipulation and linguistics people prefer more sophisticated dialects like SNOBOL; artificial intelligence researchers think LISP is best (for LISt Processing, anyway); and operating systems people know that assembly language is far more efficient. "What *is* it good for?" you ask. PL/I is a good replacement for another unimaginative language: COBOL.

Computer Science types also like PL/I, because it allows them to practice structured programming, a technique necessitated by the poor quality of certain languages, notably COBOL, PL/I, ALGOL, and related diseases. I recently watched a group of Intro-to-Programming students hack their way through a PL/I program to do some simple string manipulation. What they did with mountains of DO WHILE's and SUBSTR's and INDEX's could have been done in half a dozen lines of SNOBOL. Likewise, the hairy matrix inversion and statistical problems many FORTRAN students are subjected to could be done in a few lines of APL.

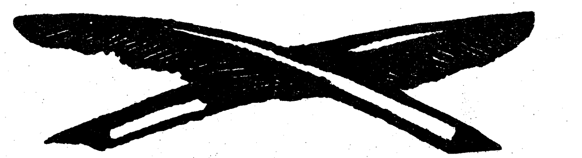
Granted, beginning programmers must learn the concepts of looping, conditional branching, etc. But this can be taught in a more comfortable environment like interactive BASIC. Some of the BASIC interpreters running on minicomputers are really beautiful. And when a student moves on to more complex areas of study, he should be supplied with the appropriate tools. SMALLTALK is an excellent example of an intelligent language for a specialized application. Viva Xerox PARC! I think you'll find that most of the creative programming is done in languages like BASIC and FORTRAN because of their flexibility. COBOL, PL/I, and RPG are for the production-oriented business world, where results are more important than aesthetics.

The high schools with PDP-8's running BASIC/PAL/FOCAL software are still way ahead of the average dp shop with its 360/50 under DOS producing umptween carbons of reports and audit trails no one would even want to read. Computers are mind-expanding tools: programming languages should be designed for this purpose.

I may have made a few readers angry by this time, but isn't that good? Maybe they'll write you about their opinions.

David Touretzky
Rutgers University

Dave's article on SNOBOL may be found on page 36.



About hardware, games cabbages, kings, etc.

Dear Editor:

I thought the last issue was perhaps the best thought out one I've seen. The program which did geometric proofs was fascinating, really. But I wish you had included the source code, no matter what particular language it was in.

Also liked that "Report on Current Equipment." That was really subtle ... despite all the clues in it, I don't think a single student in my HS would be able to identify what was being discussed. I didn't! As a matter of fact, you nearly had me writing a nasty letter about "Jehovah's Instruments" (I'm a Jehovah's Witness).

But there are a few things I forsee as a problem in CC. I'm not sure what stand you have taken with regard to writing about hardware, but its my guess that you have decided not to get too involved. Which is good, since other mags are already doing a good job, and it is nice to see some variety. So it would seem more logical to me to just completely ignore hardware entirely, rather than write (not you, but your staff) very generalized stuff like, "Microprocessors are very small. They come in ICs. They will get cheaper as time goes on." Anyone who really cares will need a lot more info, and anyone who doesn't care will skip the article. I guess system design does have a lot to do with AI but as time progresses hopefully hardware will become more transparent to the user — he won't have to bother with exact system configs. and other meaningless technicalities.

Another point I wanted to bring up ... I think a lot of hobbyists are getting a little carried away with their Altairs and other home systems. I'm not referring to you, but to other people. One hears a lot of talk about computer revolutions, and computers changing our way of living, and a computer in every home, but if there is going to be a computer in every home, it won't be anything like a black box with lights on it! That is no more useful to Joe American, than say, a flashlight. Less useful. Something which does nothing but eat up around $1,000 and play games and maybe balance a checkbook (now emptied, of course!) will not appeal to many people. Computers will have to show up in ways that they can be easily used by people who know nothing about binary numbers and addresses and compilers. Most people have televisions and radios which can be operated by a 6 year old, and not shortwave sets requiring years of training to operate. Every technological advancement has become popular when it was reduced to a form where the users did not have to become familiar with its working innards ... except the business computer, perhaps. Sounds like good material for an editorial!

Some really strange things going on at the HS here. For a couple of months we had an outbreak of masochism! It is *really* strange to watch a bunch of high school seniors do stuff like pour milkshakes in their pockets ("It was thirsty") or punch themselves out, or dive headlong into a muddy ditch, or throw their watches and glasses down a hall. The masochism has died out and been replaced by total apathy. A lot ot teachers complain that they can't cover the same material with us as they did in previous years.

Please try to avoid making *Creative Computing* too much like Popluar Science by skirting over many "scientific" topics and not going into them in depth. Be sure to live up to the name of your mag. The geometric proof writing program was the most creative program I've seen in a while. I realize that this is probably a very slim possibility, but if you need any paid help with CC for cheap let me know ... I'm looking for a job this summer.*

One other thing. There seems to be some feeling that the quality of the games in *Creative* and *PCC* are too low; one person even said he never got a game out of a magazine he wanted anyway. I personally subscribe to the school of thought that part of the fun in playing a game is programming it and knowing how it works, and being able to change it. My physics teacher came up with a good rule for determining if a game is worthwhile— "If it doesn't use any unique aspect of the computer — such as its ability to store large amounts of data, manipulate them very fast, randomize numbers, or print things out in a particular way, then the game isn't worth while." Which is what was wrong with MASTERMIND in latest CC ... it could be easily played the normal way. Having the computer play would be interesting as an example of AI.. Similar with Bobstones. Deepspace fit the criterion for a worthwhile game under this rule, however. Just thought you might want to think it over ... it's my guess that you are really not a computer games person, and the types of games appearing in Creative bears this out. Don't take that as an insult, its just a fact. There are more important things than games ... recreation is an end to itself, but not a useful end with respect to other goals.

Believe it or not, the heads took over the computer room here! It is really strange ... all these freaks manage to stop smoking the little green roll-your-owns long enough to play football on a TTY. I would say it was fine with me, except they are really messing things up. Last year, 1 BH (before heads) the computer room was clean and neat, and equipment broke down rarely and was fixed immediately. Now the equipment is always in in partial disrepair and is littered with trash. I guess recreational computing can have its disadvantages.

Yours truly, etc. etc.
Steve North
7 Deerhaven Lane
Newfoundland, N.J. 07435

Not so slim after all. Steve is working for* **Creative *this summer (1976) on a variety of things (games, hardware reviews, etc.)*

Attn: Coursewriter Users

Dear Cyberfans:

The Freehold Users' Group is wondering if any of you know of any installation using the "BASIC/Coursewriter" implementation of BASIC. If you know of such an installation or if you yourself use such facilities, please send the name and location of the installation, type of system, your name and address, and your relation to that system (supervisor, student, instructor, passer-by, etc.) To: Anthony Begonja, 31 Brookside Road, Freehold Township, New Jersey 07728

We are interested in communicating with and trading programs with fellow coursewriter users, especially the kind of programs that fully accommodate the various quirks in the coursewriter 1.1 system. This includes the text editor written by the Freehold Users' Group for this kind of implementation and it is available upon request.

Anthony Begonja
For the Freehold Users' Group

Comments on the Turing Test

Dear Editor:

In the Mar/April issue of Creative Computing, Ref.1, Lewis Garrett makes a common mis-statement of the Turing Test as follows: "It consists of a man (the examiner) trying to discern whether or not the responses to questions he has proposed are being answered by a computer or another man."

The actual Turing test, in Turing's own words is given in Ref. 2: "The imitation game is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman.... A's object in the game is to try and cause C to make the wrong identification ... The object of B is to help the interrogator... We now ask the question 'What will happen when a machine takes the part of A in the game?' Will the interrogator decide wrongly as often when the game is played like this as when the game is played between a man and a woman? These questions replace our original, "Can machines think?"

There are two interesting aspects of this formulation:
1—Turing obviously felt that faking, as far as sexual identification is concerned, requires a high level of intelligence.
2—Turning seemed to feel that the sex of the examiner was unimportant.

An experiment using the Turing game

I have run this experiment half a dozen times in classes at San Jose State University, as reported in Ref. 3. Briefly, a male and a female student are sent out into the hall and the man is told to fool the class into believing he is the woman. The class makes up questions to determine which response comes from the man and which from the woman, using a courier to carry the messages back and forth. The experiment is repeated with the woman trying to fool the class into thinking that she is the man.

It turns out that Turing was indeed correct in surmising that this type of faking requires a high degree of intelligence. However, he failed to realize that the sex of the interrogator is of crucial importance: only a woman can make up good questions to detect a man's faking, and only a man can make up good questions to detect a woman's faking.

By now I know the areas for good questions for detecting both types of faking, and I now know about a certain type of question that can devastate fakers in any field, but I won't spoil the game for you by revealing them at this time. Play the game with some friends and enjoy learning about the nuances of fakery on your own.

Oscar Firschein
Lockheed Palo Alto Research Laboratory
3251 Hanover St.
Palo Alto, CA 94304

References

1—"Primer on Artificial Intelligence," Lewis E. Garrett, **Creative Computing,** Vol. 2, No. 2, Mar.Arpil 1976, pp. 20-24
2—"Creative Machinery and Intelligence," A.M. Turing in "Computers and Thought, edited by Edward Feigenbaum and Julian Feldman, McGraw Hill 1963, pp. 11-35
3—"The Turing Women's Liberation Index," O. Firschein, **DATAMATION,** April 1974, p. 28

"Learning Programs — A Challenge"

Dear Editor:

For some time, I've toyed with the idea of developing a program that "learns." I realize that this has been done by the chess-playing monsters, but how about a simple learning tic-tac-toe program? I have not had the time to work on this much, but the task is a little more difficult than it seems on the surface. I have taken the approach that the tic-tac-toe program "remembers" all previous games. Through some algorithim, it never repeats a losing game. Therefore, eventually only winning (or tie) games are left. I'll admit that this is more "remembering" than learning, but its a crude approach that works. The challenge is in getting the program to fit into a minicomputer *without* a disk. So—if you're interested, I'd like to see what your readers can come up with under these constraints.

1. Tic-Tac-Toe must fit into 32k and not use a mass storage devise.
2. Use any "learning" approach you can dream up.
3. Program must start out ignorant—how long (how many games) does it take before the program wins 10 in a row?

Peter Weiss
COMSAT, Room 4051
950 L'Enfant Plaza S.W.
Washington, DC 20024

Regarding People's Computer Company

Dear Editor:

Upon re-reading the articles on WUMPUS which you have so kindly published I noticed several remarks I made regarding People's Computer Company and its staff. PCC was the birthplace of Wumpus and its first proving ground. The open and creative atmosphere and playful attitude towards computers at PCC allowed the generation of many computer games with truly new ideas.

The creative ferment caused the strong expression of many viewpoints towards ideas. My remarks are meant as views I had towards certain IDEAS and are not meant to be taken personally.

Gregory Yob
Lo-Op Center
8099 LaPlaza
Cotati, CA 94928

"Some men make artifacts. To date, artifacts haven't made any men."

April 3, 1976

Dear Editor:

"A bird is an instrument working according to mathematical law..."

Which, of course, is manifestly untrue. Birds aren't instruments, they're birds (as da Vinci well knew); the statement shocks, and was meant to shock, precisely because this is so and known to be so. Otherwise one is simply saying that instruments are instruments which if unexceptionable is damn' dull.

We are dealing here with metaphor, the indispensable (if hazardous) tool of all science as it is of all poetry. We are attempting to describe on *aspect* of a real thing ("bird") in terms of something else ("instrument") which it is *not*.

The power and the limits of metaphor, more particularly of its proper subset, analogy, were favorite discussion topics in the Medieval universities. The whole of theology, for example, hangs on this one peg.

Our age hates metaphysic and pays the price: it blunders constantly into logical traps. What da Vinci is saying, I suspect, is this: with respect to those apparent regularities of nature that we like to call "laws", a natural flying creature flies in the same universe as an artificial one.

Whether merely understanding a natural entity infallibly leads to our replicating part of its behaviour with an artificial device I don't know. Airplanes probably had to wait for a most unbirdlike engine and fuel before they could be invented; still, in the meantime a good deal of effort was wasted in the futile attempt to fly *as birds do*. If one excepts the lost work of Prof. Icarus of Greece, it appears that the ornithopter is not cost-effective.

For that matter we still can't fly as birds fly, nor do they fly by our methods. Helium-filled & rotating-wing species are rare birds in Rhode Island, and only in Aristophanes do they propel themselves by ejecting a jet stream from the backside.

I rather think there's a moral to this.

M.1: The analogy between nature and artifact is a useful starting point that is quickly outgrown. Then it becomes a dead weight and generates fallacies.
Computers are not human, which is why they are useful. There are plenty of humans around, after all.

M.2: Statements made about one aspect of a thing become false when the qualifiers are removed.
With reference to electrical shock, I once read that "a human being is a 50,000 ohm 1-watt resistor."

M.3: The claim that some man or some academic discipline possesses real "hard" knowledge undimmed by analogy or metaphor is self-contradictory.
There are at least 4 metaphors involved in making such a claim!

M.4: The curse of our age is the sundering of human culture, and human psychology, into the "two cultures". This fragmentation is greatly accelerated by fallacies like M.3 above.
The result is to oppose one "hemisphere" (to use current mythology, against the other, one man against another, one branch of learning against its badly needed counter-poise.
And so art will lose the love of order, structure, meaning and concrete reality which alone can give it form; and computer programming loses the intuitive, the aesthetic, the all-important criterion of elegance which — to me — is the only justification for programming at all.

I am suggesting, in other words, that motorcycle maintenance and Zazen are alike prerequisites to Satori, to enlightenment.

Digression: I weary of hearing that all programming depends on the ability to flowchart, in approved standard symbols, everything that a program will do.

Respondeo Dicendum Quod (I reply that one must say: "—the standard Medieval prelude to taking a formal position on an issue):

This pretty fable collapses in the face of reality. One never knows what the interaction will be between the actual coding process and one's preconceived notion of structure or flow. I have never written a program or system of programs of any significant complexity without having a good many preconceptions shot to hell. Like electrical circuitry, the diagrams that matter are written after the thing is working — not before.

And this, to me, is the educational value of programming! It is one of the rare places where the young are shaken out of the notion that we so sedulously force down their — alas — trusting throats, that there is an immediate "right answer" to every question; that Teacher knows it, or else has the special edition of the book with bound-in answers (never mind that a substantial fraction of them are wrong!); that wisdom consists in memorizing what Teacher says and imitating what he does.

Well, I've never known a teacher who learned to program with half the ease of an average, if motivated, student.

End Digression.

Parting Thoughts:

1. Another analogy to the processor, suggested by *Travels in Computerland:* the super-duper Lionel electric train set, the biggest and most expensive available toy!

Surely no one believes that adults differ *in kind* from children? Children are as highly sexed as we are, they seek as we do challenge, interests, amusement in their games, and like us again they play sometimes with dangerous toys they cannot control. Or have we adults subjected to Pure Reason the bomb, the Presidency, napalm, "bugs", radioactive waste or the pace and tempo of modern life?

2. Re birds, most of the "birds" man has built were built to kill men; or (a more recent improvement) cities, nations, perhaps the whole planet. As for computers, they are a spinoff from aerospace, the most militarized of all industries. I don't draw any immediate necessary conclusion, but a little sobriety seems called for.

3. Some men make artifacts. To date, artifacts haven't made any men. To explain man by the machine, to explain Henry Ford by the Model T, is preposterous in the ancient sense (backwards). It ignores the orders of causality and of temporal sequence. It is somewhat more natural to proceed the other way, and to explain the work of art or of artifice by studying the artist or the artisan.

4. Must "Creative" stoop to the arguement from scorn? Is it only a fool that thinks human knowledge "mystical" or mysterious, when it is the one thing that human knowledge inself can never quite know?

Our own mind, after all, is not a "thing": it is not something external to us, which we can observe and not contaminate by observation, nor colour by the interposition of self.

Man pondering man is subject to all the self-referential paradoxes of Lord Russell, analyzed rather shrewdly by St. Gregory of Nazianzus some 1600 years before his modest Lordship named them for himself.

So we compromise, as da Vinci did with his bird. We work by analogy, we analyze a part rather than the whole, But ruin awaits us if we drop those qualifiers!

5. One cannot help noticing that hand-in-hand with the reduction of man to mechanics goes a new mysticism of its own, a deification of the machine. Have we come, then, to the worship of the works of our own hands? Has "left-hemisphere" rationality brought us full circle to idolatry?

"Similes illis fiant qui faciunt ea", as another page of your Gutenberg would show. Men who make idols become what they make: sub-humans.

6. Is it too rash to suggest, instead, what your magazine cover carries so splendidly, the first words man set to movable type: In Principio erat Verbum. Before all else there was the Word ... all things were made through it (or him) and apart from the Word was nothing made. In him was made Life, the light of man: the light shines in the darkness, and the darkness cannot overpower it.

True, maybe; false, maybe; "mystical", certainly. But foolish, surely. And yet is it altogether to be scorned?

7. Remember, in parting, that someone once proved hummingbirds can't possibly fly. This saves one from all sorts of observational error!

8. Two quick thank-yous to your authors:

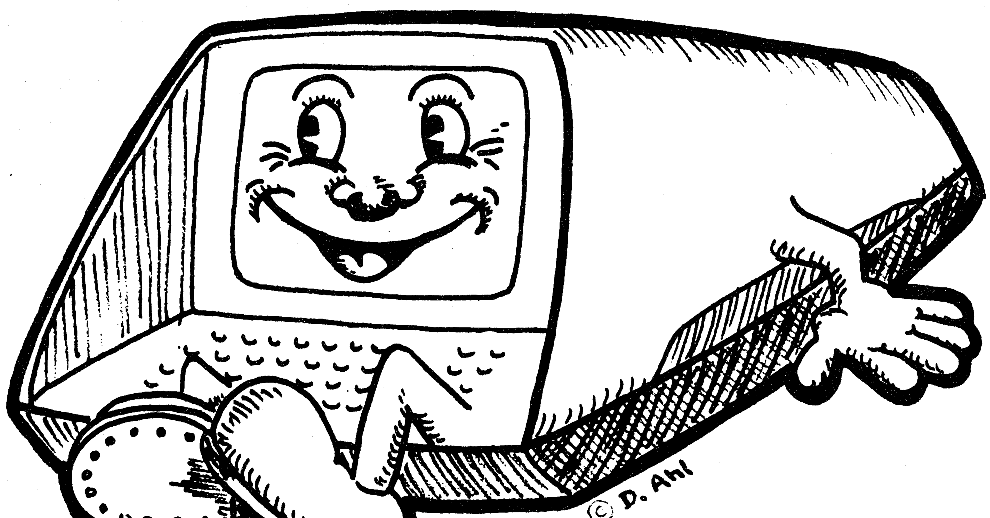
A. "Information is in no sense truth" (and some of it is downright false). Amen

B. The positive theme of your issue — the future of the computer is unforseeable, but certainly vast — is true & needs saying.

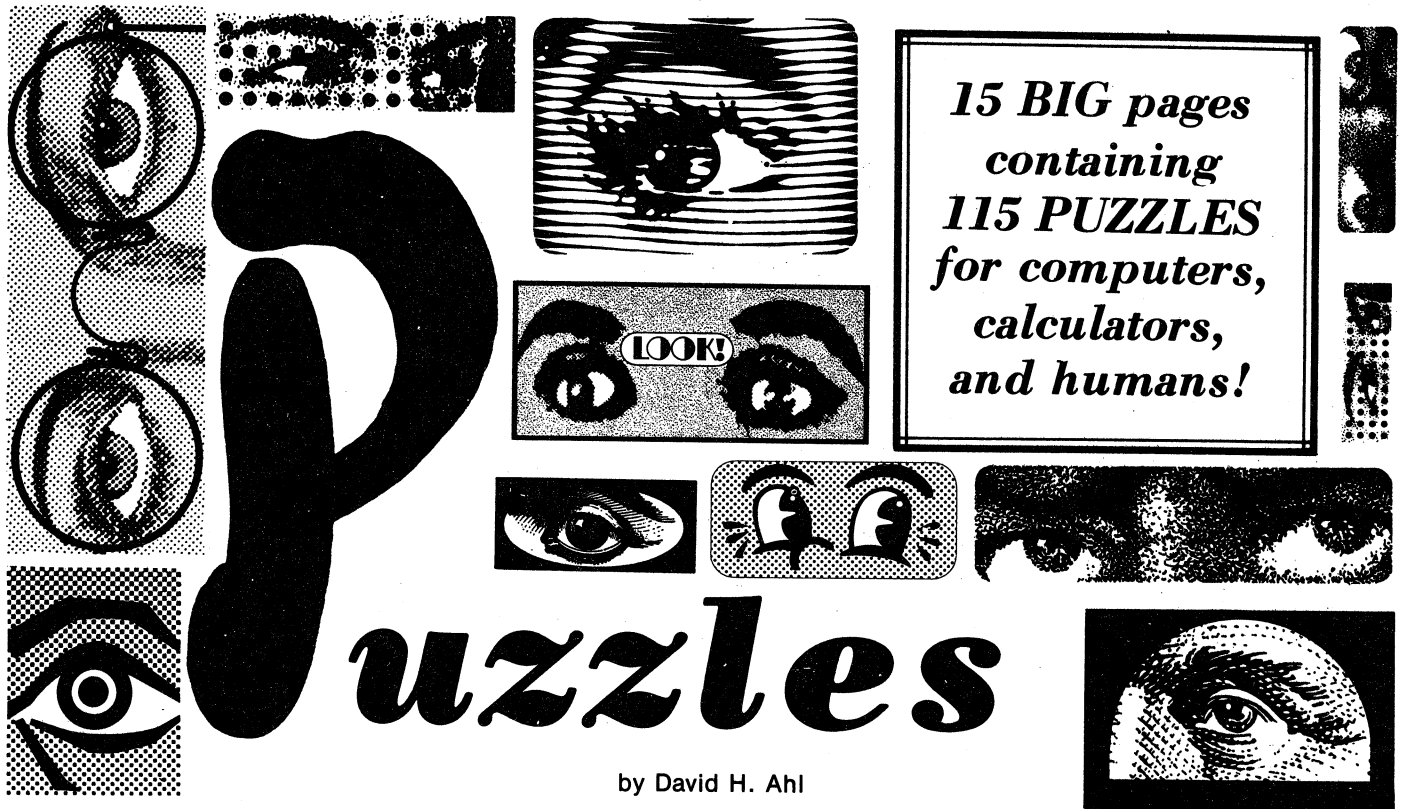
Yours devotedly,

(Rev. Dom) Geoffrey Chase
O.S.B., S.T.L.
Abbey
Portsmouth RI 02871

Programs, Puzzles, Problems, and Activities



Cyrus CRT



by David H. Ahl

How does one solve a puzzle or problem? There are direct and indirect approaches. Frequently, the "scientific method" is advocated as the generalized approach to solving all kinds of problems. Stated briefly, the scientific method consists of the following steps:

1. State the problem. Break it down into manageable pieces, if necessary.
2. Collect facts and data.
3. Using the data, try a solution. Does it meet the objectives, i.e., does it solve the problem? If so, is it the best solution? If not, go back to Step 2.

But how do we get the solution from the data? Several possibilities exist:

1. *Deduction.* Reaching a conclusion from something already known.
2. *Inference.* Reaching a conclusion from facts and evidence.
3. *Trial and Error.* Keeping at it, avoiding past mistakes until you get it right.
4. *Experimentation.* Trying something new and observing the results to achieve a goal.
5. *Intuition.* Direct perception of the truth that bypasses analysis.

In the puzzles and problems which follow, you'll probably have to use all five of the above methods at one time or another. Sometimes a calculator or computer will come in handy—but it's up to you to decide when you need outside aids. And although you can write a computer program to solve some of the problems, you have to decide whether the programming effort is "worth it," i.e., would it take less total time to solve it by pencil, paper, and human brain than with the assist of a machine.

I assembled this collection of puzzles and problems from about 20 sources which are cited at the end, plus a pile of original puzzles which I've been writing and accumulating in a dog-eared folder for years. If you like this kind of stuff, let me know, and we'll run more.

Mathematical Puzzles and Pastimes, Philip Haber (Ed.); The Peter Pauper Press, Mt. Vernon, NY; 1957. $1.95.

Pencil Puzzles & Word Games, 5th Edition, Dell Publishing, New York; 1975. 35c.

The Math Entertainer, Philip Heafford, Emerson Books, Buchanan, New York; 1959. $5.95.

150 Puzzles in Crypt Arithmetic, Maxey Brooke, Dover Publications, New York; 1963. $1.25. (Available thru *Creative Computing* Library.)

101 Puzzles in Thought and Logic, C. R. Wylie, Jr., Dover Publications, New York; 1957. $1.35.

Invitation to Mathematics, W. H. Glenn and D. A. Johnson, Dover Publications, New York; 1960. $3.50.

Pillow Problems and a Tangled Tale, Lewis Carroll, Dover Publications, New York; 1958. $1.50.

Fun With Mathematics, Jerome S. Meyer, World Publishing Co., Cleveland; 1952.

Mathematical Puzzles for Beginners & Enthusiasts, Geoffrey Mott-Smith, Dover Publications, New York; 1954. $1.75.

Computer Programming Problems, NCC Publications, U.K.; 1975. $0.75.

Gamesmag (Vol. 1, No. 1), Center for Open Learning, Box 9434, Berkeley, CA; 1975. $0.75.

Games & Puzzles, 11 Tottenham Court Road, London W1A 4XF, England. One-year subscription to USA $11.40. (One of the very best games magazines!)

The Mathematical Puzzles of Sam Lloyd, Vol. II, Martin Gardner (Ed.), Dover Publications, New York.

The Floodgate Scandal and Other Puzzles, Ivan Morris. (Preprint of a forthcoming book.)

Computer Problem Solving, R. P. Watkins, John Wiley (Australasia); 1975.

15 pages of puzzles

SPEED TRAP

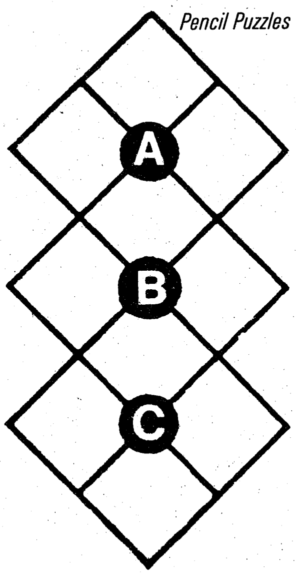
One day as a motorcycle cop was watching the street for speeders, he spied one coming down at breakneck speed about ¼ mile away. If the cycle covered 55 miles in an hour and caught up to the car in 300 seconds, how fast was the car traveling?

Mathematical Puzzles and Pastimes

NUMBER DIAMONDS

Can you distribute the numbers 1 through 10 (using each number only once) in the blank spaces in such a way that the numbers surrounding each letter—A, B, or C—will total 20? Several different combinations are possible.

Pencil Puzzles & Word Games



HOW LONG, MONET?

A water lily doubles itself in size each day. From the time its first leaf appeared to the time when the surface of the pond was completely covered took forty days. How long did it take for the pond to be half covered?

The Math Entertainer

GROOVY

The diameter of a long-playing record is 12 inches. The unused center has a diameter of 4 inches and there is a smooth outer edge 1 inch wide around the recording. If there are 91 grooves to the inch, how far does the needle move during the actual playing of the recording?

The Math Entertainer

ELECTRONIC LULLABY

Our small neighbor was given an electronics set for Christmas and we have had no peace since. His latest model is an electronic organ. Unfortunately, it only plays three notes, a high note, ping; a middle note, mmmmmm; and a low note, boing. He has wired these up so that the same note repeated; for example, ping ping is immediately followed by an mmmmmm. A note followed by a lower note is followed by ping, and a note preceding a higher note leads to a boing. Really quite impressive for a twelve-year-old, but it's getting on our nerves. Can you explain why?

Games & Puzzles

"ABRACADABRA"

An old numerologist from the Left Bank reports a secret Grecian formula for finding prime numbers (numbers divisible by nothing but 1 and themselves, like 7, 17, 19): $W^2 - W + 41 = P$.

For every whole number, W, P is prime. Correct?

WIRED UP

A telegraph wire is laid around the surface of the moon at its equator. A second telegraph wire is laid around the surface of the planet Mars at its equator. It is now required to raise each of these wires six feet above the equators. Clearly, in each case, more wire is required to do this, but how much more for each of the two bodies?

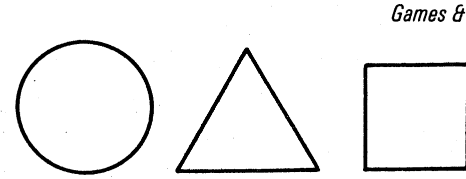
You may assume that Mars and the moon are both perfect spheres, having respective diameters of 4140 miles and 2160 miles. You may also assume that when the wire is raised six feet off the surfaces concerned, it will not sag between whatever supporting poles there are.

Games & Puzzles

SQUARE ROTATION

This circle, square and triangle each have the same length perimeter. If the circle is rotated without slipping around the other two, in which case will it rotate through the largest number of degrees?

Games & Puzzles



FLOWER CHILDREN

If five girls pack five boxes of flowers in five minutes, how many girls are required to pack fifty boxes in fifty minutes?

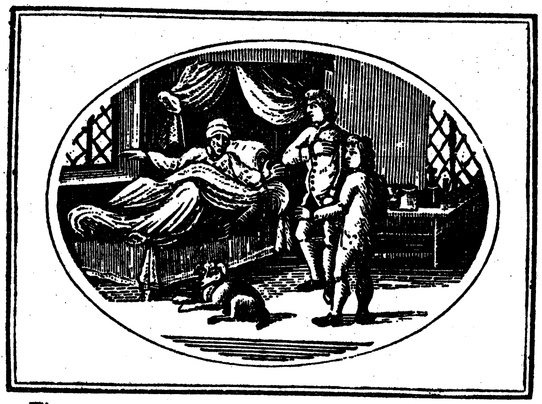
The Math Entertainer

IMPOSSIBLE DIVISION

Aunt Jenny had 3 greedy nephews—Philip, Sam and George, who eagerly looked forward to the day when she would die and leave her money to them. But Aunt Jenny decided to play them a little trick. She called in her lawyer one day and made out her will as follows:

The total estate amounting to $1,717 is to be shared by the nephews as follows:

Philip is to receive ½, Sam ⅓ and George 1/9, with the proviso that each is to receive an amount in even dollars only, according to his share. Each nephew is to have 24 hours counting from the hour of death, to calculate the exact amount in dollars he is to receive. If in the event any calculated share amounts to dollars and cents, or if no exact amount in dollars is arrived at at the expiration period of 24 hours, the whole sum becomes forfeited and is to be bequeathed to a worthy charity designated under Paragraph 7 of said will.



The very next day, Aunt Jenny passed away and her lawyer called in her three nephews to hear him read the will. At the end of the reading, they started to calculate their shares, but to their consternation found that no matter how they figured their shares they could not make them come out in an even amount in dollars.

At the end of 23 hours and 50 minutes, they were desperate, but in the last 10 minutes all went well. How did they solve their problem?

Mathematical Puzzles and Pastimes

ALL NINES

What six digit number added to itself five times will give a total each time having the same digits as that number but differently arranged, and after the sixth addition will give a total of all nines?

FREDDY THE FROG

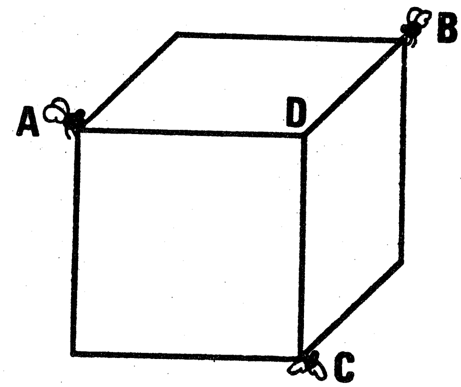
Freddy the Frog falls into a 99-foot well with a great splash, and at once starts climbing to the top. He goes up three feet every day and falls back two each night. Freddy continues in this indomitable fashion until he has reached the very top of the well. How long has it taken him?

Floodgate Scandal

FLIES ON CUBE

Three flies start at three corners of a cube, at A, B and C. The starting-gun fires and each starts crawling towards one of the others, all moving at identical speeds. A moves towards B, B to C and C to A. Each moves in the direction which would take it to its target in the shortest possible time *if* that target were stationary. Presumably they will end up arriving simultaneously at D, but in getting there will cross the edges of the cube, and if so, how many times?

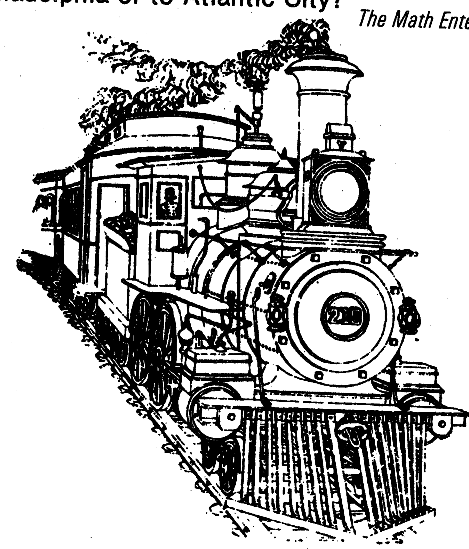
Games & Puzzles



JERSEY CENTRAL FLYER

From Philadelphia to Atlantic City is 60 miles. Two trains leave at 10:00 A.M., one train from Philadelphia at 40 miles an hour and the other from Atlantic City at 50 miles an hour. When they meet, are they nearer to Philadelphia or to Atlantic City?

The Math Entertainer



A CURIOUS PROPERTY

The number 142857 has many remarkable features. Here is one of the least known: 142857^2 = 20408122449 and 142857 = 20408 + 122449! There are four numbers of three figures with the same property. Square the number, add the number formed by the last three digits to the remaining number and the original number appears. One is the trivial 001; slightly less trivial is 999. And your puzzle is to find the other two. It may encourage you to know that their sum is exactly 1000.

Games & Puzzles

CRYPTOGRAMS

In these cryptograms, another letter of the alphabet has been substituted for the right letter. By noticing the frequency of certain letters (e, a, o, and n are usually the most often-used letters in English), and by looking for repeated patterns of letters in the words, you should be able to break each code. Each cryptogram is in a different code.

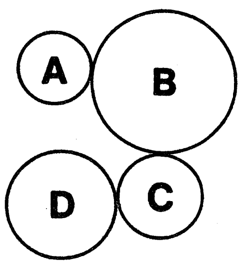
1. CTW EYDW GXH
FC PW PCDSTV
SE FC FWUU
XUU HCY RTCG.

2. BOC ALWZG
KJZU J WLRM
GILPC GOLPI:
IOUC BJRI IL
IUWW JRLIOUP.

Pencil Puzzles & Word Games

A NON-SLIPPERY PROBLEM

The figure shows four rollers which roll against each other, in sequence, without slipping. The diameters of A and D are 4 inches and 6 inches respectively. B is twice the diameter of C and the total diameters of B and C together are equal to four times the diameter of A. If A is rotating at 3 revolutions per second, how fast will D rotate? *Games & Puzzles*



CHARMER

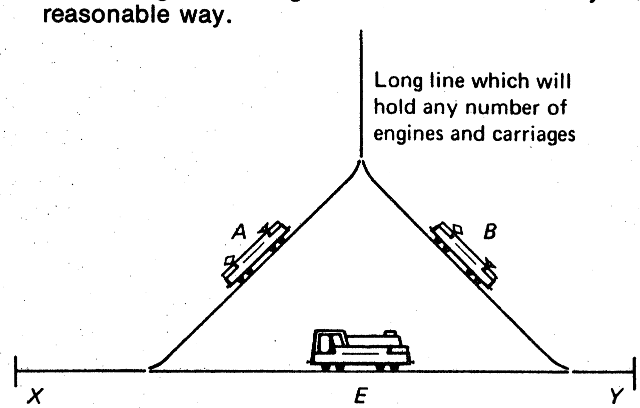
In a remote country village there lived a poor woman who found it necessary to cross a certain bridge every day in order to earn her livelihood in the next village. One day, as she was about to cross, she was approached by a stranger who promised her riches if she followed his directions. "Take this charm," he said, "and you'll find that each time you cross, your money will be doubled. At the end of each day I'll be waiting for you to pay me ½ the money you then possess. But the charm will bring you this luck for only 3 days. Then you must return it to me or ill-luck will befall you."

After 3 days, the stranger had collected 14 times as much money as the woman had had originally, while she came out $7.00 richer.

Could you find out how much the woman had originally and the amount the stranger was able to collect? *Mathematical Puzzles and Pastimes*

SWITCH THE CARS

There are two railway carriages and an engine on the network of lines and sidings as shown below. As with all railway line intersections, points are used. Trains cannot go round corners, so they must move into a siding and then come out of it into the new track in the same order. The siding labelled *X* can hold either two carriages or a carriage and an engine, and nothing more. The siding labelled *Y* can only hold a carriage (an engine will not fit on it). Devise a method for shifting the engine and carriages around so that the carriages labelled *A* and *B* are swapped over and the engine *E* is returned to its original position. The engine only has a motor, and the carriages and engine can be linked in any reasonable way.



SANDAL SALES

A town in India has a population of 20,000 people. 5 percent of them are one-legged, and half the others go barefoot. How many sandals are worn in the town? *The Math Entertainer*

M.C.P.

A boy is chosen president and a girl vice-president of the senior class of a school. In how many ways is this possible if the class has twelve boys and ten girls? *The Math Entertainer*

SLUM HOUSING

A landlord owns a multiple dwelling housing project consisting of 2-room, 3-room, 4-room and 5-room apartments, renting for $60.00, $90.00, $110.00 and $120.00 respectively, per month. Altogether he has 100 apartments for which he receives $10,500 monthly from his tenants.

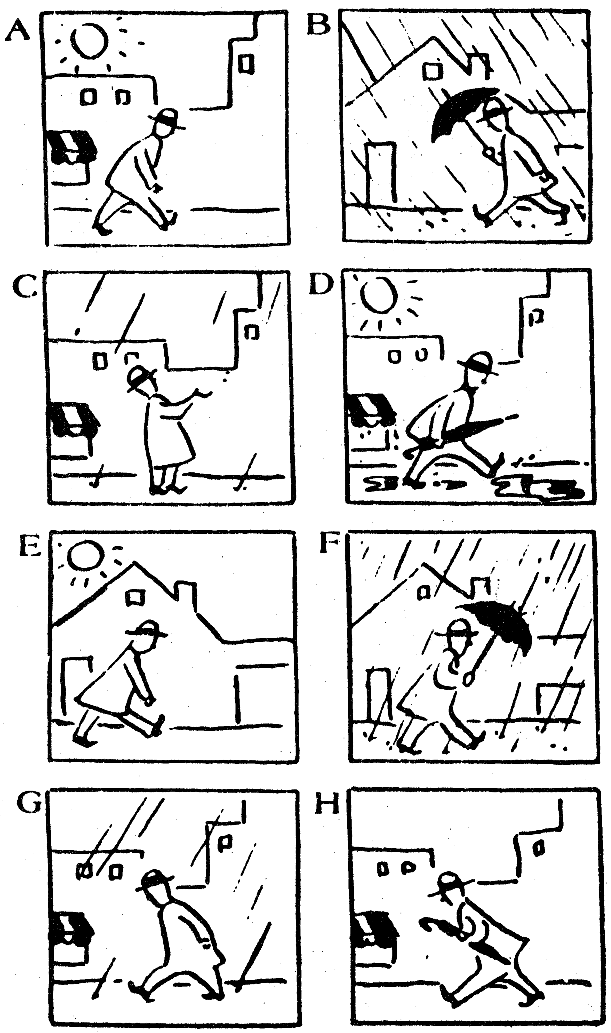
Find the number of apartments he can rent of each type. *Mathematical Puzzles and Pastimes*

SEQUENCE TEST

Study the series of pictures below. Then, starting with picture E, see if you can arrange them in the proper order so that they will present a logical sequence of events.

If you complete this test correctly, you rate very high in combining your power of observation with your power of deduction.

Pencil Puzzles & Word Games



ARTFUL ARITHMETIC

Johnny is not very keen on mathematics so he jumped at what seemed to him to be a quick way of discovering which of two fractions is the larger. Asked to find, for example, the larger of 2/5 and 3/7 he simply replaced them by 2/3 (2/5-2) and 3/4 (3/7-3) respectively, which he immediately replaced by 2/1 and 3/1 concluding triumphantly that the first, 2/5 is the smaller.

The teacher's problem, and yours, is of course to decide whether Johnny's method is valid, or whether it is nonsense, and his correct answer to this particular problem only a lucky fluke.

Games & Puzzles

HMMMM

$8^m = 32$. Find m.

SEVEN PAIRS

If two 1's, two 2's, and two 3's are arranged thus:

2 3 1 2 1 3

then the two 1's enclose 1 other digit, the two 2's enclose 2 other digits, and the two 3's enclose 3 other digits. Can you find a similar arrangement using the seven pairs 1,1; 2,2; . . . 7,7? Counting reflections, there are 52 different solutions, so it shouldn't be too hard finding one!

Games & Puzzles

MONEY, MONEY

One day Phil found to his dismay that he was short in funds by a certain amount. So he wired his sister Amy, as follows:

```
$  W I . R E
   M O . R E
$M O N . E Y
```

How much should Amy send Phil?

Mathematical Puzzles and Pastimes

THE BLETCHLEY LAWN TENNIS CLUB

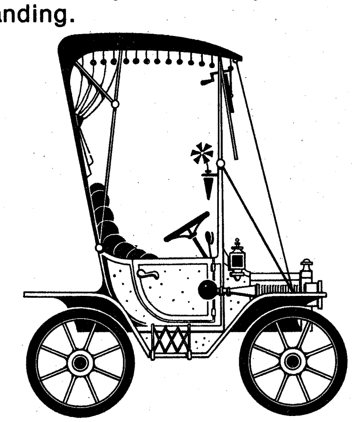
A total of 111 players belong to the Bletchley Lawn Tennis Club. What is the smallest number of games (all singles) necessary to determine who is the best player in the club? (Time limit: 30 seconds.)

Floodgate Scandal

A PRIME DEAL

Buying a new car is no joke, but my dealer was very understanding.

```
 WE'LL
 HE LP
 WI TH
     A
 -----
 DE AL
```



he told me, 'And it's a prime deal, the best deal you could possible get.' What was the deal?

Games & Puzzles

PROBLEM FOR MA BELL

If the first three letters of a telephone number indicate the name of the exchange, how many such arrangements of three letters is it possible to devise from the twenty-six letters of the alphabet?

(*Afterthought: Without looking at a telephone, what two letters are not used on an actual dial?*)

The Math Entertainer

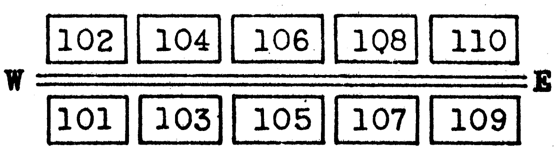
THE 100-BLOCK GANG

Bob and four other youngsters who live in the 100 block of Center Street have formed a group known as "The 100-Block Gang." Center Street runs east-west; numbers start at the west end of the block, with odd numbers on the south side of the street. There are five houses on each side of the block, so the numbers run from 101 to 110. From the following clues about the five and their families, you should be able to decide the full name and address of each member of the group.

1. The Baileys and the Prices live directly opposite each other in the center of the block.
2. John lives directly between Ethel and the Greens.
3. Vera lives directly across the street from Martha, and next to the Baileys.
4. The Berrys live immediately east of the Prices; there are no young people directly across from the Berrys.
5. The Golds live on the south side of the street, and John on the north side.

For this problem; we have presented a diagram of the 100 Block. We found it to be more of a solving aid than a regular chart would be.

Pencil Puzzles & Word Games



DOUBLE OR TAKE

This is a game for two players for which you need at the very most, pencil and paper. One player names a number. His opponent has a choice between doubling it, or taking from it a perfect square or a perfect cube. The first player plays again in the same manner and so on alternately, until one player on his turn reaches zero. He is the winner.

Some numbers are obviously losing. For example, left with 2, you have a choice between doubling to 4, or taking 1, leaving 1. Either way he will subtract a square and win immediately. Similarly 5 is a loss. If you double, take 1, take 4, your opponent will take 8 ($=2^3$), take 4 or take 1 respectively. Now suppose you are left with 21. Should you win or lose? If you are left with 17?

Games & Puzzles

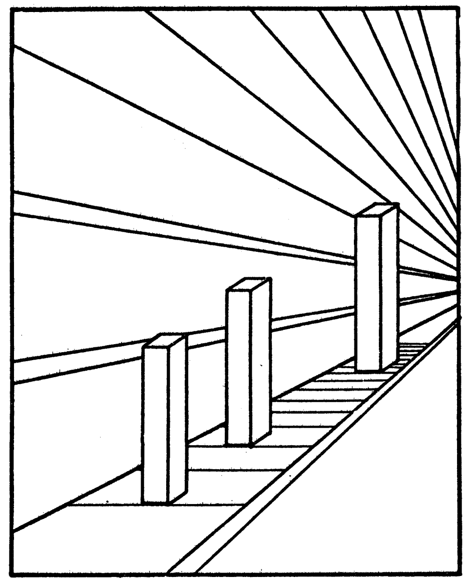
NUMBER COMBINATIONS

(A) Use three sixes to make 7.
(B) Use three fours to make 11.
(C) Use three threes to make 24.
(D) Use three fives to make 5.
(E) Use three fives to make 10.

Floodgate Scandal

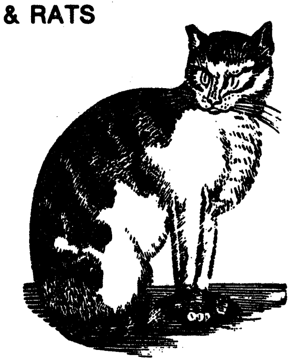
NEXT?

What is the next number in this series:
92, 74, 46, 22, 18, . . .



The large oblong in rear is the same size as the others.

CATS & RATS

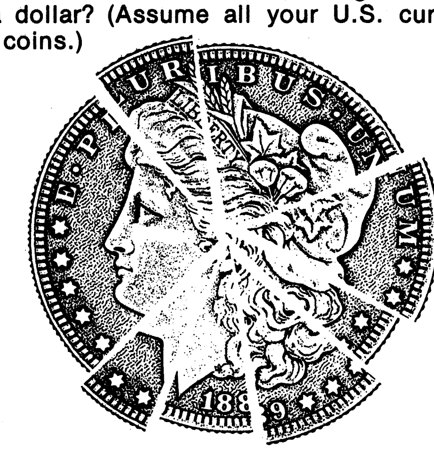


If 6 felines can devour 6 rodents in 1/10 of an hour, how many would it take to devour 100 rodents in 6,000 seconds?

Mathematical Puzzles and Pastimes

CHANGE FOR A DOLLAR

What is the largest amount of money you can have and still be unable to give change for a dollar? (Assume all your U.S. currency is in coins.)



ZILCH—MILCH—PILCH

If 1 zilch is equal to 13 milches, and 1 milch is equal to 23 pilches, would you accept 8,000 pilches for 26 zilches?

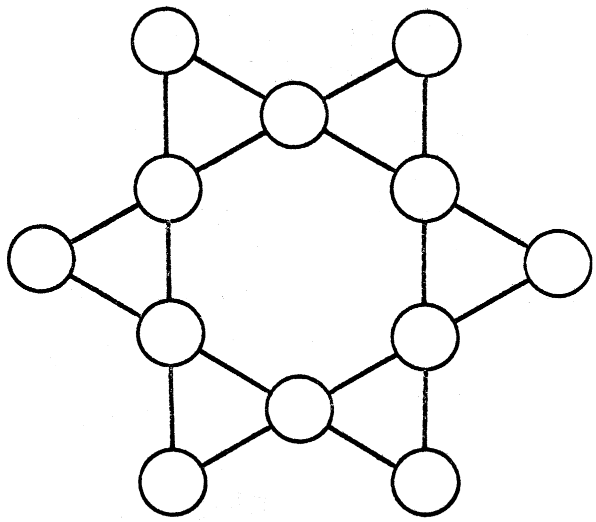
AND COMPUTER SCIENCE?

Among one hundred applicants for a certain technical position it was discovered that ten had never taken a course in chemistry or in physics. Seventy-five had taken at least one course in chemistry. Eighty-three had taken at least one course in physics.

How many of the applicants had haa some work in both chemistry and physics?

101 Puzzles in Thought and Logic

MAGIC HEXAGON



Can you insert the numbers 1 to 12 in the diagram above so that the sum of the four numbers along each line totals 26? If you manage that successfully, now try it with the added restriction that the six points of the hexagon must add to 26 also!

Games & Puzzles

TRUTH & FALSEHOOD

In a faraway land there dwelt two races. The Ananias were inveterate liars, while the Diogenes were unfailingly veracious. Once upon a time a stranger visited the land, and on meeting a party of three inhabitants inquired to what race they belonged. The first murmured something that the stranger did not catch. The second remarked, "He said he was an Anania." The third said to the second, "You're a liar!" Now the question is, of what race was this third man?

Mathematical Puzzles

HALVING NUMBERS

When is one half of ten equal to five, but one half of nine equals four, one half of eleven is six and one half of twelve, seven?

Games & Puzzles

ODD ARRANGEMENT

Arrange the nine digits so that the first three shall be ⅓ of the last three; and the central three equal to the difference between the first three and the last three.

Numbers shall be as they are arranged, not totalled.

TELL ME WHY, DAD

Sometimes little boys who ask too many questions tie their parents into mental knots. Take, for example, the young brat who asked his father (a professor of philosophy) if God could do anything. Of course the father said "Yes." "Then can he make a stone so large that he can't roll it?" asked junior. The professor was about to say "Yes" again but remained silent.

Fun With Mathematics

TOO MUCH BEER?

The following is a portion of a report submitted by an investigator for a well-known market analysis agency with standards of accuracy so high that it boasts that an employee's first mistake is his last.

Number of consumers interviewed	100
Number who drink coffee	78
Number who drink tea	71
Number who drink both tea and coffee .	48

Why was the interviewer discharged?

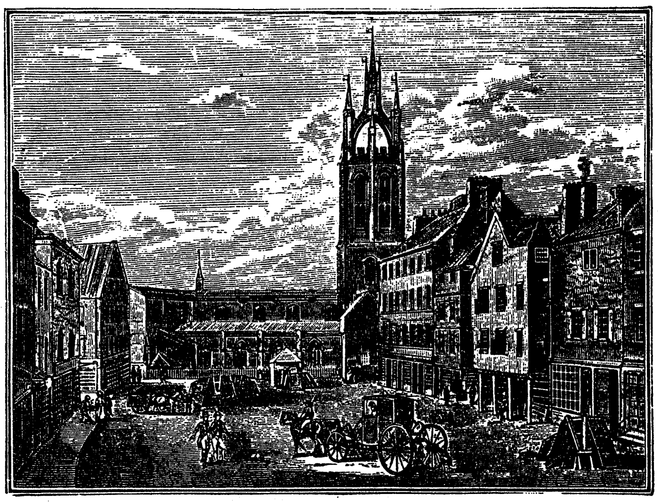
101 Puzzles in Thought and Logic

OLYMPIC WALKERS

A and *B* begin, at 6 a.m. on the same day, to walk along a road in the same direction, *B* having a start of 14 miles, and each walking from 6 a.m. to 6 p.m. daily. *A* walks 10 miles, at a uniform pace, the first day, 9 the second, 8 the third, and so on: *B* walks 2 miles, at a uniform pace, the first day, 4 the second, 6 the third, and so on. When and where are they together?

Pillow Problems

A SMALL TOWN AFFAIR



There is a small town of a few hundred inhabitants of which the following statements are, surprisingly, true:

1) Every man in the town is a perfect logician and is aware that this is true of every other man in the town.
2) Every man in the town knows *all* about the behavior of every woman in the town, with the exception, if he is married, of his own wife. It is taboo for anyone to speak about a woman to her husband.
3) It is an immutable custom (abhorrent to us maybe, but as inevitable as night following day to them) that, when a man discovers that his wife has been unfaithful, he takes her out into the town square that same night, and, on the stroke of midnight, shoots her.
4) There are 40 unfaithful wives in town.

Now, life has been continuing its uneventful course for some time when, one fateful summer's day, June 1st actually, the Mayor summons all the townsmen to a meeting in the town hall. 'I am very sorry to have to tell you this,' he says, 'but there is an unfaithful wife in this town.' The meeting ends and the men disperse. What, if anything, happens, and when? (*Not* an easy problem!)

Games & Puzzles

ONE BAG OR TWO?

There are two bags, one containing a counter, known to be either white or black; the other containing 1 white and 2 black. A white is put into the first, the bag shaken, and a counter drawn out, which proves to be white. Which course will now give the best chance of drawing a white—to draw from one of the two bags without knowing which it is, or to empty one bag into the other and then draw?

Pillow Problems

VERY ODD

There are 8 consecutive odd numbers and when they are multiplied by each other you get 34,459,425.

You'll find them—if you try hard enough.

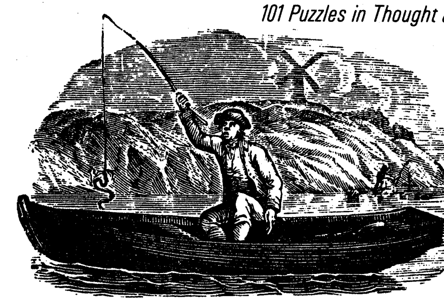
Mathematical Puzzles and Pastimes

PERSISTENT PADDLER

A woodsman paddling steadily across the still surface of a northern lake saw a magnificent bass break water directly ahead of him. Twelve strokes he counted until his canoe first crossed the ever-widening circle the fish had made, and then twelve more before he broke through the circle on the opposite side. For a time thereafter he sought relief from the pleasant monotony of his journey by calculating how far away the fish had been at the moment it jumped, but it proved too much for him and he soon gave himself up to less specific thoughts.

Can you complete his calculation?

101 Puzzles in Thought and Logic



TYPEWRITER TWISTER

For this puzzle you need to know your typewriter keyboard very well, or otherwise have a typewriter in front of you. Each letter is coded by the row it is in and its position in the row. Thus Q which is the first letter in the first row will be 11, and P, the last letter in that line and the tenth will be 110. D the third letter in the second row will be 23.

A certain word is coded in this manner and then each number from 1 to 10 is replaced by a different number from 1 to 10. Finally the resulting number pairs are converted back, using the same code reversed, into letters. The letters in sequence are: SMFQNEHUX. What was the original word?

Games & Puzzles

SKEW ADDITION

In this addition sum, the powers of 3 have been written under each other, but with each unit digit displaced one to the right compared to the preceding number. Assuming that the powers of 3 are added up forever in this manner, what will the answer be?

Games & Puzzles

```
1
 3
  9
  27
   81
   243
    729
   2187
     6561
    19683
```

NONE GIVEN

If the third is JAS and the fourth is OND, what are the first two?

SEE-SAW

Nine boys, none of whom weighed more than 100 pounds or less than 50 pounds, experimented on a see-saw and found the following lucky thirteen facts:

A. The see-saw balanced because the total weight at each end was exactly equal:
   1. Art and Don with George and Henry
   2. Art and Chuck with Don, George and Ira

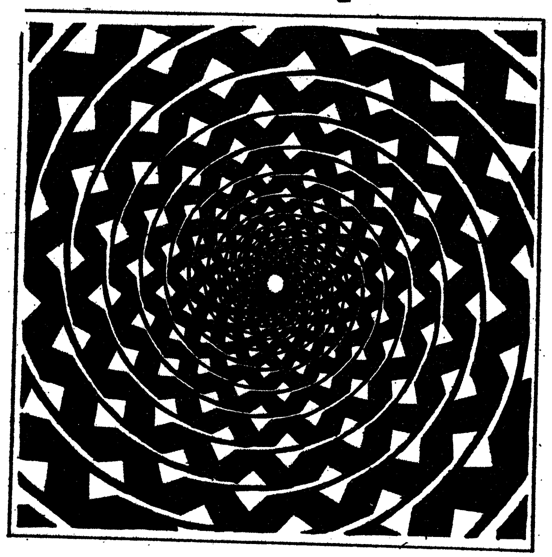
B. These groups were excellent, the differences in weight at the two ends being no more than ten pounds:
   3. Bob and Chuck with Ed and Fred
   4. Chuck and George with Ed and Henry
   5. Don and George with Ed
   6. Art, George & Henry with Chuck and Ed

C. These groups were quite unbalanced, the differences in weight at the two ends being 25 pounds or more:
   7. Ed with Fred
   8. Art with Don
   9. Don with Fred
   10. Chuck with Ira
   11. Bob and Don with Henry and Ira
   12. Chuck and Henry with Fred and Ira
   13. Art, Chuck and Henry with Don, Ed and Fred

How many pounds does each boy weigh? Thinking caps will be required for this one.

Games & Puzzles

Circle or Spiral ?



THREE BAGS

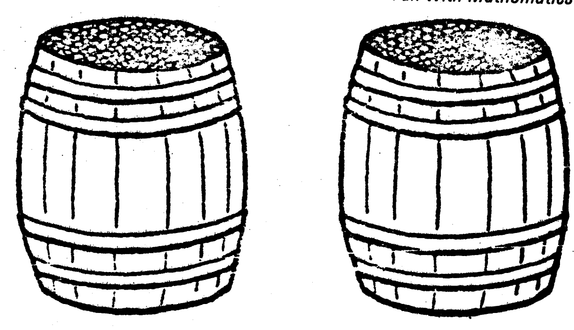
There are 3 bags; one containing a white counter and a black one, another two white and a black, and the third 3 white and a black. It is not known in what order the bags are placed. A white counter is drawn from one of them, and a black from another. What is the chance of drawing a white counter from the remaining bag?

Pillow Problems

BARREL OF FUN

If a millionaire offered you your choice between a barrel filled with half dollars and the same barrel filled with dimes, which would you choose?

Fun With Mathematics



TENNIS AT HILLCREST

Eight men entered the recent tennis tournament at Hillcrest. The tournament was played in three consecutive days, one round per day, and happily no match was defaulted. The first and second round matches were stipulated to be 2 sets out of 3, while the final was 3 sets our of 5. A spectator who was present on all three days reports the following facts:

1. Eggleston never met Haverford.
2. Before play began, Gormley remarked jocularly to Bancroft, "I see that we meet in the finals."
3. Chadwick won a set at love but lost his first match.
4. Altogether 140 games were played, of which the losers won 43.
5. When the pairings were posted, Abercrombie said to Devereaux, "Do you concede, or do you want to play it out?"
6. On the second day, the first-round losers played bridge, and the same table gathered on the third day with Eggleston in place of Abercrombie.
7. Bancroft won 9 games.
8. Franklin won 37 games.
9. The first score of the tournament was a service ace by Gormley, at which Eggleston shouted "Hey, I'm not over there!"

Who won the tournament? Whom did he beat and by what score?

Mathematical Puzzles

THREE COINS

Show the different ways that three coins turn up. There are eight ways in all.

a. What is the chance that the three coins will turn up all heads?
b. What is the chance that the three coins will turn up two heads and one tail?

Invitation to Mathematics

GIGO

What is wrong with the conclusions based on the data given in problems 1 through 4?

1. How many days do you go to school?

	Days
Days in a year:	365
You sleep at least 8 hours per day or 1/3 of the year:	− 122
This leaves:	243
You have 52 Saturdays and 52 Sundays off:	− 104
This leaves:	139
You have summer vacation for three months:	− 90
This leaves:	49
You have Christmas and Easter vacations:	− 19
This leaves:	30
And you spend at least 2 hours each day eating:	− 30
Days left to go to school:	0

2. More people were killed in airplane accidents in 1960 than in 1928. Therefore, it was more dangerous to ride an airplane in 1960 than in 1928.

3. Checkered cows produce 26 percent more milk than other cows. Therefore, checkered cows are the best milkers.

4. There are fewer accidents in France than in Germany. Therefore, it is safer to drive a car in France than in Germany.

Invitation to Mathematics

KRYPTIC KIDS

There are eight brothers and sisters in a family: Georgette, Ulysse, Yvon, Marie, Annette, Roger, Isadore, and Emile. All are less than ten years old. If you represent the age of each child by the first letter of its name, you obtain:

```
           YU
     GUY )MARIE
          MARE
          ----
            EE
```

If Marie is the youngest of the sisters, what is her age?

150 Puzzles in Crypt-Arithmetic

SIMPLE MULTIPLICATION

Let us take any two numbers + fractions and multiply them together. Then let us take the same two numbers + the equivalent decimals and multiply them again. Naturally you would expect the two answers to be the same. But let's see:

16½	16.5
12½	12.5
32	8.25
16	33.0
8¼ (half of 16½)	165
6¼ (half of 12½)	Answer: 206.25
Answer: 206½	

What has happened to the other ¼?

Fun With Mathematics

COUNTERWEIGHTS

The large flats and other pieces of scenery used in a vaudeville theater are counterweighted by sandbags, so that when they are moved only a small portion of the weight has to be borne by the stagehands.

The theater keeps on hand a set of metal counterweights for occasional use with special pieces. Any or all of the weights can be attached quickly to an elevator rope. There are five weights in the set, so arranged that it is possible to compound any load which is a multiple of 10, from 10 pounds up to the total of all five weights together. The choice of weights is such as to reach the maximum possible total load. What are the several weights?

Mathematical Puzzles

CHINESE MENU

How many guests were present at a Chinese party if every two used a dish for rice between them, every three a dish for broth, every four a dish for meat, and there were 65 dishes altogether?

The Math Entertainer

LIFT SIMULATION

Write a program to simulate a lift, capable of stopping at 4 floors. The lift is controlled by 2 buttons on the middle 2 floors (requesting a lift up and a lift down), a single button on the top and bottom floors and a set of 4 buttons in the lift itself corresponding to the 4 floors.

The lift is controlled in such a way that the buttons within the lift take precedence over those on each of the floors. However, the lift can be made to stop at a floor as a result of a request there, if the lift is passing thru that floor in the right direction or alternatively if there is no request from within the lift itself.

Requests for the lift are queued and, apart from the provisos given above, are dealt with on a first come first served basis.

The program simulating the lift allows the user to press any of the buttons (by typing at the teleprinter), just before it is about to reach a floor, whether or not it is due to stop there. If there are no requests at all, the program terminates. Output from the program should indicate where the lift is, the direction it is moving and where it stops.

Computer Programming Problems

BATTER UP

How many possible batting orders are there for a baseball team of 9 players?

WEIGHT WATCHERS

The fat men in a club outnumber the thin men by sixteen. Seven times the number of fat men exceeds nine times the number of thin men by thirty-two. Find the number of fat and thin men in the club.

The Math Entertainer

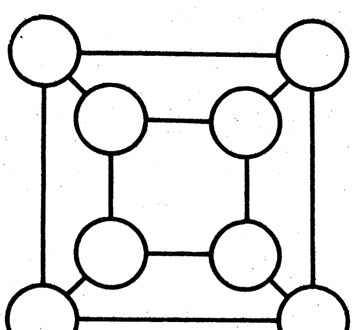
SEESAW

Three brothers go to a playground to play on the seesaw. The teeter board has a fixed seat at each end, 5 feet away from the trestle on which the board swings. When Alfred and Bobby take seats, Charles, who weighs 80 pounds, balances them by sitting on Alfred's side 21 inches away from the trestle. When Charles sits on a seat, it takes both his brothers to balance him, Alfred in the other seat and Bobby one foot nearer the center. Now if Bobby takes Alfred's place, where must Alfred sit to balance Charles?

Mathematical Puzzles

SQUARE EIGHTEEN

How can the numbers 1 to 8 be placed, one in each of these circles, so that the sums of the numbers at the corners of each of the five areas, and the sum of the four numbers on the outside are all equal to 18 and the 7 and 8 are not in two corners of the same area or square?



Games & Puzzles

FAST FREIGHT

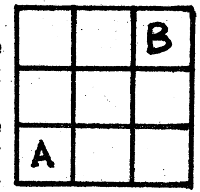
If it takes twice as long for a passenger train to pass a freight train after it first overtakes it as it takes the two trains to pass when going in opposite directions.

How many times faster than the freight train is the passenger train?

101 Puzzles in Thought and Logic

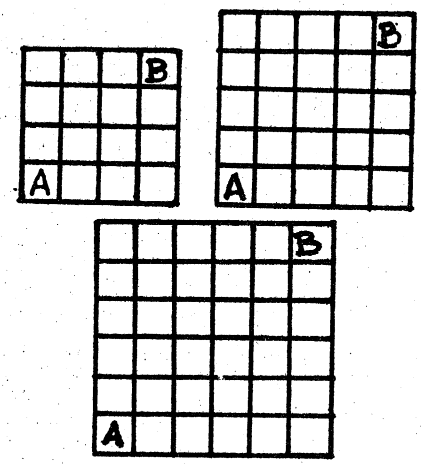
A SIMPLE PUZZLE

Try this set of simple problems with your students and see if they can draw general conclusions about how the problem would be solved on different game boards. Draw a 3" by 3" board with A and B marked as follows:



Put a penny on A. What is the smallest number of moves it would take the penny to go from A to B if it could move 1 square either horizontally or vertically per turn? The answer which is 4 is pretty easy to discover. Here's a slightly more difficult question however: How many different paths are there that takes 4 moves to get from A to B? The answer is 8.

Now try to answer the same questions on a 4" x 4", 5" x 5" and a 6" x 6" board.



Now fill in the following chart:

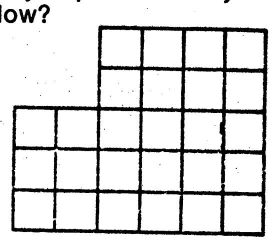
Board	*Smallest number of moves from A to B*	*Number of smallest paths*
3x3	4	8
4x4		
5x5		
6x6		

Do you see any patterns? Can you describe them in simple words? Can you also describe them in mathematical symbols?

Gamesmag

SQUARES

How many squares can you find in the diagram below?



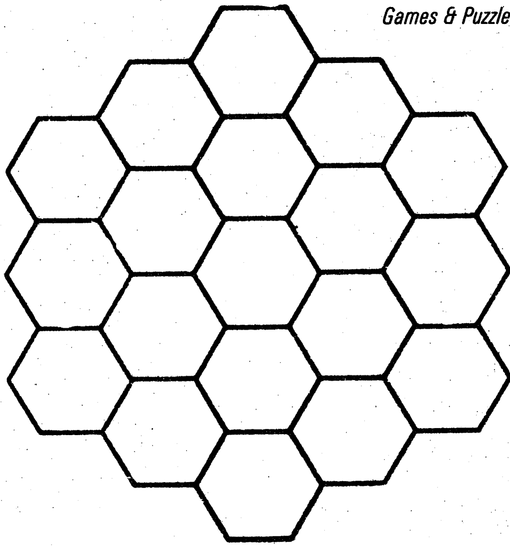
LOGIC ONLY

$$\begin{array}{r} A\,B\,C\,D\,E \\ \times\ 4 \\ \hline E\,D\,C\,B\,A \end{array}$$

1 TO 19

Can you place the numbers 1 to 19 in the honeycomb's nineteen cells in such a way that there is a difference of at least 4 between any one cell and its neighboring cells?

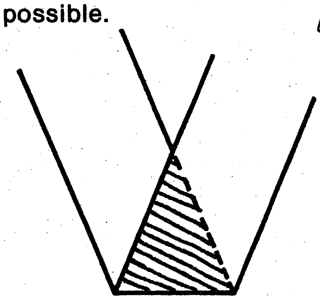
Games & Puzzles



A DOUBLE STRIP

A strip of paper is folded once, as in the figure. At what angle must it be folded so that the double thickness area (shaded) is as small as possible.

Games & Puzzles



ALPHABET SOUP

The word "crypt-arithmetic" was first introduced by M. Vatriquant writing under the pseudonym "Minos". In the May, 1931 issue of *Sphinx,* a Belgian magazine of recreational mathematics, he proposed this problem with these remarks:

"Cryptographers, to hide the meaning of messages, put figures in places of letters. By way of reprisal, we will replace each digit of the following problem with a distinct letter."

```
   A B C
 × D E
   -----
   F E C
 D E C
 -------
 H G B C
```

150 Puzzles in Crypt-Arithmetic

Although the word was new, the type of puzzle was older. The earliest one in my collection is from the *Strand Magazine* for July, 1924:

TWO X TWO = THREE

ADJACENT SQUARES

49 and 169 are both squares. They have something else in common—they can both be formed by placing two squares adjacent to each other, using 4 and 9, and 16 and 9 respectively.

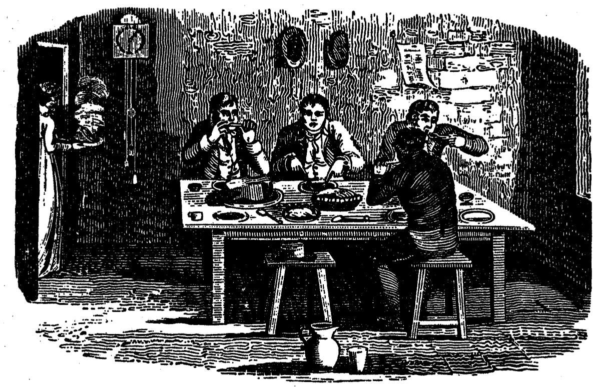
What are the next two squares with this property?

Games & Puzzles

SIX BOYS

The same six boys are to sit around a table for lunch. How many different arrangements can be made of the order in which they are to sit?

The Math Entertainer



AND WHO'S THE TRACK COACH?

In the Stillwater High School the economics, English, French, history, Latin, and mathematics classes are taught, though not necessarily respectively, by Mrs. Arthur, Miss Bascomb, Mrs. Conroy, Mr. Duval, Mr. Eggleston, and Mr. Furness.

The mathematics teacher and the Latin teacher were roommates in college.

Eggleston is older than Furness but has not taught as long as the economics teacher.

As students, Mrs. Arthur and Miss Bascomb attended one high school while the others attended a different high school.

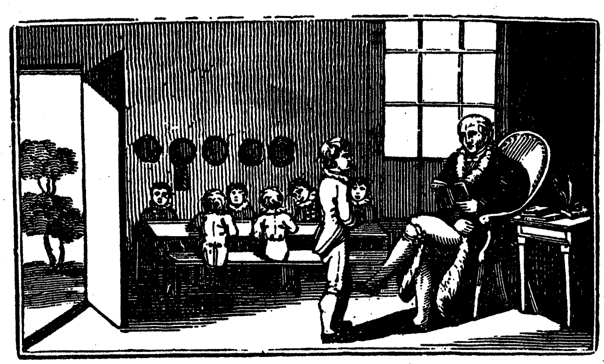
Furness is the French teacher's father.

The English teacher is the oldest of the six both in age and in years of service. In fact he had the mathematics teacher and the history teacher in class when they were students in the Stillwater High School.

Mrs. Arthur is older than the Latin teacher.

What subject does each person teach?

101 Puzzles in Thought and Logic



SANTA CLAUS

If the value of CHRISTMAS is 110, what is the most likely value of NEW YEAR?

LAST DIGIT ONLY

What is the last digit in 7^{1000}? Try this in BASIC (remember: not the entire number, just the last digit).

SHIRT SALE

"I made a smart move marking down those shirts from $2.00," remarked Mr. Gaberdine to his wife. "We have disposed of the entire lot."

"Good!" said Mrs. Gaberdine. "How much profit did you made?"

"We haven't figured it yet, but the gross from the sale was $603.77."

"Well, how many shirts did you sell?"

Let the reader answer the question.

Mathematical Puzzles

NOT QUITE USELESS

The well-known firm of Shyster, Shyster, Shyster, Shyster & Sons is in trouble. The calculating machines they use to work out their fees are badly broken down. One will only add, another will only subtract, the third will only multiply, and the fourth will do nothing but divide. Which machine will be of most use to the poor clerks, who are totally useless at mental arithmetic?

Games & Puzzles

NEXT IN SEQUENCE PLEASE

1. Ten letters—what is the eleventh?

N W H O I I E I I E ?

2. The answer to this one is far out, outa sight, just too much.

M V E M J S U N ?

Games & Puzzles

THE THREE BEGGARS

A charitable lady met a poor man to whom she gave one cent more than half of the money she had in her purse. The poor fellow, who was a member of the United Mendicants' Association, managed, while tendering his thanks, to chalk the organization's sign of "a good thing" on her clothing. As a result, she met many objects of charity as she proceeded on her journey.

To the second applicant she gave two cents more than half of what she had left. To the third beggar she gave three cents more than half of the remainder. She now had one penny left.

How much money did she have when she started out?

Puzzles of Sam Lloyd

MARK TWAIN'S SUGGESTION

Is it true (as Mark Twain once suggested) that with two dice one is twice as likely to throw 7 as to throw 10?

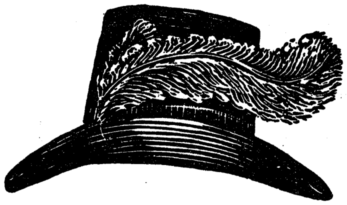
Floodgate Scandal

THE OVERWORKED LIBRARIAN

Our local librarian has been very busy. On Monday she cataloged only some of the new books received. Tuesday she received as many new books as were uncataloged on Monday, and cataloged 10. Wednesday she received 12 more than on Monday, and cataloged as many as she had done on that day. Three times as many books arrived on Thursday as she had cataloged on Wednesday, and 8 were cataloged. On Friday, 6 books arrived and 12 fewer were cataloged than were received on Wednesday. On Saturday she was able to catalog the outstanding 16 books as the library was closed. How many books arrived on Monday?

Games & Puzzles

COMING TO TOWN



Uncle Reuben and Aunt Cynthia came to town to shop. Reuben bought a suit and hat for $15. Cynthia paid as much for her hat as Reuben did for his suit; then she spent the rest of their money for a new dress.

On the way home, Cynthia called Reuben's attention to the fact that his hat cost $1 more than her dress. Then she added: "If we had divided our hat money differently so that we bought different hats, mine costing 1 and ½ times the cost of yours, then we each would have spent the same amount of money."

"In that case," said Uncle Reuben, "how much would my hat have cost?"

Can you answer Reuben's question and also tell how much money the couple spent altogether?

Puzzles of Sam Lloyd

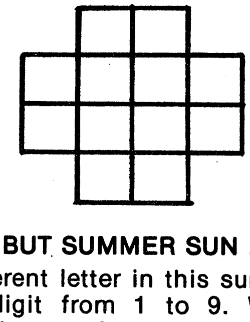
TO THE MOVIES

An old-time movie house charged admission prices of 25 cents for an adult and 10 cents for a child. If the cashier in the box-office after closing time counted the ticket stubs and found that they totaled 385 while the money amounted to $62.65—how many children had entered?

Mathematical Puzzles and Pastimes

1 TO 12

Place the integers 1 through 12 in the 12 spaces in the grid below so that each of the two rows, two columns, and five squares that can be formed with 4 numbers in each have a total of 26. How many *fundamentally* different ways can the numbers be arranged in the grid (don't count rotations or mirror images)? Can you find the arrangement in which no two consecutive numbers are next to each other vertically, horizontally or diagonally?



BUT SUMMER SUN . . .

Each different letter in this sum stands for a different digit from 1 to 9. What is the original addition sum?

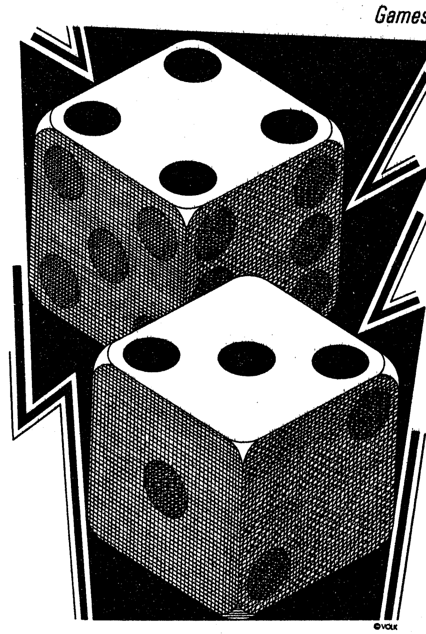
```
S P R I N G
    R A I N S
    B R I N G
    G R E E N
P L A I N S
```

Games & Puzzles

ERIC'S FAIR DICE

'Let's decide like this,' said Eric, 'I've got two dice here. We each choose a number and I'll throw the dice until the total on the two uppermost faces is a number one of us has chosen, and he will go first.' 'That's no good,' said David, 'some totals are more likely than others, and anyway there are too many of us.' 'Wrong on both counts,' replied Eric, grinning, 'these are special dice. Every different pair of faces gives a different total, and the totals start from 1 and go up consecutively. OK?' 'I'll believe that when I see the dice,' said David. He looked at them and Eric was quite right: they had what numbers on their faces?

Games & Puzzles



SUMMARIZING

The numbers 1 to 16 have already been placed in a square array. Your puzzle is to turn each horizontal and vertical column into a correct sum by placing between each pair of numbers, vertically and horizontally, a plus, minus or equals sign using eight of each.

1	2	3	4
8	7	6	5
9	10	11	12
16	15	14	13

Games & Puzzles

HARDWARE SUPPLIES

A man went into a hardware store to buy something for his house. He asked the clerk the price, and the clerk replied, "The price of one is twelve cents, the price of 30 is twenty-four cents, and the price of 144 is thirty-six cents." What did the man want to buy?

EVEN ADDITION

The number 156 is the sum of the first 12 even numbers. Curiously enough, if it is beheaded we get 56 which is the sum of the first 7 even numbers, and beheading again, 6 is the sum of the first two. There is one other three-figure number with the same property and just one 4-fiaure number. What are they?

Games & Puzzles

MAP FOLDING

Here is a teaser for anyone who has ever had trouble refolding a road map. Try solving it in your head. If our six-section map below is folded so that section A is on top facing you, which of the other five sections would it be possible to have on the bottom facing the other way?

A	1	2
3	4	5

TWO COINS

In how many ways can two coins turn up? The possible ways are heads-heads, heads-tails, tails-heads, tails-tails.

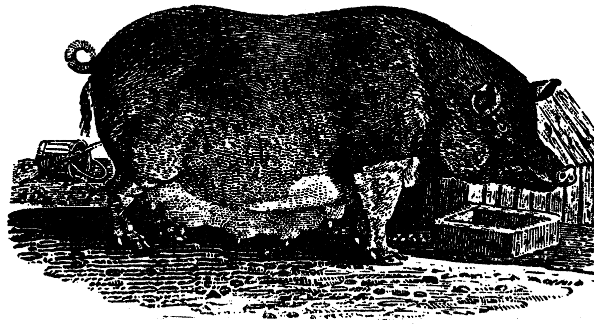
a. What is the chance that the two coins will turn up two heads?
b. What is the chance that the two coins will turn up one head and one tail?

Invitation to Mathematics

A POWER PROBLEM

The integer 844,596,301 is the 5th power of what number?

TOM THE PIPER'S SON



According to Mother Goose, Tom the Piper's son stole the pig and away he run. When Tom started after the pig, he was standing 250 yards due south of the pig. Both began running at the same time and ran with uniform speeds. The pig ran due east. Instead of running north-east on a straight line, Tom ran so that at every instant he was running directly toward the pig.

Assuming that Tom ran 1 and ⅓ times faster than the pig, how far did the pig run before he was caught? The simple rule for solving this type of problem is based on elementary arithmetic, but will doubtless be new to most of our puzzlists.

Puzzles of Sam Lloyd

SUMS IN CODE

This coded sum uses all four rules of arithmetic. Can you supply the figures? Each different letter stands for a different digit from 0 to 9.

```
      P Y F
          P  (multiplication)
    -------
    G P G Y
        F A  (addition)
    -------
    G P T A
        L E  (subtraction)
    -------
F ) G F Z G  (division)
    -------
      T G A
```

Games & Puzzles

GOT THE POINT?

The arithmetic in line A below is obviously incorrect, but with the insertion of two dots (as decimal points) in line B, the line A equation is made correct. Now see if you can insert two dots somewhere in the third line to correct the faulty arithmetic:

A. (72 x 3) — 5 = 166
B. (7.2 x 3) — 5 = 16.6
Problem: (51 — 3) x 2 = 34

A DEAL IN CANDY

Three boys received a nickel each to spend on candy. The stock offered by the candy store comprised lollipops at 3 for a cent, chocolate bonbons at 4 for a cent, and jujubes at 5 for a cent. Each boy made a different selection, but each spent his entire 5 cents and returned with just 20 pieces of candy. What were their selections?

Mathematical Puzzles

ONE PILE

This purely numerical game has been traced back to remote antiquity, and probably it antedates the games of position, such as tic-tac-toe.

A number of pebbles or counters of any description is massed in one pile. The two players draw alternately from the pile, the object being to gain the last counter.

If it were permitted to seize the whole pile, the first player would of course win; if the draw were limited to one counter at a turn, the result would depend upon whether the number in the pile were originally odd or even. Therefore, a minimum draw of one counter is set, with a maximum greater than one.

Suppose the limits are 1 to 3 counters. Then if a player finds just 4 counters left in the pile, he loses. Whatever he takes, his opponent can take the remainder. It is readily seen that the number 4 is a critical one because it is the sum of the minimum and maximum limits of the draw.

In order to leave his opponent with 4 counters to draw from, a player must previously have left him 8. Whether he then drew 1, 2, or 3, it was possible to reduce the pile to 4. Evidently the series of winning combinations, each of which is a number to be left in the pile for the opponent to draw from, is simply the muliples of 4.

If we denote "a winning combination" by *w,* and the least and most that may be drawn at a turn by *a* and *m* respectively, then

$$w=(a+m)n$$

where *n* is any integer. This formula is quite general, and is independent of the number of counters originally in the pile. If this number is of form *w,* the first player loses; if it is not, he wins by reducing it to *w.*

TO LEAVE THE LAST

The game can also be played with the object of forcing one's opponent to take the last counter. I leave it to the reader to write the formula for *w* in this case.

Mathematical Puzzles

BASIC

In base 10, 88^2 is 7744 (two different pairs of like integers). What is the smallest number squared in base 3 which will produce the same pattern?

In base 8?

Write a computer program to identify all such numbers up to base 20.

QUICKIES

1. What is the next letter in this sequence NNNENEENEEE?
2. If 6*8 = 8, 7*13 = 9, 10*15 = 12½, 10*16 = 14, what is 9*11?
3. If COS scores 0 and MEW scores 4, how many will HANKY score?

Games & Puzzles

FIND THE PATH

Can you trace a path from the lower left to upper right square (vertical, horizontal, or diagonal moves are OK) that totals exactly 100?

17	41	29	3	2
81	4	22	11	8
62	1	56	42	15
35	16	13	14	21
4	39	5	19	56

→ End

↗ Start

Richard Latta
Joliet, Illinois

DECAY CURVES

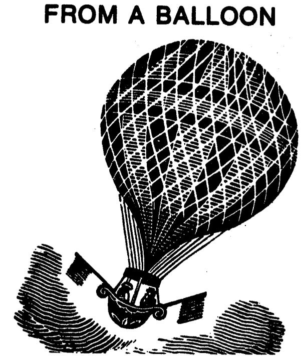
1000 coins are tossed. Those which fall tails are removed and the remainder tossed again. Those which fall tails are removed and so on. A tedious experiment to perform but one which is easy to simulate. A program written to do this should record the result of each tossing as a point on a graph. The graph would have 'coins remaining' across the page and 'throw number' down the page.

Changing the probability from ½ to an input value λ , the simulation reflects radioactive decay. λ is the probability that an atom of a particular element decays in unit time and the graph shows the number of atoms which are not decayed at each time.

The program can be improved by allowing any decayed atoms to decay again as a second radioactive material with a probability λ. Output is then to consist of a combined graph showing the number of atoms of each material.

Computer Programming Problems

FROM A BALLOON



If a stone is dropped from a balloon on a still day, does it fall directly below the balloon, or to the west of it, or to the east?

Floodgate Scandal

More Puzzles . . .

THREE BOYS

Three boys weigh a total of 350 pounds, of which Bill weighs 105 pounds. The barefoot boy weighs exactly 15 pounds less than the heaviest boy. Chuck weighs more than the boy with sneakers on. Art weighs less than the boy with loafers on.

Which boy is barefoot?

Games & Puzzles

NO COMPUTER NEEDED

Solve the following equations for x and y *in your head.*

$6751x + 3249y = 26751$
$3249x + 6751y = 23249$

ARRANGE

Can you arrange the letters of the magazine title CREATIVE COMPUTING in a 5 x 5 square grid so that starting at a C and proceeding one step at a time to a neighboring letter (horizontally, vertically or diagonally), the phrase can be spelled out in as many different ways as possible. For example, in the arrangement below CREATIVE COMPUTING can be spelled out in no fewer than 100 different ways. Can you improve on that? Any letter may appear any number of times and a spelling out of the phrase may pass through the same letter more than once.

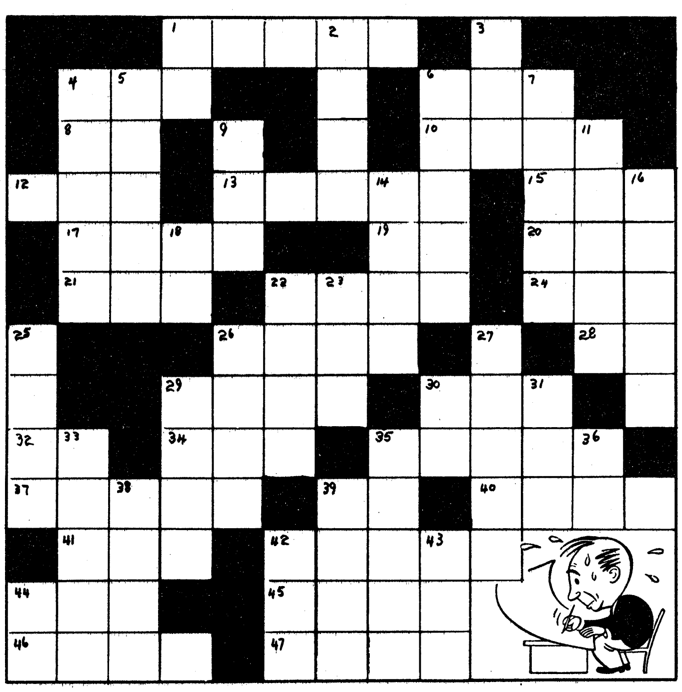
R	R	A	T	I
C	E	E	I	T
O	C	V	A	G
M	O	T	I	N
P	U	I	N	G

You try here.

Adapted by David Ahl
from a puzzle in *Games & Puzzles.*

"COMPUTER CAREERS"

by John K. Young
Braintree, Mass.



ACROSS

1 Repeat a set of operations
4 Price
6 Pin for separating ropes
8 Else
10 Separate piece of information
12 Make a mistake
13 Clear
15 The lion
17 Transfer from one memory register to another
19 Old Latin
20 Part of curved line
21 Period of time
22 Electromagnet for reading
24 Youth of Greater Tacoma
26 Repetition of instructions
28 Electronic Association
29 Storage for information unit
30 Path for transferring information
32 Member of Parliament
34 (•)
35 Artificial unit
37 Adolescent hair growth
39 One hundred one
40 Combine different fields of information into one machine word
41 Nickname
42 Counterfeit
44 Pool
45 Arrange information for output unit
46 Hoofed, cud-chewing animal
47 Circuit with two inputs and one output

DOWN

1 Computer electronics
2 Volcano flow
3 Single pulse
4 Intervene
5 Loss of precision
6 Set of columns in punch cards
7 Line for storing information in train of pulses
9 Set of identifying characters
11 Produce single sequence from two or more
14 Anathema to hippies
16 Digit in scale of eight
18 Dad
22 Retain information
23 Eastern Order of Lithographers
25 Withdraw power
26 Glass for converging spreading rays of light
27 Directs computer to next instruction
29 City in northern France
30 Boston University
31 Southern Methodist Alumni
33 Sharp voltage change
35 Symbol in scale of ten
36 Youth Corps
38 Radix in numbers scale
39 Close of musical composition
42 Implore
43 Indian tribe
44 Paid newspaper notice

Thinkers' Corner

by Layman E. Allen © 1976

MATHEMATICS PUZZLES

How many of the problems (a) through (f) below can be solved by forming an expression equal to the GOAL? (Suppose that each symbol below is imprinted on a disc.)

The expression must use:

(1) only single digits combined with operators,
(2) all of the discs in the REQUIRED column,
(3) as many of the discs in PERMITTED as you wish, and
(4) exactly one of the discs in RESOURCES.

Special Rules:

The '*' indicates "to the power of." Thus $3*2 = 3^2 = 9$.
The '√' indicates "the nth root of." Thus $^3\sqrt{8} = 2$.
Parentheses can be inserted anywhere to indicate grouping, but never to indicate multiplication.

Problem	GOAL	REQUIRED	PERMITTED	RESOURCES
(a)	18	5+	48+	-X÷√ 249
(b)	9	7-	28÷	-X÷* 678
(c)	11	8÷	26-	+- X * 148
(d)	6	23 -	158	-- X ÷245
(e)	1	34 ÷	26+ -	-X÷0 125
(f)	10	46 *	23÷	+- √ 1 249

Some Suggested Answers (frequently there are others):

(a) 4 + 5 + 9
(b) (8 X 2) - 7
(c) (6 ÷ 2) + 8
(d) 5 - (2 - 3)
(e) (2 ÷ 4) + (3 ÷ 6)
(f) (4 * 2) - 6

If you enjoy this kind of puzzle, you might like playing EQUATIONS: The Game of Creative Mathematics. Free information about this and other instructional games is available upon request from The Foundation for the Enhancement of Human Intelligence, 1900-E Packard Road, Ann Arbor, MI 48104.

COMPUTER RECREATIONS

by D. Van Tassel

Self-reproducing program

Every so often one runs into a program that is really interesting to write but very simple to understand. I won't claim credit for this program since it has appeared in various forms in other places. Here is the program: Write a program that prints an exact copy of itself. No input statements are allowed.

This can be done in any programming language. The program is an interesting exercise in program planning. For those who do not feel the above program is a sufficient challenge I offer the following interesting variation: The Dizzy Operator Program.

Write a program that prints an exact copy of itself. No input statements are allowed. In addition because the computer operator is quite careless and sometimes puts the cards (or paper tape) in the machine upside down the program should execute exactly the same, either way.

Now if anyone sends me a nice solution to either problem, I will publish it in a later column. (Send to D. Van Tassel, Computer Center, Univ. of California, Santa Cruz, CA 95064).

Dennie Van Tassel is the author of *Program Style, Design, Efficiency, Debugging, and Testing*. Many of the problems in this "Computer Recreations" series come from this book. Dennie also edited the newly-released *The Compleat Computer* published by SRA.

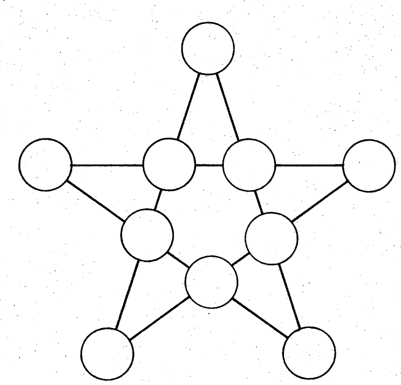
A BICENTENNIAL MAGIC STAR

by Carl E. Heilman
Pa. Dept. of Education
P.O. Box 911
Harrisburg, PA 17126

Of course, all of you know the story of Betsy Ross and the clever manner in which she folded a piece of paper or cloth and with a single snip of her shears, cut it to form a symmetrical five-pointed star. I hope you know how to duplicate her construction and will take advantage of its appropriateness for both Christmas and the Fourth of July.

But my intention is to suggest that you relate more closely to George Washington and the cherry tree when George's father, after the tree was cut down, told George he wanted him to plant ten trees in place of the one chopped down. It was a sort of tithing program in reverse. But George had to plant the ten trees in five rows with four trees in each row. Imagine George's surprise when he found Betsy Ross's five-pointed star staring him in the face!

Now my suggestion is to place a different integer at each of those points of intersection, so that the sum of the integers in each row is 76. The word "different" avoids the obvious solution which uses 19 at each point of intersection. Using integers, I have obtained three different solutions, one using the values 13, 14, 15, 16, 17, 20, 22, 23, 24, 26. I should like to know how many different solutions can be discovered and whether there is any way of determining the total number of distinctly different solutions, those using different groups of ten numbers. I should appreciate any reference which studies this problem of the magic five-pointed star.



Since this problem first appeared in the Fall 1975 *PCTM Newsletter*, I have received some 60 different solutions from teachers and students in the Wyoming Valley West High School, Plymouth, PA. But there is still no conjecture as to the total number of possible solutions. Can anyone out there in *Creative Computing* land help?

More Puzzles . . .

FILLERUP

Try to arrange the full names of the fifty states into an interlocking pattern, crossword-fashion, minimizing the area of the rectangle into which the completed pattern will fit. All of the state names must be used once and only once. All of the names must be interconnected; that is, no name or group of names may be unconnected from the rest of the names in the completed diagram. Your crossword diagram must be of the kind which uses blacked-out squares. Spaces in state-names are to be ignored.

Games & Puzzles

JOKER

Four cards, one of each suit, and one each Jack, Queen, King and Ace are laid out in a row.

1. The heart isn't next to the club.
2. No card is next to its immediate senior in rank.
3. The colors of the suits alternate.
4. The king and queen face in opposite directions.
5. The Jack of diamonds is not in the row.

Identify the four cards.

READING MATTER

A printer uses 1215 characters to number the pages of a book. How many pages are in the book?

YEAR IN, YEAR OUT

If you take the figures in the year 1974 and multiply them together (1 x 9 x 7 x 4) you end up with the product 252. What is the next date on which this happens?

Similarly 1975 multiplies out to give 315; how long will we have to wait until this happens again?

Games & Puzzles

Thinkers' Corner

by Layman E. Allen © 1976

WORD PUZZLES

How many of the problems (a) through (f) below can you solve by forming a network of words that have exactly as many letters as the number listed as the GOAL? (Suppose that each symbol below is imprinted on a disc.)

To qualify as a network

(1) all sequences of discs across and down must be words,
(2) the words must have two or more letters and not be proper names,
(3) all of the discs in the REQUIRED column must be used,
(4) as many of the discs in PERMITTED as you wish may be used, and
(5) at most one of the discs in RESOURCES may be used.

Example: The number of letters in the words of the network

```
CAT   is 7:        CAT=3, TO=2, ON=2
  ON                  3 + 2 + 2 = 7
```

The number in the network CAT is 3.

Problem	GOAL	REQUIRED	PERMITTED	RESOURCES
(a)	7	DR	EORT	CEKLMPS
(b)	8	EIZ	EHKO	DENOPRS
(c)	10	CEO	AEMOT	ACKMNQU
(d)	14	EOS	AMPSU	ADNORST
(e)	16	NOW	ANNOTW	ACFGHNP
(f)	21	EITV	ELNNRST	ABEMSRX

If you enjoy this kind of puzzle, you may like playing ON-WORDS: The Game of Word Structures. Free information about this and other instructional games is available upon request from The Foundation for the Enhancement of Human Intelligence, 1900-W Packard Road, Ann Arbor, MI 48104.

Some Suggested Answers (frequently there are others):

```
(a) R        (c) ONCE       (e) WANT
    O              O             NOON
    S             AT               W
  RED              E               N

(b) SHOE     (d) USE        (f)  L
    I            SOAP           SIR
    Z              R            EVEN
    E                           TEN
                                  T
```

COMPUTER RECREATIONS

by D. Van Tassel

Chess Programs

Chess is a popular game but not many of us can write a program to play chess. But there are many interesting programs that can be based on chess pieces.

A real simple program is one that reads the row and column of the queen as input, and as output prints a picture of the board with the square the queen is on marked with a Q. Next mark the squares the queen could move to with *'s and mark all other squares with +'s. A queen can move vertically, horizontally, or diagonally as far as desired. For example, a chess-board with a queen in the second row and third column would look as follows:

```
+ * * * + + + +
* * Q * * * * *
+ * * * + + + +
* + * + * + + +
+ + * + + * + +
+ + * + + + * +
+ + * + + + + *
+ + * + + + + +
```

A more interesting program is the Eight Queens problem: Write a program to place 8 queens on a chessboard such that no queen can take any of the others. This means that no two queens may be on the same row, on the same column, or on the same diagonal. This is not a trivial program nor is impossible. I suggest you try it by hand before attempting the program. An elegant solution by Niklaus Wirth is discussed in the book *Structured Programming* by Dahl, Kijkstra, and Hoare, Academic Press.

And Still More . .

THE KING'S SEAL

The King of Dnal Retupmoc, a tiny kingdom lying between India and Laos, has a serious problem. His computer, through a malfunction, destroyed a very important program which produces the King's seal. To be official the seal must be printed by the magic terminal with a special printing device. The seal must look as follows:

```
       X
      XXX
     XXXXX
    XXXXXXX
   XXXXXXXXX
  XXXXXXXXXXX
 XXXXXXXXXXXXX
XXXXXXXXXXXXXXX
 XXXXXXXXXXXXX
  XXXXXXXXXXX
   XXXXXXXXX
    XXXXXXX
     XXXXX
      XXX
       X
```

To avoid unauthorized use of his terminal in producing the seal the King's computer is designed to reject any program which uses the words "Tab" and "Print" more than once and the computer operates only in the BASIC language.

Your mission, should you accept, is to fly to Dnal Retupmoc and re-program the King's computer. (This paper will self-destruct upon contact with a wastebasket.)

BUT NO INCEST

There are three families each with two sons and two daughters. In how many ways can all these young people be married?

MAX MACHINE

Given a machine which can do addition, subtraction, multiplication, division, and find absolute values, show how to program it to find max (x,y); that is, if two numbers are fed in, the output should be the larger of the two. No "IF" statements allowed.

Thinkers' Corner

by Layman E. Allen © 1975

SET THEORY PUZZLES

How many of the problems (a) through (f) below can you solve by forming an expression that will name the number of cards in the universe that is listed as the GOAL? (Suppose that each letter and symbol below is imprinted on a disc.)

The expression must use:

(1) all of the discs in the REQUIRED column
(2) as many of the discs in PERMITTED as you wish, and
(3) exactly one of the discs in RESOURCES

Universe of Cards

Card	1	2	3	4	5	6
Letters	A B D	B C	A B C	B	B D	C

Examples:

The expression A names 2 cards (1,3).

The expression A' (complement) names 4 cards (2,4,5,6).

The expression B ∩ C (intersection) names 2 cards (2,3).

The expression B ∪ C (union) names 6 cards (1,2,3,4,5,6).

The expression C-B (difference) names 1 card (6).

Problem	GOAL	REQUIRED	PERMITTED	RESOURCES
(a)	4	∪	B C D -	A B C ∩ - '
(b)	4	D	B D ∪ ∩	A B C D - '
(c)	1	A ∪	A B C ∪	A B C D ∪
(d)	6	C '	A B D	A B C D ∪ ∩
(e)	1	D	B D ∩ '	B D ∩ '
(f)	1	C ∩	B C - '	B C D ∩ -

Some Suggested Answers (frequently there are others):

(a) A ∪ C (b) D' or D ∪ (B ∩ C) (c) Impossible

(d) (C ∩ D)' (e) B' ∩ D'

(f) D' ∩ (B-C) or C'-(B ∩ D) or (C ∩ D)'-B or D' ∩ (C-B).

If you enjoy this kind of puzzle, you may like playing ON-SETS: The Game of Set Theory. Free information about this and other instructional games is available upon request from THE FOUNDATION FOR THE ENHANCEMENT OF HUMAN INTELLIGENCE, 1900-S Packard Rd., Ann Arbor, MI 48104.

FREE THROWS

If a basketball player averages 7 out of 10 at the free throw line, what is the expected value of his score if he is allowed to shoot until he misses?

The puzzles above are from *The Point Set* published by the Dept. of Mathematics, Univ. of Wisconsin, Stevens Point, WI 54481. Editor is Gordon Miller

'Not every problem is one to be solved by computer programming.'

TURNING A PUZZLE INTO A LESSON

Eugene D. Homer
C. W. Post College, Greenvale, NY

The second problem in the feature column "Puzzles and Problems For Fun", *Creative Computing*, 1, 4, (May-June, 1975) proved to be an ideal nucleus for class discussion in a course in advanced programming, although the results were not what, I presume, the author intended.

The problem was stated thusly:

"Mr. Karbunkle went to the bank to cash his weekly paycheck. In handing over the money, the cashier, by mistake, gave him dollars for cents and cents for dollars.

"He pocketed the money without examining it and spent a nickel on candy for his little boy. He then discovered the error and found he possessed exactly twice the amount of the check.

"If he had no money in his pocket before cashing the check, what was the exact amount of the check? One clue: Mr. Karbunkle earns less than $50 a week."

I assume the intent of the author was to have readers write a computer program to solve the problem by trial and error. My intent was to show the class how analysis of the problem before coding could simplify the program. I would like to share this lesson and the resulting conclusion with the readers of *Creative Computing*.

Our first step was to state whatever relationships we could from the problem in mathematical form.

Let D be the integer number of dollars and C the integer number of cents on Mr. Karbunkle's paycheck. The total amount printed on his check, expressed in cents, is:

$$A = 100D+C \qquad (1)$$

Since the teller reversed D and C, the amount of cash Mr. Karbunkle received, again expressed in cents, is:

$$R = 100C+D \qquad (2)$$

We are told that

$$R-5=2A \qquad (3)$$

Substituting Equations 1 and 2 in Equation 3, we obtain:

$$100C+D-5=2(100D+C),$$

which can be simplified to:

$$199D=98C-5, \qquad (4)$$

We have one equation, in two integer unknowns, which does look like a problem for trial and error solution. If we were to code at this point, we might come up with something like this, remembering that D is less than 50:

```
   INTEGER D,C
   DO 1 D = 1, 49
   N = 199*D
   DO 1 C = 1, 100
   L = 98*(C-1)-5
   IF (N-L) 1,2,1
2  A = (100*D+C)/100
   WRITE (5,3) A
   CALL EXIT
1  CONTINUE
   WRITE (5,4)
   CALL EXIT
3  FORMAT (F7.2)
4  FORMAT (1X,'NO SOLUTION')
   END
```

Although this looks like a fairly simple program, I pointed out that it would require a maximum of 4,900 repetitions of the main loop. (In the following, it becomes convenient to measure interatives by the number of times the IF statement is executed.) In view of this large amount of computation we agreed that the analyst should attempt two things:

a) Reduce the amount of computation in the loop, and

b) Reduce the number of times the program must loop.

Tackling the first idea, the class came up with such suggestions as replacing lines 2 and 3 of the above program with the line:

```
DO 1 N = 199, 9751, 199
```

This led to a similar discussion about simplifying lines 4 and 5 of the above program. In this round, it became apparent that C must be greater than 1, since for C=1, L = –5, and L would never be equal to N. It also became apparent that our inner loop could be terminated as soon as L became greater than N. We now had our program down to something like this:

```
   INTEGER D,C
   DO 1 N=199, 9751, 199
   DO 5 L=93, 9697, 98
   IF (N-L) 5, 2, 1
2  A=(100*D+C)/100.
   WRITE (5, 3) A
   CALL EXIT
5  CONTINUE
1  CONTINUE
   etc.
```

We had reduced our maximum number of executions of the IF statement by almost one-half (to 2,499 executions) and had removed all arithmetic calculation from the loops.

We were still not happy with the program, since the inner loop, on L, was too repetitive. We saw that if a particular value of L was less than a particular value of N, there was no need to try that value of L again for the next value of N. This led us to the removal of the inner loop altogether:

```
   INTEGER D,C
   L = 93
   DO 1 N = 199, 9751, 199
   IF (N–L) 1, 2, 5
2  A = (100*D+C)/100.
   WRITE (5, 3) A
   CALL EXIT
5  L=L+98
1  CONTINUE
   etc.
```

A few "runs" by hand of this program indicated that we would try only two values of L for each value of N, reducing the maximum number of IF statement executions to 98.

Before we left this approach, we took another look at Equation 4. For any non-negative integer value of C, the right side of the equation will be odd. Therefore the term 199 D must be odd, and therefore D must be odd. Thus, D may assume only the values 1,3,5,...,49, and N = 199 D will increase in increments of 2(199)=398. Our last program, then, can have its DO statement changed to

```
DO 1 N = 199, 9751, 398,
```

resulting in another halving of the iterations.

It was now time to take another tack. I reminded the class of last week's work with modular numbers, and showed them that Equation 4 satisfied the first definition of a modular number,

$$N = q \cdot m + r,$$

but failed the second definition,

$$0 < r < (m-1)$$

where $N = 199D$
$q = C$
$m = 98$
$r = -5$

However, we could rewrite Equation 3 as:

$$199D = 98C-98-5+98$$

or

$$199D = 98(C-1)+93$$

which satisfied both definitions of modular numbers with

$N = 199D$
$q = C-1$
$m = 98$
$r = 93$

We could then write, from the familiar expression

$$r \equiv N \bmod m,$$

$$93 \equiv 199D \bmod 98 \qquad (5)$$

Our strategy then could be to use the MOD function as we increment N. If we find a value of N satisfying Equation 5, we can solve for

$$D=\frac{N}{199}$$

and, by rewriting equation 4

$$C=\frac{N+5}{98}$$

Our new program follows. Note that we have also dropped the integer declaration as being unnecessary.

```
   DO 1 N = 199, 9751, 398
   IR = MOD(N,98)
   IF (IR–93) 1, 2, 1
2  D = N/199
   C = (N+5)/9800.
   A = D+C
   WRITE (5, 3) A
   CALL EXIT
1  CONTINUE
   WRITE (5,4)
   CALL EXIT
3  FORMAT (F7.2)
4  FORMAT (1X,'NO SOLUTION')
   END
```

This looked like a reasonable program, requiring only 24 repetitions of the IF statement, maximum. We set out to run it by hand, with these results:

N	IR
199	3
597	9
995	15
1393	21

At this point the class saw that as N increased by 199, IR increased by 6, and that it might be possible to "figure out" when IR would hit 93.

It took only a few minutes to work out the fact that

for $N = 199+398i; i=0,1,2,...$
$IR = 3+6i$

If IR is to be equal to 93:

$$93 = 3+6i$$

or,

$$i = 15$$

This would occur when

$$N = 199+398(15)=6169,$$

and

$$D = \frac{6169}{199} = 31,$$

$$C = \frac{N+5}{98} = 63,$$

$$A = \$31.63$$

To be sure, we checked with equations 2 and 3:

R = \$63.31
\$63.31 – \$.05 = \$63.26 = 2(\$31.63).

Our "program" now has reduced to:

```
   WRITE (5,5)
   CALL EXIT
5  FORMAT (1X,'$31.63')
   END
```

We spent about two hours going over this puzzle. While much of our work was useless in terms of the final solution, the class did learn some valuable lessons from the discussion. They learned that careful analysis of a problem can lead to a startling reduction in the amount of computing to be done. They learned that it pays to run through a program "by hand" a few times to discover hidden relationships. Finally, they learned that not every problem presented in a computer-oriented environment is a problem to be solved by computer programming.

Here's a challenging new 2-person game to play on your pocket calculator.

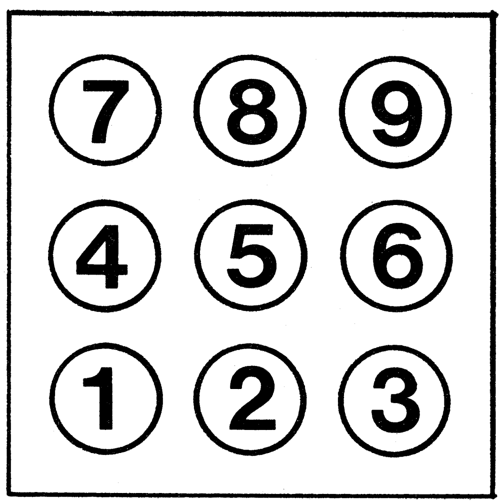
THE KEYBOARD GAME

by L. D. Yarbrough

Suppose someone approaches you, electronic calculator in hand, and suggests that you play the following game:

"We start by selecting a number to enter into the calculator—say 100. Suppose I choose the number: then you pick a key, from 1 to 9. You subtract that number from the total. Then I pick a key, only my key must be next to your key on the Keyboard. I subtract that, then you pick a key next to mine, and so on. The first one to turn on the 'minus' sign loses the game."

For example, if the *4* key is chosen, the next player may subtract only 1,2,5,7, or 8 from the total. If he chooses *5*, the other player may choose any key *except* 5. If he chooses *1*, the next key chosen must be *4*, *2*, or *5*. Choosing a key always means subtracting the value of that key from the total.



Before reading further, I suggest you have some fun playing the game with a friend. If you don't have a calculator, try pencil and paper. The calculator just makes it go faster. And if you play with a calculator, remember always to hit the (—) key after *every* number key.

The winning strategy for this game is unusual in a way which I will explain shortly. To appreciate just how unusual it is, though, let's consider the solution of a related Keyboard game which is much simpler to analyze. Suppose we take away any restrictions on which of the keys (1-9) can be used at each turn. Now the winning method will soon become easy to see: just subtract the right-hand digit from the total. That, of course, produces a zero in that position, unless there was already a zero there. So if, at your turn, the total is a multiple of ten—especially if it is exactly zero!—you lose. Otherwise, you win. Now this means that the player who chooses the original total can win simply by choosing a multiple of ten, and subtracting the right-hand digit whenever it is his turn to play.

This situation holds in a very large number of games in which two players play alternately: the selection of the initial conditions of the game is quite often enough to win the game. NIM is a classical example. The literature of recreational mathematics is filled with other examples.

Now back to our original Keyboard Game. In his game, as we shall see, the player who makes the choice of the first *key* has a tremendous advantage. If we assume that the initial entry into the calculator is large enough—say greater than 15—the player who chooses the first key can announce his choice to the whole world, let the other player choose whatever starting total he wishes, and still win the game. The winning strategy depends on the position of the first key, not on its numeric value!

We can develop a winning strategy for any given initial total by building up a table, starting at zero, and calculating the winning keys as a function of currently available keys and winning strategies for lower totals. Figure 1 shows the winning keys for totals up to 100. In this figure, "T" appearing at a given row and column means, "For the current total equal to this row number, the key at the head of this column is a winner, if we can get to it." "F" designates a key choice which gives our opponent a chance to win. For example, if the total is 10, *2* is a winning key: it reduces the total to 8, and the only winning keys for 8 are the *7* and *8* keys, which are on the opposite side of the keyboard. *3* is a loser; it reduces the total to 7 and leaves the *6* key to our opponent.

From Fig. 1 it is clear that the corners of the keyboard, especially the *1* and *3* keys, offer winning opportunities for all initial totals above a certain limit. Choosing any other key nearly always loses because our opponent can grab a corner key and keep returning to it no matter what we try. For small totals we need to be careful because the winning keys are not always in the corner anymore.

What if we include the *0* key? If we give it its nominal value, the game becomes very simple: choose that key and keep returning to it; our opponent is the only one who subtracts anything from the total. So give it a value of 10. You will enjoy investigating this variant of the Keyboard Game for yourself. Actually, it has several variants: on some calculators the *0* key is under the *2* key, on others it is under the *1*, etc. In this last version, with *0* under *1*, there is *one* key which is a winner for all totals above 35. I leave it to you to figure out which one it is.

	1	2	3	4	5	6	7	8	9
0	F	F	F	F	F	F	F	F	F
1	T	F	F	F	F	F	F	F	F
2	T	T	F	F	F	F	F	F	F
3	F	F	T	F	F	F	F	F	F
4	T	F	T	T	F	F	F	F	F
5	F	F	F	F	T	F	F	F	F
6	F	F	T	F	F	T	F	F	F
7	T	F	T	T	F	T	T	F	F
8	F	F	F	F	F	F	T	T	F
9	T	F	F	F	F	F	T	T	T
10	T	T	F	T	T	F	T	T	T
11	F	F	T	F	F	F	F	T	T
12	T	F	T	F	F	F	F	F	T
13	T	F	F	F	F	F	T	F	T
14	T	F	T	F	F	F	F	F	F
15	T	F	T	F	F	F	F	F	F
16	T	F	T	F	F	F	F	F	F
17	T	F	T	F	F	F	F	F	F
18	T	F	T	F	F	F	F	F	F
19	T	F	T	F	F	F	T	F	F
20	T	F	T	F	F	F	T	F	F
21	T	F	T	F	F	F	T	F	T
22	T	F	T	F	F	F	T	T	T
23	T	F	T	F	F	F	T	T	T
24	T	F	T	F	F	F	T	T	T
25	T	F	T	F	F	F	T	T	T
26	T	F	T	F	F	F	T	T	T
27	T	F	T	F	F	F	T	F	T
28	T	F	T	F	F	F	T	F	T
29	T	F	T	F	F	F	F	F	T
30	T	F	T	F	F	F	F	F	T
31	T	F	T	F	F	F	F	F	F
32	T	F	T	F	F	F	F	F	F
33	T	F	T	F	F	F	F	F	F
34	T	F	T	F	F	F	T	F	F
35	T	F	T	F	F	F	T	F	F
36	T	F	T	F	F	F	T	F	T
37	T	F	T	F	F	F	T	F	T
38	T	F	T	F	F	F	T	F	T
39	T	F	T	F	F	F	T	T	T
40	T	F	T	F	F	F	T	T	T
41	T	F	T	F	F	F	T	T	T
42	T	F	T	F	F	F	T	F	T
43	T	F	T	F	F	F	T	F	T
44	T	F	T	F	F	F	T	F	T
45	T	F	T	F	F	F	T	F	T
46	T	F	T	F	F	F	F	F	T
47	T	F	T	F	F	F	F	F	T
48	T	F	T	F	F	F	F	F	F
49	T	F	T	F	F	F	T	F	F
50	T	F	T	F	F	F	T	F	F
51	T	F	T	F	F	F	T	F	T
52	T	F	T	F	F	F	T	F	T
53	T	F	T	F	F	F	T	F	T
54	T	F	T	F	F	F	T	F	T
55	T	F	T	F	F	F	T	F	T
56	T	F	T	F	F	F	T	T	T
57	T	F	T	F	F	F	T	F	T
58	T	F	T	F	F	F	T	F	T
59	T	F	T	F	F	F	T	F	T
60	T	F	T	F	F	F	T	F	T
61	T	F	T	F	F	F	T	F	T
62	T	F	T	F	F	F	T	F	T
63	T	F	T	F	F	F	F	F	T
64	T	F	T	F	F	F	T	F	T
65	T	F	T	F	F	F	T	F	F
66	T	F	T	F	F	F	T	F	T
67	T	F	T	F	F	F	T	F	T
68	T	F	T	F	F	F	T	F	T
69	T	F	T	F	F	F	T	F	T
70	T	F	T	F	F	F	T	F	T
71	T	F	T	F	F	F	T	F	T
72	T	F	T	F	F	F	T	F	T
73	T	F	T	F	F	F	T	F	T
74	T	F	T	F	F	F	T	F	T
75	T	F	T	F	F	F	T	F	T
76	T	F	T	F	F	F	T	F	T
77	T	F	T	F	F	F	T	F	T
78	T	F	T	F	F	F	T	F	T
79	T	F	T	F	F	F	T	F	T
80	T	F	T	F	F	F	T	F	T
81	T	F	T	F	F	F	T	F	T
82	T	F	T	F	F	F	T	F	T
83	T	F	T	F	F	F	T	F	T
84	T	F	T	F	F	F	T	F	T
85	T	F	T	F	F	F	T	F	T
86	T	F	T	F	F	F	T	F	T
87	T	F	T	F	F	F	T	F	T
88	T	F	T	F	F	F	T	F	T
89	T	F	T	F	F	F	T	F	T
90	T	F	T	F	F	F	T	F	T
91	T	F	T	F	F	F	T	F	T
92	T	F	T	F	F	F	T	F	T
93	T	F	T	F	F	F	T	F	T
94	T	F	T	F	F	F	T	F	T
95	T	F	T	F	F	F	T	F	T
96	T	F	T	F	F	F	T	F	T
97	T	F	T	F	F	F	T	F	T
98	T	F	T	F	F	F	T	F	T
99	T	F	T	F	F	F	T	F	T
100	T	F	T	F	F	F	T	F	T

Figure 1

A POWERFUL PROBLEM

Some years ago, a mathematician employed by one of America's leading aircraft manufacturing companies got into the habit of 'number-doodling'. Number-doodling is where you start from some particular number (a telephone number, a birthdate, a car number, etc.) and see what sort of mathematical peculiarities can be developed from the number.

It turned out that the mathematician lived at 153 Westpark Street. After some preliminary doodling with this number, he noticed that if he added together the third powers of the constituent digits, he would arrive back at the number 153. That is:

$$1^3 + 5^3 + 3^3 = 153$$

When he tackled his four digit telephone number in the same way, he was agreeably surprised to find the same effect. His telephone number was 8208. Summing the fourth powers of the constituent digits gives the total 8208. That is:

$$8^4 + 2^4 + 0^4 + 8^4 = 8208$$

Armed with these two fascinating pieces of number-doodling, he began to treat methodically every number connected with his existence in the same way. That is, for a number with *n* digits, each of the constituent digits would be raised to the *n*th power and then summed, in the hope that the sum was equal to the original number.

Surprisingly, the zip code of the area in which he lived, 54748, was notable for exactly the same property. That is:

$$5^5+4^5+7^5+4^5+8^5=54748$$

And his car's registration number, 548834, displayed exactly the same peculiar property:

$$5^6+4^6+8^6+8^6+3^6+4^6=548834$$

And his employee number at the aircraft company, too, was equally odd. Who else but a mathematician could see any beauty in a number like 1741725? Still, you must admit that it is rather interesting that:

$$1^7+7^7+4^7+1^7+7^7+2^7+5^7=1741725$$

Three other numbers that formed part of his life also had this property. His bank account number (24678050), his driving licence number (146511208), and his Social Security number (4679307774). Thus:

$$2^8+4^8+6^8+7^8+8^8+0^8+5^8+0^8=24678050$$

$$1^9+4^9+6^9+5^9+1^9+1^9+2^9+0^9+8^9=146511208$$

$$4^{10}+6^{10}+7^{10}+9^{10}+3^{10}+0^{10}+7^{10}+7^{10}+7^{10}+4^{10}=4679307774$$

Would the reader care to search for other numbers, up to ten digits long, that have the same properties? All told, there are no more than 15 of these additional numbers, assuming that you exclude all the one-digit numbers. One last question: what was the name of the town in which our number-doodling mathematician lived?

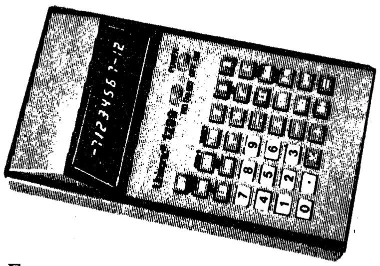
Is your number up?

Everyone's attention wanders once in a while, but the lapse could be fatal at the top of the stairs, on the road or while using a power tool.

J. R. Block of Hofstra University has researched a new technique to single out people who have trouble paying attention. The method can be used to predict a person's performance on tasks requiring uninterrupted attention.

Block's method involves an illuminated board with randomly arranged numbers that glow in five different colors. The task is to locate the numbers in order. Most people will find each in 10 to 20 seconds. He says slow performance indicates momentary attention gaps, but very rapid performance may indicate a person who sacrifices accuracy for speed. Both extremes can be dangerous.

34	19	42	54	45
26	16	39	28	57
40	35	14	56	30
12	29	44	51	23
50	43	36	24	11
37	20	55	32	47
25	41	17	53	38
13	22	48	10	58
52	18	21	31	46
27	49	33	15	59



7 POCKET CALCULATOR GAMES

From THE CALCULATING BOOK *by James T. Rogers, copyright © 1974 by James T. Rogers, reprinted by permission of the publisher, Random House, Inc.*

To order *The Calculating Book,* see *Creative Computing Library* ad elsewhere in this issue.

← PAGE 78

Now your informative and versatile machine is going to tell you the month and day of your birth, together with your age. Hold on to your hat.

Take the number representing the month of your birth (January is 1, February, 2, and so on) and multiply it by 100. Add the date of your birth. Multiply by 2, add 9, multiply by 5, add 8, multiply by 10, subtract 422, add your age and subtract 108. The result will be a five-digit number; the first digit will tell the month of your birth, the next two the date of your birth (as, say, 08 if it was before the 10th of the month) and the last two your age.

Your calculator can give you advice on driving. Suppose you are cruising along at 85 miles per hour and you want to know what advice your calculator would offer in that situation. Divide 85 by 79.069767 and read the answer upside down. (If you are driving that fast, maybe it would be better if you put yourself upside down and left the calculator right side up.)

If you want to talk back to the calculator, giving it either an instruction or an opinion of its character (depending on your mood), enter 7334 and read the answer upside down.

Reach in your pocket and pull out whatever change is there. Count the amount, and with the total as the starting point (used as a whole number, omitting the decimal) do the following things: multiply by 10, add 1, multiply by 2, add 21 and multiply by 5.

The result will be a number ending in 15. Discard the 15 and subtract 1 from what is left. The answer will be the amount of change you started with.

Imagine a set of chips, each of which bears a number from 0 through 9. They are lined up in the following order: 6328907154.

Problem: Without changing the order except by moving digits from end to end (making, for example 4632890715), find the two groups that can be multiplied to produce the third group. Here again you can put your calculator through a multiplication drill. (Hint: One multiplier has three digits and one has two; the answer has five.)

Answer: 715×46=32890.

Take a number having any reasonable number of digits (as many as six, say) and go through the following abracadabra:

Multiply by 2.
Add 4.
Multiply by 5.
Add 12.
Multiply by 10.
Subtract 320.

The result will be a number ending in one or more zeroes. Drop them and you will be left with the number that you started with.

Suppose you start with 52871. (You'd better write it down.) Then: 52871 × 2 + 4 × 5 + 12 × 10 − 320 = 5287100.

Starting with any three single digits, as long as they are different from one another, see if you can find a combination besides 6 × 21 where the product contains the digits that you multiplied (126 in this case).

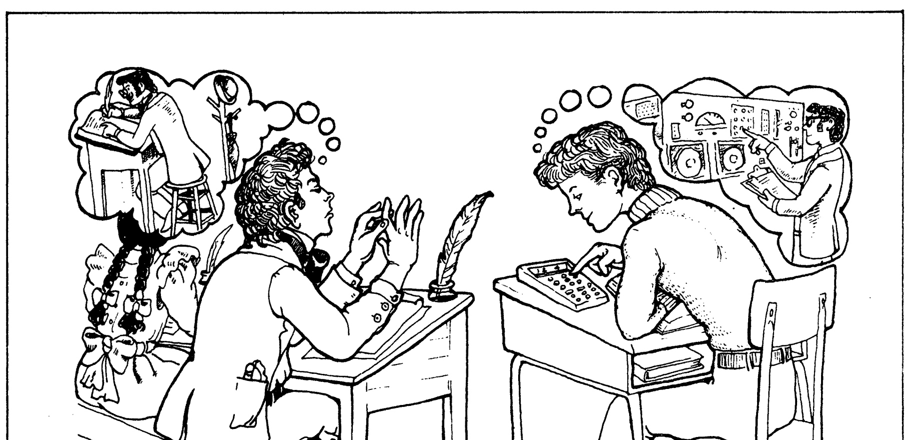
It might strike you that this is a search for a needle in a haystack. In fact, however, there is only one combination to be found. (Hint: One of the multipliers is 51. You therefore can limber up your calculator on some trial multiplications of 51 and the digits from 1 through 9 until you find a combination that gives you an answer containing 5, 1 and the number you multiplied 51 by.)

Answer: 3 × 51 = 153.

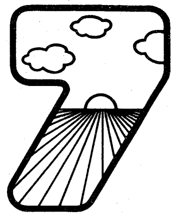
You can drive yourself a bit closer to distraction by trying the same thing with four digits. In this case, as with three, you can arrange them in any way (4716, for example, could be 4 × 716, 47 × 16, 471 × 6, 7 × 416 and so on).

The possible solutions are: 8 × 473 = 3784; 9 × 351 = 3159; 15 × 93 = 1395; 21 × 87 = 1827; 27 × 81 = 2187; and 35 × 41 = 1435.

The number 3025 displays a remarkable quirk when it is split into two parts, 30 and 25. Add the two parts and square the result. ■



The Mystic



by Anthony Dickins

FROM time immemorial, the number 7 seems to have possessed certain mysterious, or even 'magic', properties. Probably this originated with the calculations of the calendar in the early prehistoric period when chief priests, magicians, astrologers and adepts of all kinds all over the world were engaged in observing the motions of the Sun, the Moon and the Stars, carefully recording their results. The 365 days of the solar year were (nearly) divisible into 52 'weeks' of 7 days each (with only one day left remaining to account for). It is possible that at one time and in some places the year was divided into 12 months of 30 days each, the five 'extra' days being treated as 'dark' days around the winter solstice from 21st December (the shortest day) to the 25th when it was clear beyond doubt that the sun was not dead but was being reborn. But with the four solstices and equinoxes of the year occurring at regular intervals, it seems most likely that the year was first divided into four 'quarters', each having 91 days. This would leave only one day per year unaccounted for. The number 91 has only two factors, 7 and 13, and this may well have been the influence that led to establishing 13 weeks of 7 days each in the 91-day quarter. If so, this, too, probably accounts for the very deep-rooted superstition about the number 13, which is found in cultures at great distances apart. It is notable that 7 and 13 are the two numbers most heavily charged with the aura of superstition and mysticism.

Mathematically, the number 7 has several remarkable properties. Here are some well-known ones, together with some less well-known and some quite new.

The fraction one-seventh expressed in decimal form is 0.142857142857... The six figures 142857 recur infinitely. Treating 142857 as a six digit number, and setting out a multiplication table by the first six numbers, we get this curious result:

142857 x 1 = 142857
142857 x 3 = 428571
142857 x 2 = 285714
142857 x 6 = 857142
142857 x 4 = 571428
142857 x 5 = 714285

These are the six 'cyclic permutations' of the number 142857 which is the only number less than a million whose cyclic permutations are all multiples of itself. It will also be noticed that there is a palindromic effect in the digits, columns 1 to 6 from left to right being identical with rows 1 to 6 from top to bottom.

Another curious feature of this cyclic number is that the dirits form a cyclic set, not only in our usual scale 10, but also all other scales from base-9 inclusive, upwards. Obviously, this does not hold for the binary scale, base-2, and other lower-scales where only digits of lesser value than 9, 8, 7, 6, etc., may be used.

In passing, it may be noted that the next known 'cyclic' number of this sort has 18 digits — 052, 631, 578, 947, 368, 421. The leader who wishes to verify this will have a pleasant surprise when he multiplies this number by 2, 3, 4, ... 10, and sets the products down in rows beneath the original number just as was done with 142857 in the previous paragraph.

Reverting to our original decimal expression for the fraction one-seventh (0.142857), if we divide this by 7, we get another recurring decimal, this time of 42 (7×6) digits:

0.020408163265306122448979
591836734693877551 ...

In this number, there are 7 six-digit sequences, each beginning with a different digit, 0134578. Notice that the ten different digits are represented as evenly as possible in a 42-digit number. There are five 3's, five 6's, and four each of the other digits.

Another 'sevenly' number is 5040, or factorial 7. (Factorial seven, written 7!, is equal to 7x6x5x4x3x2x1). 5040 was noticed by Plato as being remarkable for having a very large number of divisors. It is divisible by all but one of the first 12 numbers, and all but six of the first 24 numbers. How quickly can you identify these exceptions?

The following arrangements show properties that are not unique to 7, but which it shares, to a greater or lesser extent, with other numbers.

(1234567 x 9) + 8 = 11111111
12345679 x 63 = 777777777
1 + 2 + 3 + 4 + 5 + 6 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 7^2

Here is an even more complex peculiarity:

111111 = 7 x 15873
222222 = 7 x 31746
333333 = 7 x 47619
444444 = 7 x 63492
555555 = 7 x 79365
666666 = 7 x 95328
777777 = 7 x 111111
888888 = 7 x 126984
999999 = 7 x *142857* and we arrive back at this unique number.

The number SEVEN is found in various non-thematical connections, both in classical times, and also in more recent history. There were the SEVEN Wonders of the World; there were the SEVEN Sages of Greece, the SEVEN Argive heroes, the SEVEN Hills of Rome. There were also the SEVEN Seas (the Arctic, the Antarctic, the North Pacific, the South Pacific, the North Atlantic, the South Atlantic, and the Indian Ocean).

In mystical and religious connections, the number SEVEN has played a large part, too. In the Book of Revelations, there are SEVEN Stars and SEVEN Golden Candlesticks, which represent SEVEN Churches; a lamb with SEVEN horns and SEVEN eyes which are the SEVEN spirits of God, which are also SEVEN lamps of fire; SEVEN seals of the Book; SEVEN angels; SEVEN trumpets; SEVEN thunders; SEVEN vials; SEVEN kings; and a dragon with SEVEN heads. The number 7 also seems to have held a rather special importance for St. John. The Dance of the SEVEN Veils and the SEVEN Pillars of Wisdom should also be mentioned.

One common theory for the construction of the Universe in classical times, when the Sun was supposed to be of the Heavenly Bodies circling the Earth, included seven concentric circles or 'rings' perpetually revolving round the Earth somewhere many miles up in the sky — hence the term 'Seventh Heaven'.

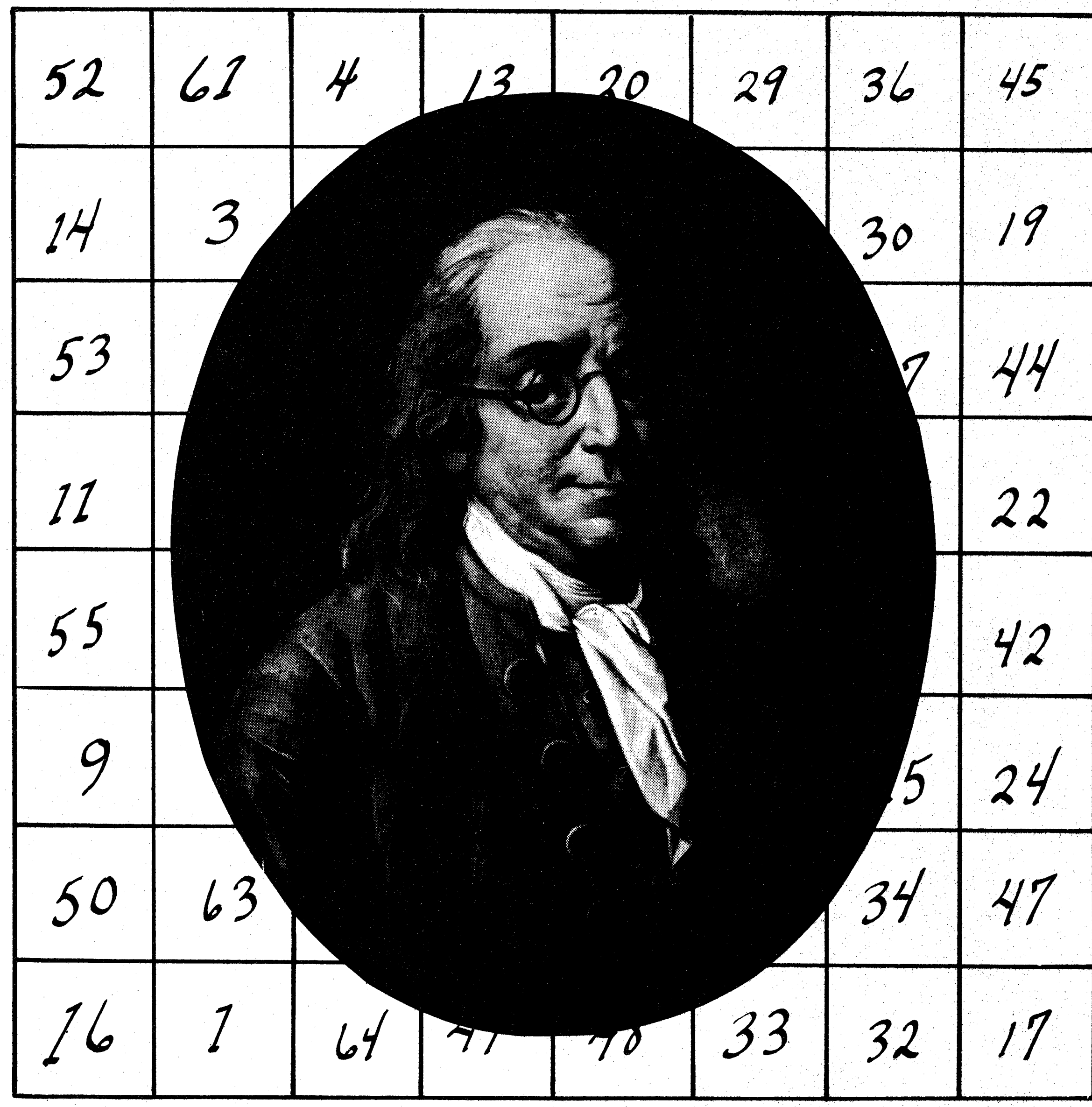
In the twentieth century, the famous physicist, Niels Bohr, who contributed so much towards the mathematics and the understanding of the structure of the atom, postulated a nucleus having, as it were, seven shells, analogous to the seven notes of the musical scale or the seven planets of antiquity.

The Seven Days of the Creation and the Seven Deadly Sins are further religious uses. In Isaiah, there are the Seven Gifts of the Holy Ghost; and in Matthew (chapter 15), there are seven loaves to feed the four thousand and seven baskets of crumbs left over. The constellation of stars, the Pleiades, is also known as the Seven Sisters; and the Seven Sisters are also the chalk cliffs just east of Cuckmere Haven, Sussex. In popular legend, there are the Seven-leagued boots of Hop o' my Thumb, and the man with Seven Wives going to St. Ives. We shouldn't forget, either, the supposed seven years' bad luck that accompanies the breaking of a mirror.

One way and another, the number 7 has had a good innings in mathematics and history. To finish off with, here is one final fraction:

$$1234567654321 = \frac{7777777 \times 7777777}{1+2+3+4+5+6+7+6+5+4+3+2+1}$$

Magic Squares



on the Computer

Donald T. Piele
Assistant Professor of Mathematics
University of Wisconsin-Parkside
Kenosha, Wisconsin 53140

Magic squares? Humbug! I've never been able to get excited over someone's special arrangement of numbers that total up to the same sum whether you add across a row or down a column or diagonally. Benjamin Franklin when first confronted with them wrote,

> ". . . it is perhaps a mark of good sense of our (English) mathematicians that they would not spend their time in things that were merely *difficiles nugae* incapable of any useful applications."(1)

Franklin had to confess, however,

> "In my younger days, having once more leisure time (which I still think I might of employed more usefully) I had amused myself in making these kind of magic squares, and, at length acquired such a knack at it, that I could fill the cells of any magic square of reasonable size with a series of numbers as fast as I could write them, disposed in such a manner that the sum of every row, horizontal, perpendicular or diagonal, should be equal; but not being satisfied with these, which I looked on as common and easy things, I imposed on myself more difficult tasks, and succeeded in making other magic squares with a variety of properties, and much more curious."(2)

In spite of the fact that I knew Benjamin Franklin had been a statesman, a scientist, a politician, a philosopher, and a writer, I was surprised to discover that playing with magic squares was also among his lengthy list of avocations. Reading further, I discovered that there are ways of testing magic squares, besides the usual rows, columns, or diagonals, that I had never seen before. For example, there are generalized diagonals, broken diagonals, corner diagonals, horizontal zig-zags, vertical zig-zags, just to name a few. Next, I found that algorithms existed for generating magic squares which looked relatively easy to program. Maybe magic squares aren't so bad after all? Besides, the computer can be programmed to do all the arithmetic and print out a listing of magical properties for each square. That did it.

I began with a generalized version of the algorithm of De la Loubère.(3)
This method fills an n x n square matrix with consecutive integers from 1 to n^2 by putting the ith integer in the matrix position P_i as follows:

1. Place the number 1 in any initial position, $P_i = (i,j)$. The standard initial position is the middle of the top row, $(1,(n+1)/2)$ for n odd.
2. Place the successive integers in vacant cells separated by jumps (A,B), $P_i = P_{i-1} + (A,B)$.
3. If P_i moves outside the square n x n matrix, adjust the coordinates modulo (1,2,3, . . . ,n) so that P_i moves back into the square. e.g. For n = 3, (2,4) = (2,1) and (0,3) = (3,3).
4. If you encounter a position P_j, $j \leq n^2$, that has already been filled, switch for one move to the rule $P_j = P_{j-1} + (C,D)$ and continue as in 2.

1. Originally from *Letters and Papers on Philosophical Subjects* by Benjamin Franklin, LL.D., F.R.S., London, 1769. See [1] p. 89.

2. See footnote 1 pp. 89-90.

3. "De la Loubère was the envoy of Louis XIV to Siam in 1687-1688, and there learnt his method." See [3] p. 195.

De la Loubère's original method specified that 1 be placed in the middle of the top row, $P_1 = (1,(n+1)/2)$, and that (A,B,C,D) be fixed at (1,1,0,—1). This is illustrated in the first sample run of the De la Loubère program. But what happens when you try different starting positions P_1 and other step values (A,B,C,D)? Will any choice of (A,B,C,D) generate a square? De la Loubère used his algorithm only for odd order squares, what happens for even order squares? Given a De la Loubère magic square, can you tell how it was generated? The second sample run shows the 5 x 5 magic square of Backet de Méziriac which was generated by choosing P_1 = (3,4) and (A,B,C,D) = (1,1,2,0). Originally it was constructed by a completely different method (see [1] p. 17). Can you find other magic squares in books or magazines that can be generated with the De la Loubère program?

For an n x n square of numbers to be considered magic it must at least have the same sum for each row and column. If the square is filled with the consecutive numbers 1 through n^2 then each row and column must add up to $n(n^2 + 1)/2$ (why?). All other ways of finding n numbers, symmetrically arranged, that add up to this sum, improves the magic square and makes it more unique. For example, a square may be summed along generalized diagonals as illustrated for a 3 x 3 square in Fig. 1. For an n x n square there are 2n generalized diagonals. The De la Loubère program checks them all in addition to the rows and columns. The best you can do, with this program, is find 4n magical properties for an n x n square.

Benjamin Franklin's magic squares are entirely different and cannot be generated by the De la Loubère algorithm. The best ones are of order 8 and 16, known as the Franklin Magic Squares. The largest one is considered among the most ingenious ever developed. It was impossible even for Franklin to be modest about it.

> ". . . you will readily allow the square of 16 to be the most magically magic of any magic square ever made by any magician."(4)

The Franklin squares are characterized by magical sums along broken diagonals that change direction halfway through the square as illustrated in Fig. 2. They can be constructed to point in four different directions; North, South, East and West. In each direction an n x n square has n broken diagonals, so it is possible to have a total of 4n magic broken diagonals. Franklin Magic Squares have the maximum number. Two other special arrangements that characterize the Franklin order 8 squares are illustrated in Fig. 3.

It is not known how Franklin generated his squares, although it is very likely that they were geometrically motivated. Several investigations have found unique symmetries in the way the numbers are arranged (see [1] p. 93 and [4]). However, there exists an analytical algorithm for reconstructing his squares called the method of *alternation with binate transposition* (see [1] pp. 100-106). It sounds difficult but it is really not. In fact the method can be easily generalized to construct much more than just the Franklin Magic Square. Since there is no difference in the algorithm for squares of order 8 or 16, I will describe, for convenience, the order 8 scheme.

4. See footnote 1 p. 93.

Begin with the *plan of construction matrix* (Fig. 4). Number the rows and columns, as usual, and let RC stand for the number at the intersection of row R and column C. The magic square is created as follows:

1. Choose a permutation of the row values, 1 through 8, and denote it by $R_1, R_2, \ldots, R_8$. For the Franklin Square choose 7,8,1,2,3,4,5,6. Let $\bar{R}_i$ be the complementary row $9 - R_i$.
2. Choose an arrangement of the column values 1 through 8, and denote it by C_1, C_2, $\bar{C}_1$, $\bar{C}_2$, C_3, C_4, $\bar{C}_3$, $\bar{C}_4$, such that $C_i + \bar{C}_i = 9$. For the Franklin Square choose 4,6,5,3,7,1,2,8. Notice that 4 + 5, 6 + 3, 7 + 2, and 1 + 8 all equal 9.
3. Rearrange the numbers in the *plan of construction matrix* as shown in Fig. 5. For example, using the row and column sequence given in 1 and 2 for the Franklin Magic Square, $R_1 = 7$ and $C_1 = 4$. Thus, $R_1C_1 = 52$ and is found in the *plan of construction matrix* at the intersection of row 7 and column 4.

Notice the repetition in C values as you move across the columns and in R values as you move down the rows. This characterizes the Franklin squares and makes the computer algorithm relatively short (see program listing).

The original Franklin Magic Square is generated in the first sample run and has a total of 50 magical sums. It has a few other nice properties too, but they are not tested for here (see [1] p. 96). What happens when you try other permutations? If you ignore the restriction on the column permutations, the computer still generates a square but some numbers will be repeated. Try it! Each new row and column permutation will generate a Franklin-like magic square. But, given a Franklin type magic square, can you find a row and column permutation that will generate it?

It probably never occurred to Franklin that anyone would want to, much less be able to, improve upon his ". . . most magically magic of any magic square." But magic square buffs are a tenacious lot and they should never be underestimated. The most obvious weakness with Franklin's square exists on the main diagonals which are not magic. Many devotees of the subject have tried in vain to remove this imperfection. In 1945, Andrew S. Anema succeeded by constructing, for the first time, a magic square that has all the Franklin properties and in addition is magic along the main diagonals and generalized diagonals (see [2]). His method uses complementary pairs and takes three pages to describe. It turns out that you can generate Anema's improved Franklin square, and many others like it, with the Franklin program described here (see sample run 2). Can you generate other improved Franklin Magic Squares? There are lots of them.

The literature on magic squares is enormous. Probably no other single recreational topic has had more written about it. With very limited experience, my impression is that many of the special methods that have been devised to construct magic squares are merely special cases of more general algorithms. Students who are interested and have a little knowledge of BASIC should be able to step into this area and, with the computer, perform a little magic of their own.

Happy hunting!

POSTSCRIPT

I'm not sure exactly how many magical properties there are in a Franklin Magic Square, but I do know that the number is much larger than I, or Franklin, ever dreamed. This became apparent one evening when I discovered, or perhaps rediscovered, 139 additional magical arrangements already present and waiting to be counted in a Franklin Square of order 8.

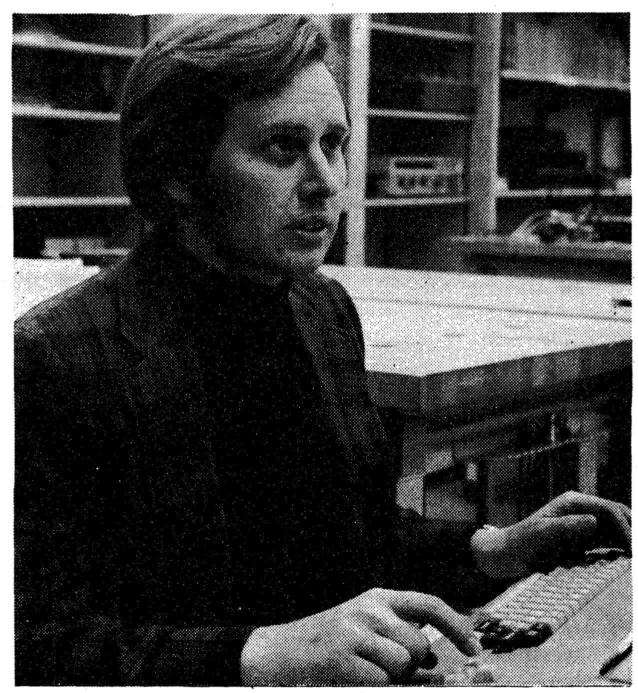
It is a relatively easy exercise to add three subroutines to the Franklin program to check these arrangements for magical sums. Can you do it? Franklin squares appear to have magical properties almost everywhere you look. Can you find other arrangements that sum to 260?

Again, Happy Hunting!

References

1. Andrews, W.S. *Magic Squares And Cubes.* The Open Court Publishing Co. 1908.
2. Anema, Andrew S. "Franklin Magic Squares." *Scripta Mathematica* 11:88-96; 1945.
3. Ball, W.W. Rouse. *Mathematical Recreations and Essays.* The Macmillan Co. New York. 1947.
4. Bragdon C. "The Franklin 16 x 16 Magic Square." *Scripta Mathematica* 4:158-60; 1936.

Photograph: Benjamin Franklin 1706-1790
Oval P.M. Alix, c. 1790, after painting by C.P.A. Van Loo, c. 1777-1785, at the American Philosophical Society, New York.



Donald Piele

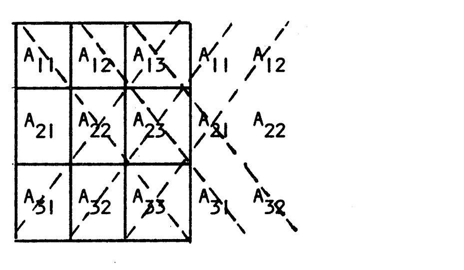


Fig. 1 Generalized diagonals for a 3 x 3 square.

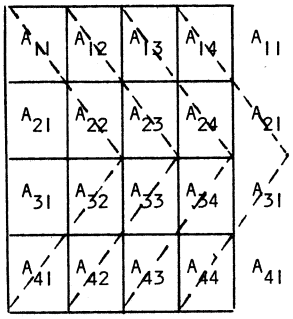


Fig. 2 Broken diagonals in one direction for a 4 x 4 square.

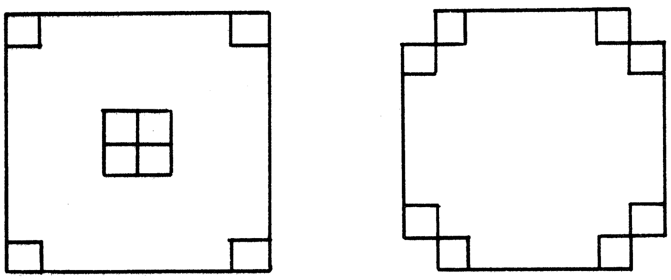


Fig. 3 Special arrangements in the Franklin Magic Square that are magic.

a. Center 4 squares and four corner squares.

b. The 4 near-corner squares.

	Columns 1	2	3	4	5	6	7	8
ROWS 1	1	2	3	4	5	6	7	8
2	9	10	11	12	13	14	15	16
3	17	18	19	20	21	22	23	24
4	25	26	27	28	29	30	31	32
5	33	34	35	36	37	38	39	40
6	41	42	43	44	45	46	47	48
7	49	40	51	52	53	54	55	56
8	57	58	59	60	61	62	63	64

Fig. 4 The Plan of Construction Matrix.

R_1C_1	$R_2\bar{C}_1$	R_3C_1	$R_4\bar{C}_1$	R_5C_1	$R_6\bar{C}_1$	R_7C_1	$R_8\bar{C}_1$
$\bar{R}_1C_2$	$\bar{R}_2\bar{C}_2$	$\bar{R}_3C_2$	$\bar{R}_4\bar{C}_2$	$\bar{R}_5C_2$	$\bar{R}_6\bar{C}_2$	$\bar{R}_7C_2$	$\bar{R}_8\bar{C}_2$
$R_1\bar{C}_1$	R_2C_1	$R_3\bar{C}_1$	R_4C_1	$R_5\bar{C}_1$	R_6C_1	$R_7\bar{C}_1$	R_8C_1
$\bar{R}_1\bar{C}_2$	$\bar{R}_2C_2$	$\bar{R}_3\bar{C}_2$	$\bar{R}_4C_2$	$\bar{R}_5\bar{C}_2$	$\bar{R}_6C_2$	$\bar{R}_7\bar{C}_2$	$\bar{R}_8C_2$
R_1C_3	$R_2\bar{C}_3$	R_3C_3	$R_4\bar{C}_3$	R_5C_3	$R_6\bar{C}_3$	R_7C_3	$R_8\bar{C}_3$
$\bar{R}_1C_4$	$\bar{R}_2\bar{C}_4$	$\bar{R}_3C_4$	$\bar{R}_4\bar{C}_4$	$\bar{R}_5C_4$	$\bar{R}_6\bar{C}_4$	$\bar{R}_7C_4$	$\bar{R}_8\bar{C}_4$
$R_1\bar{C}_3$	R_2C_3	$R_3\bar{C}_3$	R_4C_3	$R_5\bar{C}_3$	R_6C_3	$R_7\bar{C}_3$	R_8C_3
$\bar{R}_1\bar{C}_4$	$\bar{R}_2C_4$	$\bar{R}_3\bar{C}_4$	$\bar{R}_4C_4$	$\bar{R}_5\bar{C}_4$	$\bar{R}_6C_4$	$\bar{R}_7\bar{C}_4$	$\bar{R}_8C_4$

Fig. 5 Alternation with permutation scheme.

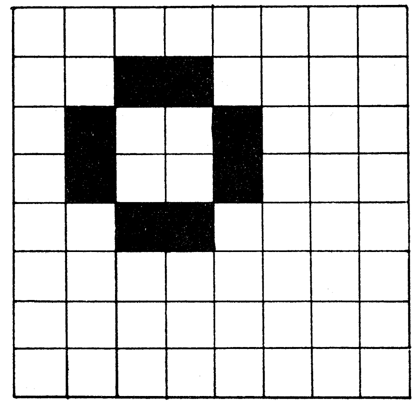


Figure 6. Magic Octagons. Each octagon arrangement sums to 260 wherever it is placed on the 8 x 8 square. There are 25 Magic Octagons.

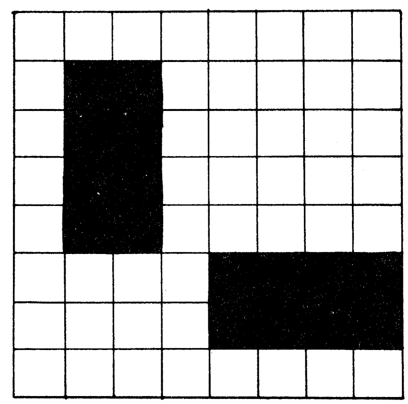


Figure 7. Magic "2 x 4's". Each 2 x 4 rectangle standing up or lying down sums to 260 wherever it is placed on the 8 x 8 square. There are 70 magic "2 x 4's".

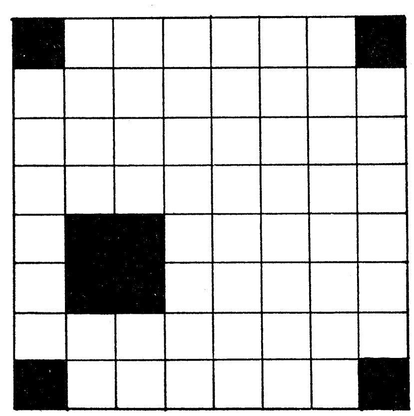


Figure 8. Any 2 x 2 square sums to 130. When it is combined with the four corner squares the sum is 260. There are 45 such arrangements but one has already been counted.

DE LA LOUBERE PROGRAM

```
10  REM PROGRAM WRITTEN BY D.T.PIELE 7/15/75
20  REM     ****** THE DE LA LOUBERE PROGRAM   ******
30  PRINT 'THIS PROGRAM TESTS NXN SQUARES OF ODD ORDER FOR '
40  PRINT 'THEIR MAGICAL PROPERTIES. YOU CAN EITHER ENTER YOUR'
50  PRINT 'OWN SQUARE OR LET THE COMPUTER GENERATE ONE FOR YOU '
60  PRINT 'USING THE ALGORITHM OF DE LA LOUBERE.'
70  PRINT 'FOR THE COMPUTER SQUARE TYPE 1. FOR YOUR OWN TYPE 2.'
80  INPUT X
90  PRINT 'HOW MANY ROWS DO YOU WANT?'
100  PRINT 'PICK AND ODD NUMBER BETWEEN 3 AND 11.'
110  INPUT N
120  DIM A[12,12],C[12,12],B[12]
130  MAT A=ZER[N,N]
140  MAT C=ZER[N,N]
150  MAT B=ZER[N]
160  IF X=2 THEN 1190
170  K=0
180  PRINT 'PICK THE POSITION I,J FOR 1. (MIDDLE OF ROW 1 IS STANDARD)'
190  INPUT I,J
200  PRINT 'CHOOSE A,B,C,D (1,1,0,-1 IS STANDARD)'
210  INPUT A,B,C,D
220  REM   ****** THE ALGORITHM FOR GENERATING THE SQUARE ******
230  K=K+1
240  A[I,J]=K
250  IF K=INT(N*N+.5) THEN 370
260  I=I-B
270  J=J+A
280  IF I=0 THEN 700
290  IF I<0 THEN 740
300  IF I>N THEN 700
310  IF J=0 THEN 720
320  IF J<0 THEN 760
330  IF J>N THEN 720
340  IF A[I,J]>0 THEN 670
350  GOTO 230
360  REM ******  END OF ALGORITHM  ******
370  PRINT 'HERE IS YOUR SQUARE OF ORDER'N
380  PRINT
390  PRINT
400  MAT  PRINT A;
410  S=INT(N*(N*N+1)/2+.5)
420  MAT C=A
430  GOSUB 790
440  R1=C
450  MAT C=TRN(A)
460  GOSUB 790
470  C1=C
480  PRINT
490  PRINT 'HERE IS A LIST OF ITS MAGICAL PROPERTIES.'
500  PRINT
510  PRINT 'ROWS AND COLUMNS:'R1+C1
520  GOSUB 920
530  D1=C+D
540  PRINT 'GENERALIZED DIAGONALS:'D1
550  PRINT
560  PRINT 'TOTAL MAGICAL SUMS: 'R1+C1+D1
570  PRINT
580  PRINT 'DO YOU WISH TO TRY AGAIN? TYPE 1 FOR YES, 0 FOR NO.'
590  INPUT Y
600  PRINT
610  IF Y=1 THEN 70
620  PRINT 'GOODBYE. SEE YOU AT THE FRANKLIN FESTIVAL'
630  PRINT 'OCT. 5 TO 11 AT UW-PARKSIDE.'
640  PRINT
650  PRINT
660  STOP
670  I=I+B-D
680  J=J-A+C
690  GOTO 280
700  I=ABS(I-N)
710  GOTO 290
720  J=ABS(J-N)
730  GOTO 320
740  I=N+I
750  GOTO 300
760  J=J+N
770  GOTO 330
780  REM  ****** TEST ROWS AND COLUMNS ******
790  C=0
800  FOR I=1 TO N
810  E=0
820  FOR J=1 TO N
830  E=E+C[I,J]
840  NEXT J
850  IF E=S THEN 880
860  GOTO 890
870  GOTO 890
880  C=C+1
890  NEXT I
900  RETURN
910  REM ****** TEST GENERALIZED DIAGONALS ******
920  C=0
930  D=0
940  FOR J=0 TO N-1
950  E=0
960  F=0
970  FOR I=1 TO N
980  R=I+J
990  T=N+1-I-J
1000  IF R <= N THEN 1060
1010  R=ABS(R-N)
1020  GOTO 1060
1030  IF T >= 1 THEN 1080
1040  T=T+N
1050  GOTO 1080
1060  E=E+A[I,R]
1070  GOTO 1030
1080  F=F+A[T,I]
1090  NEXT I
1100  IF E=S THEN 1130
1110  IF F=S THEN 1150
1120  GOTO 1160
1130  C=C+1
1140  GOTO 1110
1150  D=D+1
1160  NEXT J
1170  RETURN
1180  REM  ****** ENTER YOUR OWN SQUARE ******
1190  PRINT 'LIST THE MEMBERS OF EACH ROW SEPARATED BY A COMMA.'
1200  FOR R=1 TO N
1210  PRINT 'ROW'R
1220  MAT  INPUT B
1230  FOR I=1 TO N
1240  A[R,I]=B[I]
1250  NEXT I
1260  PRINT
1270  NEXT R
1280  PRINT
1290  GOTO 370
1300  END
```

SAMPLE RUN

1

```
THIS PROGRAM TESTS NXN SQUARES OF ODD ORDER FOR
THEIR MAGICAL PROPERTIES. YOU CAN EITHER ENTER YOUR
OWN SQUARE OR LET THE COMPUTER GENERATE ONE FOR YOU
USING THE ALGORITHM OF DE LA LOUBERE.
FOR THE COMPUTER SQUARE TYPE 1. FOR YOUR OWN TYPE 2.
?1
HOW MANY ROWS DO YOU WANT?
PICK AND ODD NUMBER BETWEEN 3 AND 11.
?3
PICK THE POSITION I,J FOR 1. (MIDDLE OF ROW 1 IS STANDARD)
?1,2
CHOOSE A,B,C,D (1,1,0,-1 IS STANDARD)
?1,1,0,-1
HERE IS YOUR SQUARE OF ORDER 3


 8     1     6

 3     5     7

 4     9     2


HERE IS A LIST OF ITS MAGICAL PROPERTIES.

ROWS AND COLUMNS: 6
GENERALIZED DIAGONALS: 2

TOTAL MAGICAL SUMS:  8

DO YOU WISH TO TRY AGAIN? TYPE 1 FOR YES, 0 FOR NO.
?1
```

2

```
FOR THE COMPUTER SQUARE TYPE 1. FOR YOUR OWN TYPE 2.
?1
HOW MANY ROWS DO YOU WANT?
PICK AND ODD NUMBER BETWEEN 3 AND 11.
?5
PICK THE POSITION I,J FOR 1. (MIDDLE OF ROW 1 IS STANDARD)
?3,4
CHOOSE A,B,C,D (1,1,0,-1 IS STANDARD)
?1,1,2,0
HERE IS YOUR SQUARE OF ORDER 5

 3     16    9     22    15

 20    8     21    14    2

 7     25    13    1     19

 24    12    5     18    6

 11    4     17    10    23


HERE IS A LIST OF ITS MAGICAL PROPERTIES.

ROWS AND COLUMNS: 10
GENERALIZED DIAGONALS: 2

TOTAL MAGICAL SUMS:  12

DO YOU WISH TO TRY AGAIN? TYPE 1 FOR YES, 0 FOR NO.
?0

GOODBYE. SEE YOU AT THE FRANKLIN FESTIVAL
OCT. 5 TO 11 AT UW-PARKSIDE.
```

FRANKLIN PROGRAM

```
10   REM PROGRAM WRITTEN BY D.T.PIELE 7/15/75
20   PRINT "THIS IS THE FRANKLIN MAGIC SQUARE PROGRAM."
30   PRINT "IT WILL GENERATE AND TEST 8X8 SQUARES. YOU "
40   PRINT "CAN ALSO ENTER AND TEST YOUR OWN 8X8 SQUARES."
50   PRINT
60   DIM A[8,8],C[8],G[8],D[8,8],B[8],E[8,8],F[8]
70   MAT  READ A,C,G
80   PRINT "FOR THE COMPUTER GENERATED SQAURE TYPE 1."
90   PRINT "TO ENTER YOUR OWN TYPE 2."
100  PRINT
110  INPUT A
120  IF A=2 THEN 1860
130  PRINT "FIRST PERMUTE THE ROWS 1 THROUGH 8 AND SEPARATE"
140  PRINT "WITH COMMAS. FOR THE FRANKLIN SQUARE CHOOSE"
150  PRINT " 7,8,1,2,3,4,5,6"
160  PRINT
170  MAT  INPUT F
180  PRINT
190  PRINT "NEXT SUPPLY A PERMUTATION OF THE COLUMNS 1 THROUGH 8"
200  PRINT "IN THE FORM A,B,C,D,E,F,G,H SUCH THAT A+C=9, B+D=9,"
210  PRINT "E+G=9, AND F+H=9.FOR THE FRANKLIN MAGIC SQUARE CHOOSE"
220  PRINT " 4,6,5,3,7,1,2,8"
230  PRINT
240  MAT  INPUT B
250  REM  ****** ALGORITHM TO GENERATE SQUARES  ******
260  J=1
270  FOR N=1 TO 7 STEP 2
280  FOR I=1 TO 7 STEP 2
290  K=I+1
300  R=B[I]
310  L=F[N]
320  S=B[K]
330  M=9-L
340  D[I,J]=A[L,R]
350  D[K,J]=A[M,S]
360  NEXT I
370  J=J+2
380  NEXT N
390  J=2
400  FOR N=2 TO 8 STEP 2
410  FOR I=1 TO 7 STEP 2
420  K=I+1
430  R=9-B[I]
440  S=9-B[K]
450  L=F[N]
460  M=9-L
470  D[I,J]=A[L,R]
480  D[K,J]=A[M,S]
490  NEXT I
500  J=J+2
510  NEXT N
520  PRINT
530  PRINT "YOUR 8X8 SQUARE IS"
540  PRINT
550  MAT  PRINT D;
560  REM  ****** END OF THE ALGORITHM ******
570  PRINT
580  FOR I=1 TO 8
590  IF F[I] <> G[I] THEN 670
600  NEXT I
610  FOR I=1 TO 8
620  IF B[I] <> C[I] THEN 670
630  NEXT I
640  PRINT " THIS IS THE BENJAMIN FRANKLIN MAGIC SQUARE OF ORDER 8."
650  PRINT
660  REM  ****** TABULATION OF THE MAGICAL PROPERTIES ******
670  PRINT "HERE IS A LIST OF ITS MAGICAL PROPERTIES."
680  PRINT
690  MAT E=D
700  GOSUB 1420
710  R1=C
720  MAT E=TRN(D)
730  GOSUB 1420
740  C1=C
750  PRINT "ROWS AND COLUMNS:"R1+C1
760  M1=0
770  M=D[1,1]+D[2,2]+D[3,3]+D[4,4]+D[5,5]+D[6,6]+D[7,7]+D[8,8]
780  IF M <> 260 THEN 800
790  M1=1
800  M=D[8,1]+D[7,2]+D[6,3]+D[5,4]+D[4,5]+D[3,6]+D[2,7]+D[1,8]
810  IF M <> 260 THEN 830
820  M1=M1+1
830  PRINT "MAIN DIAGONALS:"M1
840  G=0
850  FOR J=1 TO 7
860  E=0
870  F=0
880  FOR I=1 TO 8
890  R=I+J
900  T=9-I-J
910  IF R <= 8 THEN 930
920  R=R-8
930  E=E+D[I,R]
940  IF T >= 1 THEN 960
950  T=T+8
960  F=F+D[T,I]
970  NEXT I
980  IF E <> 260 THEN 1000
990  G=G+1
1000  IF F <> 260 THEN 1020
1010  G=G+1
1020  NEXT J
1030  PRINT "GENERALIZED DIAGONALS:"G
1040  MAT E=D
1050  GOSUB 1520
1060  B1=C
1070  MAT E=TRN(D)
1080  GOSUB 1520
1090  B2=C
1100  MAT E=D
1110  GOSUB 1660
1120  B3=C
1130  MAT E=TRN(D)
1140  GOSUB 1660
1150  Y=B1+B2+B3+C
1160  PRINT "BROKEN DIAGONALS:"Y
1170  C=0
1180  D=0
1190  E=D[1,1]+D[1,8]+D[8,1]+D[8,8]+D[4,4]+D[5,4]+D[4,5]+D[5,5]
1200  C=1
1210  PRINT "SPECIAL CASES:"
1220  PRINT "         CENTER FOUR SQUARES PLUS FOUR CORNER SQUARES."
1230  E=D[1,2]+D[2,1]+D[1,7]+D[2,8]+D[7,1]+D[7,8]+D[8,2]+D[8,7]
1240  IF E <> 260 THEN 1270
1250  PRINT "         THE FOUR CORNER DIAGONAL PAIRS."
1260  D=1
1270  PRINT
1280  W=R1+C1+Y+C+D+M1+G
1290  PRINT "TOTAL MAGICAL SUMS:"W
1300  PRINT
1310  PRINT "DO YOU WANT TO TRY AGAIN? TYPE 1 FOR THE COMPUTER SQUARE,"
1320  PRINT "TYPE 2 TO ENTER YOUR OWN SQUARE, AND TYPE 0 TO STOP."
1330  PRINT
1340  INPUT Z
1350  IF Z=1 THEN 130
1360  IF Z=2 THEN 1860
1370  PRINT "GOODBYE. SEE YOU AT THE BENJAMIN FRANKLIN FESTIVAL"
1380  PRINT "AT UW-PARKSIDE, OCTOBER 5 TO 11."
1390  PRINT
1400  STOP
1410  REM  ****** SUBROUTINES TO CHECK FOR MAGICAL SUMS ******
1420  C=0
1430  FOR I=1 TO 8
1440  E=0
1450  FOR J=1 TO 8
1460  E=E+E[I,J]
1470  NEXT J
1480  IF E <> 260 THEN 1500
1490  C=C+1
1500  NEXT I
1510  RETURN
1520  C=0
1530  FOR J=0 TO 7
1540  E=0
1550  FOR I=1 TO 4
1560  R=I+J
1570  T=9-I
1580  IF R <= 8 THEN 1600
1590  R=R-8
1600  E=E+E[I,R]+E[T,R]
1610  NEXT I
1620  IF E <> 260 THEN 1640
1630  C=C+1
1640  NEXT J
1650  RETURN
1660  C=0
1670  FOR J=0 TO 7
1680  E=0
1690  FOR I=1 TO 4
1700  R=9-I
1710  T=9-I-J
1720  IF T >= 1 THEN 1740
1730  T=T+8
1740  E=E+E[T,I]+E[T,R]
1750  NEXT I
1760  IF E <> 260 THEN 1780
1770  C=C+1
1780  NEXT J
1790  RETURN
1800  DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
1810  DATA 17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32
1820  DATA 33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48
1830  DATA 49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64
1840  DATA 4,6,5,3,7,1,2,8,7,8,1,2,3,4,5,6
1850  REM  ****** ENTER YOUR OWN SQUARE ******
1860  PRINT
1870  PRINT "LIST THE MEMBERS OF EACH ROW SEPARATED BY COMMAS."
1880  FOR R=1 TO 8
1890  PRINT "ROW"R
1900  PRINT
1910  MAT  INPUT B
1920  FOR I=1 TO 8
1930  D[R,I]=B[I]
1940  NEXT I
1950  PRINT
1960  NEXT R
1970  GOTO 650
1980  END
```

SAMPLE RUN

1

```
THIS IS THE FRANKLIN MAGIC SQUARE PROGRAM.
IT WILL GENERATE AND TEST 8X8 SQUARES. YOU
CAN ALSO ENTER AND TEST YOUR OWN 8X8 SQUARES.

FOR THE COMPUTER GENERATED SQAURE TYPE 1.
TO ENTER YOUR OWN TYPE 2.

?1
FIRST PERMUTE THE ROWS 1 THROUGH 8 AND SEPARATE
WITH COMMAS. FOR THE FRANKLIN SQUARE CHOOSE
 7,8,1,2,3,4,5,6

?7,8,1,2,3,4,5,6

NEXT SUPPLY A PERMUTATION OF THE COLUMNS 1 THROUGH 8
IN THE FORM A,B,C,D,E,F,G,H SUCH THAT A+C=9, B+D=9,
E+G=9, AND F+H=9.FOR THE FRANKLIN MAGIC SQUARE CHOOSE
 4,6,5,3,7,1,2,8

?4,6,5,3,7,1,2,8

YOUR 8X8 SQUARE IS

 52   61   4    13   20   29   36   45

 14   3    62   51   46   35   30   19

 53   60   5    12   21   28   37   44

 11   6    59   54   43   38   27   22

 55   58   7    10   23   26   39   42

 9    8    57   56   41   40   25   24

 50   63   2    15   18   31   34   47

 16   1    64   49   48   33   32   17

 THIS IS THE BENJAMIN FRANKLIN MAGIC SQUARE OF ORDER 8.

HERE IS A LIST OF ITS MAGICAL PROPERTIES.

ROWS AND COLUMNS: 16
MAIN DIAGONALS: 0
GENERALIZED DIAGONALS: 0
BROKEN DIAGONALS: 32
SPECIAL CASES:
         CENTER FOUR SQUARES PLUS FOUR CORNER SQUARES.
         THE FOUR CORNER DIAGONAL PAIRS.

TOTAL MAGICAL SUMS: 50

DO YOU WANT TO TRY AGAIN? TYPE 1 FOR THE COMPUTER SQUARE,
TYPE 2 TO ENTER YOUR OWN SQUARE, AND TYPE 0 TO STOP.
```

2

```
?1
FIRST PERMUTE THE ROWS 1 THROUGH 8 AND SEPARATE
WITH COMMAS. FOR THE FRANKLIN SQUARE CHOOSE
 7,8,1,2,3,4,5,6

?1,2,8,7,3,4,6,5

NEXT SUPPLY A PERMUTATION OF THE COLUMNS 1 THROUGH 8
IN THE FORM A,B,C,D,E,F,G,H SUCH THAT A+C=9, B+D=9,
E+G=9, AND F+H=9.FOR THE FRANKLIN MAGIC SQUARE CHOOSE
 4,6,5,3,7,1,2,8

?1,2,8,7,5,6,4,3

YOUR 8X8 SQUARE IS

 1    16   57   56   17   32   41   40

 58   55   2    15   42   39   18   31

 8    9    64   49   24   25   48   33

 63   50   7    10   47   34   23   26

 5    12   61   52   21   28   45   36

 62   51   6    11   46   35   22   27

 4    13   60   53   20   29   44   37

 59   54   3    14   43   38   19   30

HERE IS A LIST OF ITS MAGICAL PROPERTIES.

ROWS AND COLUMNS: 16
MAIN DIAGONALS: 2
GENERALIZED DIAGONALS: 14
BROKEN DIAGONALS: 32
SPECIAL CASES:
         CENTER FOUR SQUARES PLUS FOUR CORNER SQUARES.
         THE FOUR CORNER DIAGONAL PAIRS.

TOTAL MAGICAL SUMS: 66

DO YOU WANT TO TRY AGAIN? TYPE 1 FOR THE COMPUTER SQUARE,
TYPE 2 TO ENTER YOUR OWN SQUARE, AND TYPE 0 TO STOP.

?0
GOODBYE. SEE YOU AT THE BENJAMIN FRANKLIN FESTIVAL
AT UW-PARKSIDE, OCTOBER 5 TO 11.
```

Pocket Calculator Tricks !

0 1 6

Punch these problems into your pocket calculator, then turn it around (180°) to read the answer. For loads more of calculator problems, see the four calculator books in the *Creative Computing* Library advertisement.

An Ancient Arab Proverb:
0.1283 x 3 + 47 x 15

Where?
71 x 2 + 0.15469 x 5

And Then What?
121 x 57 + 0.25 x 16 ÷ 2

John Jackobs
Heidelberg College

The Stock Market Is Dropping!
$(508^2 - 16^2 + 5^2 + 2) \times 0.03$

Familiar Principle:
$(.844561)^{0.5}$

That's A Big One!
$50 \times 125^2 - 269^2 + 120$

POOR HOUSE

If you buy 100,000 shares of IBM stock (ENTER 100000) on margin at $148.18 per share (ENTER x 148.18), pay $472 commission (ENTER + 472), and the price goes down 25% (ENTER x 0.25), what do you find yourself in?

David Ahl

Non-Usual Mathematics for Computer Solution

James Reagan
Stevenson High School, Sterling Heights, Michigan

Introduction

Mathematics instruction generally proceeds sequentially and deductively. This instructional procedure creates some misconception of the mathematics. Mathematics is not totally deductive logic; the deductive proof of any hypothesis is developed after one has become quite certain that the conjecture is true. One investigates enough specific cases to become somewhat sure that the observed cases generalize or that the proper limits on the conjecture have been found. Thus, there is a contradiction between the mathematics in its instruction and mathematics in its historical development. In formal instruction in mathematics the discovery of the theorems, rules, and properties are taught as though they were bestowed upon man as were the two tablets containing the Ten Commandments; the time and effort expended are seldom discussed. Most mathematics courses offer the student the deductive process in developing the material when, historically, the deductive process was employed late in the development of the topic.

Because of mathematics instruction's dependence upon deductive development, certain topics fall before or after certain other topics; and mathematics instruction has become characterized by its sequential approach. It is true that there are certain foundations upon which some topics rest; these pre-requisites are necessary for the development of the vocabulary and the organization of latter theorems. With the use of computers in many schools, some of the latter topics can be studied out of sequence.

Agreed, there is what might be called mathematics sophistication before one can *master* certain topics, but how much mathematics sophistication is required to *understand* and *appreciate* the material? It is this writer's experience that students in the secondary school can investigate topics and solve problems prior to the traditional time location of the topic or problem in the instructional sequence. Many topics commonly deferred to the college curriculum are suitable and interesting for the secondary school student.

What follows in this series are examples of such problems that have been studied and solved by high school students in computer programming classes at Stevenson High School and many other high schools having computer access.

Infinitely Many Primes

Background of the Problem

The great mathematician of the third century B. C., Euclid, proved that there are infinitely many primes. Euclid's proof leads to interesting problems some 2000 years later.

First the proof and then the problems.

The proof is by *reductio ad absurdum,* an indirect proof.

Suppose there are finitely many prime numbers. Then, these *n* primes can be listed in order.

$$2, 3, 5, \ldots, P_n.$$

Form a number N by adding 1 to the product of the *n* primes:

$$N = 2*3*5*\ldots*P_n + 1.$$

Either N is prime or N is composite. Each of these results for N leads to a contradiction that P_n is the largest prime.

First, if N is prime, then it is clearly greater than P_n and P_n is not the greatest prime.

Second, suppose N is composite. It has a prime factor *p*. This prime factor *p* cannot be one of the primes $2, 3, 5, \ldots, P_n$, since dividing each of the primes in the list into N leaves a remainder of 1. Thus, *p* must be a prime greater than P_n.

Therefore, there are infinitely many prime numbers. QED.

Statement of the Problem

The creation of the number N in the proof by Euclid leads to many interesting questions.

Create a set of numbers by the recursive definition:

$$P_1 = 2$$
$$P_2 = 3$$
$$P_3 = P_1*P_2 + 1 = 2*3 + 1 = 7$$
$$P_4 = P_1*P_2*P_3 + 1 = 2*3*7 + 1 = 43$$
$$\vdots$$
$$P_{n+1} = P_1*P_2*P_3*\ldots*P_n + 1.$$

Are each of the numbers in the set prime?

If some number in the list is not prime, is a prime factor of it greater than the preceding number in the list? For example, if P_6 is composite, is a prime factor of P_6 greater than P_5?

These same questions apply to a second set that can be created by subtracting 1 instead of adding 1 to the previous list product.

Create a second set as follows:

$$P_1 = 2$$
$$P_2 = 3$$
$$P_3 = P_1*P_2 - 1 = 2*3 - 1 = 5$$
$$\vdots$$
$$P_{n+1} = P_1*P_2*\ldots*P_n - 1.$$

Answer the same questions as with the first set.

Finally, create a third set and investigate.

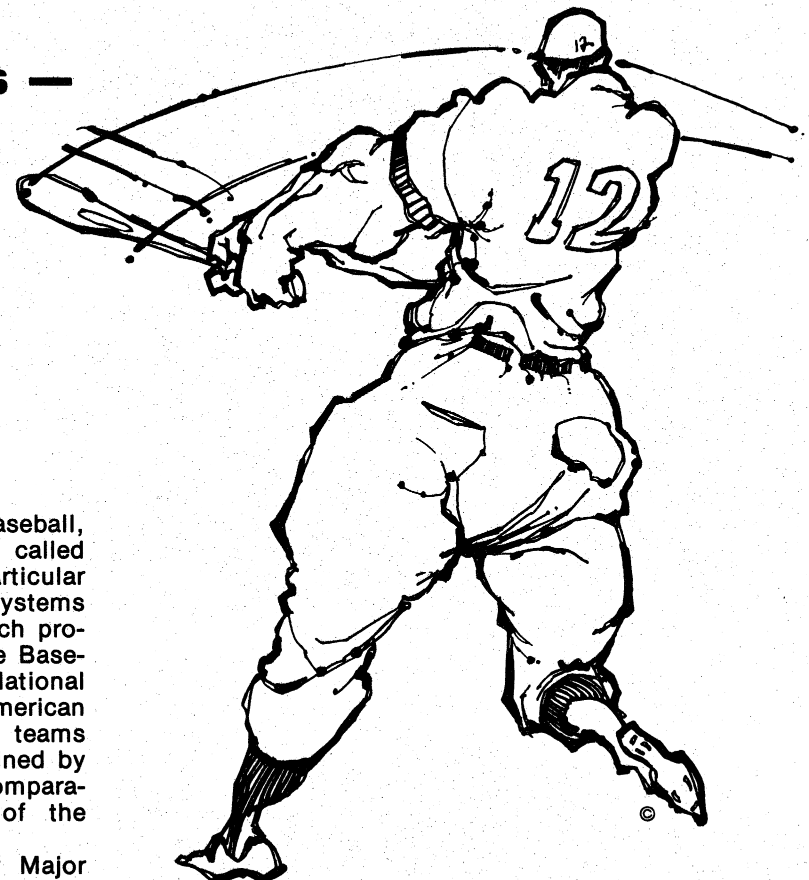
$$P_1 = 2$$
$$P_2 = 3$$
$$P_3 = 5$$
$$P_4 = P_1*P_2*P_3 + 1 = 2*3*5 + 1 = 31$$
$$\vdots$$
$$P_{n+1} = P_1*P_2*\ldots*P_n + 1.$$

Hints

1. One of the constraints on the computations is the number of significant digits of the computing machine. The numbers in each of the sets become large rapidly and can soon overflow the significant digit capacity of the machine. An extended precision routine may be needed to investigate far into the sets. [See "Computing Factorials — Accurately" by Walter Koetke, *Creative Computing* Vol 1, No 3, pp 9–11.]

2. To save on variable storage during execution of the program, instead of using high dimensioned vectors, it might be helpful to create a file in which to store new numbers in the set and from which to read out previous numbers.

The World of Series — Playoff That Is



James Reagan
Stevenson High School
Sterling Heights, Mich.

What Are Playoffs?

Traditionally the professional sports of baseball, basketball, and hockey have held what are called championship playoffs to determine the particular sport league champion. Comparable playoff systems are used after the regular season ends by such professional sport organizations as Major League Baseball, the National Football League, the National Hockey League, and the National and American Basketball Associations. An even number of teams qualify for the championship playoffs determined by their regular season record and/or by their comparative standing within a particular division of the league.

The playoff setup may be illustrated by Major League Baseball. Each of the division winners qualifies for the playoffs. The winners of the two American League divisions, East and West, play each other in a best-of-five games series while the National League divisional winners play their best-of-five series. (In a best-of-five series the first team to win three games is the winner of the series.) The winner of the World Series is considered the best team in baseball.

There has been a trend in the major professional sports to expand—add teams to the respective leagues. This has changed many previous playoff systems to include more teams qualifying for the championship playoffs. For example, the National Basketball Association now has a first round series in which there are two best-of-three series before the semi-final and final playoffs.

Some of the criticism of professional sport expansion is that the entire championship structure is designed to add revenue to the pockets of the team owners and the best team is not necessarily the one that wins.

Let us investigate some of the questions and results of employing a championship playoff system.

First Investigation

If we consider that playoff series are a money generating activity, how much money can the competing teams, not necessarily the players, expect to earn? The earnings are reflected in the number of games that can be expected to be played. In a best-of-seven series, there could be four, five, six, or seven games played. Which number of games should we expect?

Certainly games cannot or should not be fixed so that there can be more games in the series; that is both illegal and unethical. Just let things go naturally and see what happens.

We can use the computer to simulate each game of the series and keep track of how many games it takes to win each of several series simulations. For each game we will presently assume each team is equally likely to win, just as flipping a coin is usually considered equally likely to be a head or a tail. When a team wins four games the series is over and we tally the number of games played in the series. If the series ends in four games, we increase the count for four game series by one; similarly, we add to the counts for series ending in five, six, or seven games. After a large number of playoff series simulations, perhaps a hundred, we will have some idea of the expected number of games for a best-of-seven game series. The following questions arise and can be answered from the simulations:

1. Is a prediction of a "four game sweep" reasonable?
2. Is a prediction that the series will go six or seven games really going out on a limb?
3. How well do the results of the simulation agree with actual outcomes of Major League World Series or other best-of-seven championship playoffs?

Many preliminary playoffs are not best-of-seven. Some are best-of-five and some are best-of-three. How should the series results be expected to be distributed with these kinds of playoffs?

We originally assumed that the probability that a team would win any game was 0.5; but, seldom are the two teams equally likely of winning a given game. There are many factors affecting the *a priori* probability of a team winning a game. Some of the factors are:

1. The place where the game is played. Does the home team have the advantage?

2. The season records of the teams. The team with the better season record may have a better than 0.5 probability of winning a game.
3. The winner of the previous game. A psychological advantage is usually associated with the team that has won the previous game.

An investigation stating the probability influencing factors and the simulation will lead to some interesting results.

Second Investigation

An equally important investigation is one which will help answer the question: "Just because Team A wins a series, does that mean that Team A is the better of the two teams?" Certainly if we agree to use the criterion that the winner is the better team the answer to the question is "Yes". But, most sports fans have a preconceived idea of which team is the better of the two. Now I rephrase the question: "If Team A is better than Team B, what is the probability that Team A will win a best-of-seven series from Team B?"

By Team A being better than Team B I mean that for any game that they play, the probability that Team A will win, P(A), is greater than the probability that Team B will win, P(B). If we first consider P(A) = 0.55, what is the probability that Team A will win the series? The question can be answered using various probabilities for Team A.

Further, if Team A is better than Team B, what is the probability that Team A will win a best-of-five series? I recently heard a sportscaster say, "Anything can happen in a five game series." Finally, what happens in a best-of-three series as in the NBA?

How Many Ways Can You Make Change For A Dollar?

by Brian Hess

Even with all the nickels and dimes and pennies running around this country, somebody still always needs change for a dollar for this or that. Assuming that you carried around enough change, how many ways could you help out someone who needed the right change for some infernal vending machine? You can use half-dollars, quarters, dimes, nickels, or pennies to make the right change. For example, you could give him 1 half, 1 quarter, 1 dime, 2 nickels, and 5 pennies. Get it?

There are a few different ways to solve the problem. One is to break it down into smaller problems, easily solved (e.g., how many ways can you make change for a quarter?) and then combine the answers to get the "big" answer. Another mathematical method would be to write out a series of equations relating each piece of change to each other and the dollar and then solve them. Finally, you could do the problem by exhaustion.

Solving a problem by exhaustion means writing down all the answers until all the possibilities of solution are exhausted (or until you are exhausted, whichever comes first). Fortunately, you *Creative Computing* readers can exhaust a computer rather than yourselves. Write a program to figure out how many ways you can make change for a dollar. Print the ways as well as a final total. (WARNING: Printing takes time on a TTY—if you are in a hurry [or being charged] don't bother printing all the ways, just the final total.)

Hints: 1) If you use loops, counting one by one, it will probably take close to 20 minutes to compute all of it (even without printing all the combinations). Do you have to index the "pennies-counter" by one? Once the half-dollar counter reaches 2, what happens to all the other nested coin-counters? What about 4 quarters? Dimes?

2) If your program doesn't come out with well over 100 ways to make change for a dollar, it has something wrong with it. (I'm not going to tell you the exact answer—work it out for yourself!)

3) Once you have gotten the answer, ask some friends to guess at what they think it is. You'll hear some very interesting numbers. Use the computer to tabulate them, etc.

4) Write some sort of applications changes for this program. Look at how the number of combinations changes. For example, nobody uses half-dollars in vending machines, so restrict the number of halves to 1. Also, who wants more than 25 pennies? Only parking meters and gumball machines use them. Finally, include at least 1 dime in the change so that your changeless friend can make a phone call!

Brian is a high school student in Western Springs, Illinois.

Sequences

Becky Jessen, Senior
Sylmar High School Sylmar, California

Shortly before World War II, chain letters became the rage. If you receive one, typically you'll find five names in the letter. You will be asked to send an amount of money to the person whose name is at the top of the list. You then cross this name off the list, add your name to the bottom and send five copies to friends. Suppose the amount is $1.00 and let's ignore postage, which is up a bit, since the 1930's. How could you make out?

You send out 5 letters with your name at the bottom. If your friends follow through, together they'll send out 5^2 or 25 letters, with your name fourth. If their friends continue, 5^3 or 125 letters will be sent with your name third.

$$5^1, 5^2, 5^3, \; 5^4, \; 5^5$$
$$5, 25, 125, 625, 3125$$

If everyone followed through, you'd receive $3,125. Do you think this idea would work? It did in the 1930's for a few, but it's illegal, now. Can you see the problems? If it were possible to do, how many names would need to be on the list for you to collect at least a million dollars? How about a billion dollars?

This kind of sequence, 1, 5, 25, , . . . in which each term is a constant multiple of the previous term, is called a geometric sequence. The simplest of these is
1, 2, 4, 8, 16, 32, . . .
Here, the constant multiplier is 2.

An ancient legend has it, that a man given his choice of anything he desired, by a very rich king, asked for one grain of wheat on the first square of a chessboard, 2 on the second, 4 on the third, and so forth. How many grains of wheat would he have received, had his request been granted? (Rumor has it that the king found it better to take off a head, rather than provide the grain.) Find an estimate for the size of wheat and the number of grains that can be stored in a cubic foot. Then determine the amount of ground that could be covered a foot high with the amount of wheat requested.

A more recent question, based on this same sequence, is, "Which would you rather be paid; $100. per day or 1¢ the first day, 2¢ the second, 4¢ the third, and so forth, for 30 days?" Which one would you choose? Would your choice be the same if the daily rate was $1000. per day?

You had two parents, each of them had two parents (you had four grandparents), and so forth. At some point, the number of your ancestors would exceed the total world population today. In what past year, would your original ancestors have been born, assuming this ideal model? How many people were on the earth in the year you have found? What do these results mean?

An arithmetic sequence grows very slowly. The sequence of even numbers, for example, is
2, 4, 6, 8, 10, 12, . . .
The geometric sequence grows much faster.
1, 2, 4, 8, 16, 32, . . .
Here is another sequence, called the Fibonacci sequence.
1, 1, 2, 3, 5, 8, . . .

Each term is the sum of the two preceding terms. Which of the three sequences grows the fastest? How does the Fibonacci sequence compare to the arithmetic sequence? Is there an arithmetic sequence that will grow faster than the Fibonacci?

The Fibonacci sequence is of great interest to mathematicians. Apparently there is no end to the neat things to be discovered about it. For example, $1 + 1 + 2 + 3 + 5 + 8 = 20$. What is the eighth term of the sequence? Compare the sum of the first 20 terms with the 22nd term. Will this relationship always hold true?

Here is a pattern.

$1^2 + 1^2 = 1 \cdot 2$
$1^2 + 1^2 + 2^2 = 2 \cdot 3$
$1^2 + 1^2 + 2^2 + 3^2 = 3 \cdot 5$

Do you see the pattern? Do you think it continues? How far?

The Greeks were fascinated by "The Golden Ratio". Pages of books and shapes of pictures tend to have sides whose measures have approximately this ratio. Many artists, through trial and error, have been led to this ratio, while searching for ways to present pleasing patterns and correct projections. Any consecutive terms of the Fibonacci sequence, approximate this ratio. And the accuracy increases the larger the terms used. Have a computer list the sequence of ratios, comparing consecutive terms of this sequence. See if you can find a good approximation for the golden ratio, perhaps five or six decimal places.

Sequences continue to be popular with mathematicians, amateur and professional alike. In fact, amateur mathematicians have made some startling discoveries in this area. A computer gives us the power to examine large numbers of terms and allows us to consider things we never could do, if all we had was paper and pencil.

References to consider:

- Andree, Richard, *Computer Programming and Related Mathematics,* John Wiley & Sons, Inc., New York, N. Y., 1967.
- Jacobs, Harold R., *Mathematics: A Human Endeavor,* W. H. Freeman & Co., San Francisco, Ca., 1970.
- "The Fibonacci Quarterly", St. Mary's College, California.

Progression Problems

Charles A. Reeves
Tallahassee, Florida

Last week we grew paramecium in a hay infusion, as described in the experiment from our science book (*Today's Basic Science*). We put some hay in a bucket of tap water, and left it sitting by itself for 7 days. At the end of that time, we had a bucket full of the things.

The book also mentioned that paramecium reproduce by cell division about every 5 hours. Assume that there was only 1 paramecium in the bucket when we started – how many would there be at the end of the 7th day?

For those who want more: Have the computer print the number of paramecium at the end of the 4th, 5th, 6th, and 7th days, all in one run!

On page 194 of *Today's Basic Science,* you will find:
"The female grasshopper is especially adapted for egg-laying. The female lays from 20 to 100 eggs. It lays the eggs in the ground or perhaps in a rotted log. A structure at the tip of the abdomen enables the female to dig a hole in the ground or in rotted wood. This structure is called the 'ovipositer'."

Assume for a moment that you are a scientist, doing an experiment with grasshoppers over a ten-year period. You are applying for a grant from the U. S. government, and so you have to plan how much money you will spend on food, tags, etc. for these animals.

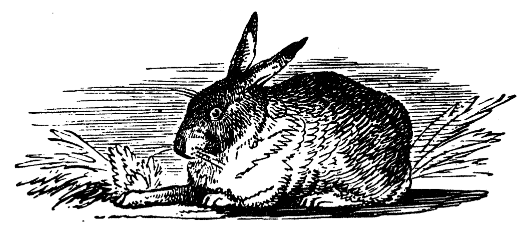
You have to first find out how many grasshoppers you will have in a ten year period (you are starting the experiment with only 1 pair, a male and a female). Have the computer calculate and report to you approximately how many grasshoppers will be born from that one pair. Grasshoppers live only one year, so the females will lay eggs only once in their lives. Assume also that half of those born will be males.

Write a program that you can use to find the average of a given set of numbers. We will use this program to find the class average on tests, and to find the average height and weight of the class. You will want to tell the computer to save this program for future usage.

It takes nature about 500 years to produce 1 inch of topsoil. Many years ago our country had an average depth of almost 9 inches of this good dirt, but now we are down to 6 inches. This type of dirt is necessary, of course, for growing food.

Careless management of our soil causes about 1% per year to erode away, and then it's lost forever. Once we get down to less than 3 inches, it will be impossible to grow crops on a major scale. Have the computer calculate and report to you the year that our country will have less than 3 inches of topsoil, assuming that it continues to erode away at 1% per year.

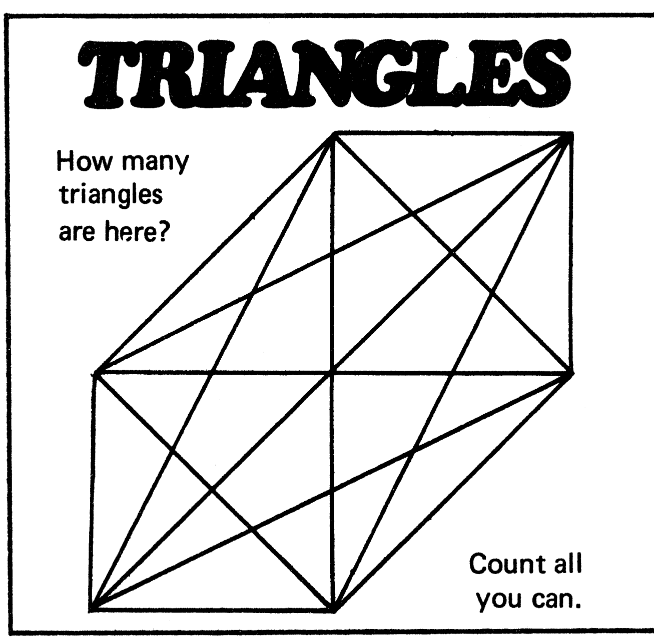
Will you be alive then? Will your children be alive?



Jack got a pair of bunnies as a New Year's Day present in 1972. This pair became a pair of young rabbits in February, and a pair of adult rabbits in March. A pair of adult rabbits produces a pair of bunnies each month from then on, and this growth cycle continues. The number of pairs of rabbits of each type is provided below, for the first six months:

	J	F	M	A	M	J	
Pairs of bunnies	1	0	1	1	2	3	...
Pairs of young	0	1	0	1	1	2	...
Pairs of adults	0	0	1	1	2	3	...
Total pairs	1	1	2	3	5	8	...

Have the computer tell you the maximum number of rabbits that Jack could have in three years. [Saving the "maximum number" means we are assuming that, of each pair born, one is a male and one a female, and that none of the rabbits die over this period of time.]



Seeing is Believing but Simulating is Convincing

by Walter Koetke, Lexington High School

Simulation represents one of the more promising areas in which the computer can be used to provide pertinent data on the options facing our society. Many computer related texts, however, pursue the subject no further than Buffon's needle, playing craps or dealing cards. While these are indeed valid examples of simulation, they are not sufficient because: they do not clearly demonstrate a potential connection between simulation results and human decision making; they seem to associate simulation with theoretical problems of mathematics as opposed to real problems of society; and they are too simple to give a feeling for the true complexity of societal simulations. More elaborate simulations are available (such as the variety of material from the Huntington Project), but these are intended for students to execute rather than write. The following two problems are neither outstanding problems of society nor outstanding problems of mathematics. They are offered because they provide the student with the opportunity to write simulations that go just a little further than the standard examples.

Problem 1: Horse Ranch. Suppose you are a rancher and own 200 horses. All of your stock is healthy, and that's very important because you expect a buyer to arrive unannounced sometime in the next 10 days. The buyer is looking for 160 healthy horses. You know from experience that if he finds fewer than 160 healthy animals, he won't buy any at all. Clearly your finances could not absorb the complete loss of the entire sale.

Just as you're ready to celebrate the pending sale, you learn that one of your horses has "day cough", a terrible sounding disease of short duration. Day cough lasts exactly one day and immunity to re-exposure results. You know that each sick horse will contact five other horses each day, and each contact has 0.6 chance of transmitting the sickness if the contacted horse is healthy and not immune. Although your situation appears bleak, do you really have such a serious problem? Which of the following three alternatives is your best course of action and why is it best?

a) Stop worrying. There's really very little chance that fewer than 160 horses will be healthy during the next 10 days.

b) Solicit the expensive support of modern medical science. "Cough shots" are available that are supposed to provide instant immunity when given to a healthy horse. However, only 13 horses per day can be innoculated by the local veterinarian, and neither he nor you can tell a healthy horse from a horse already immune. Thus you may innoculate an already immune horse. The only problem in doing this is that one of the cough shots is wasted. Actually, the cough shots have a 90% chance of providing instant immunity and a 10% chance of instantly giving day cough to a healthy horse. For all his expertise and advice, the veterinarian will charge you one healthy horse for each day of service he provides.

c) By calling right away, you can probably convince the buyer to delay his unannounced visit for 10 days. He will still come unannounced, but during the period 10 to 20 days from now rather than during the next 10 days.

Problem 2: Fish Pond. As part of a conservation and ecology project, a group of biology students has designed the following controlled experiment: A small pond is polluted with several common types of waste material. Exactly 100 male fish are then introduced to the previously fish free pond. Each day the students carefully net exactly 10 fish. All 10 fish are caught simultaneously and at the same time each day. The netted fish are examined for signs of gill disease, their tails dyed, and they are returned to the pond. The dye used is harmless and completely disappears after 13 full days in the water. If a netted fish is already dyed, it is dyed again so that it too will remain dyed for the next 13 days. The experiment continues until each of the fish netted on any one day all have dye on their tails.

How many days should the students allow to permit "a reasonable chance" for successful completion of the experiment?

A complete solution to this problem, as in many problems of society for which computer simulations might be useful, requires clarification of some human values. The definition of "a reasonable chance" is a personal one, and as such it is based on a wide variety of factors. Discussing the point alone makes this a very worthwhile problem in a classroom setting.

A closely related simulation problem can be described by not stating the number of fish in the pond, but instead specifying the number of days that pass before the 10 fish netted all have dye on their tails. The question then becomes "How many fish were originally in the pond?" For instance, if the experiment described was completed in 32 days, how many fish were originally placed in the pond?

Problem 3: Superspy. This one is included just for fun. The problem is really another disguise of the two dimensional random walk.

Each night IBF, the superspy, leaves his daytime refuge and emerges from a secret manhole cover in the center of a city. Being an exceptionally tricky spy, IBF is likely to sneak forward, backward, left or right at every intersection. If IBF happens to accidentally stray 8 blocks from the manhole (in any direction) he is captured by the arch enemies of society, the TUVEFOUT. If IBF happens to return to his manhole, he is safe for another night. What is the probability that IBF will return safely and thus complete a successful mission? What is the average number of blocks traveled during a mission?

COMPUTER GENERATED AIDS TO TEACHING GEOMETRIC CONCEPTS

by
Dr. Bruce H. Barnes
and
Frederick R. Stocker
The Pennsylvania State University

INTRODUCTION

One of the most difficult tasks for any classroom instructor is to diagramatically illustrate mathematical and geometric concepts with a blackboard and chalk as the medium of description. Frequently accuracy gives way to hand waving, especially when the concepts involved are three-dimensional in nature or involve a high degree of mathematical accuracy in representation. That a picture is worth a thousand words or handwaves is as true in the classroom as anywhere else. One solution to this problem is to use the computer to supply a textbook base of programs which provide graphical output with a high degree of visual and mathematical accuracy. With static motion* or dynamic motion added, third (structural for example) and fourth (rotation in time for example) dimensional concepts can be made to stand out. This discussion offers some approaches to the use of computer graphics, as an answer to the blackboard and chalk problem. The use of graphics in elementary mathematics is presented by matrices and transformations, while Fourier Series is used as a representative of advanced mathematics.

COMPUTER GRAPHICS

A large diversity of hardware has been developed or used in support of graphic image presentation including cameras and film, overhead projectors, computer high-speed line printers, pen and ink plotters and cathode ray tube displays. Many techniques have been developed to present information for assimilation using devices such as those mentioned. Educational television makes good use of film. Computer assisted instruction uses some combinations of the above mentioned devices. Many college courses have made use of such devices to some extent. This discussion will center around the high speed line printers, pen and ink plotters and cathode ray tube devices as mediums of communication.

Every computer installation has some type of line printer for output. These devices print lines of text on paper varying from standard 8½ by 11" to large format computer printout pages. If the printed lines are patterned or overstruck with several characters, then plots may be generated or geometric shapes approximated. Current devices in use typically output several hundred lines of characters per minute (with overprinting the number of actual lines comes down). Each page of output may be likened to a single frame of film hence enabling a sequence of pages to tell a story as is the case with a sequence of frames on film. The primary advantage to this medium of output is availability and cost. Also considerable software support programming has been done for such devices.

Typical pen and ink plotters allow a pen to be moved over paper and if pressed to the paper a line results from the movement (called a draw), while if the pen is not pressed to the paper no line results on movement, (called a move). Thus, again, plots of geometric shapes may be generated. Likewise, several pages of plotter output may be put together to tell a story as indicated earlier. The principal advantage of such devices is accuracy of detail and large output format. Both the printers and the plotters described take time in the order of seconds to minutes to produce each plot.

The CRT devices have a beam of electrons which may be deflected at variable speed across the phosphor coating on a screen causing the phosphor to glow for a period of time again producing a picture or frame of output. A sequence of such pictures can again be used to tell a story. Once the phosphor is excited, if it continues to emit visable light for a period of minutes the scope employs what is known as a storage tube which holds an image once drawn through use of a high persistance phosphor. If the image fades within a matter of microseconds then the tube utilized is a refresh tube and it employs a low persistance phosphor. Systems with storage tubes allow for viewing single frames in a sequence one at a time with a time delay between frames on the order of seconds to minutes depending on the system employed. Such systems may be used to show static motion as defined earlier. Systems with refresh tubes typically employ more sophisticated hardware to support the rapid updating of information (in the form of pictures and text). Such systems are relatively expensive; however, due to the rapidity with which each picture can be drawn (typically 1/40 second or less) and due to the interaction possible, dynamic motion may be employed and interactive techniques developed to allow the user to communicate with a program in real-time. For a discussion of Computer Graphics see Newman and Sproull (1).

COMPUTER SYSTEM

The line printer on which these pictures were developed is an IBM 1403 printer. The plotter on which these pictures were plotted is a Calcomp 564-30 inch incremental plotter. The interactive graphics system on which these programs were developed is an Adage Model 30 with 32768-30 bit words of core. (2) Control devices used include a joystick, 6 variable control dials, alphanumeric keyboard, 18 function switches and foot-pedals. With this equipment the images are dynamically manipulable by both the instructor and the student.

The programs and techniques discussed in this paper have been tested, implemented and are currently running as part of a computer graphics training program on the Pennsylvania State University Computation Center Adage Model 30. They form part of a computer graphics on-line text book base for the development of computer graphics geometric conceptualization as required by the Computer Science undergraduate course in graphics presented by Penn State.

*Incremental motion where the increments taken show large changes in information or geometric orientation.

They have been successfully used to supplement 9 lectures (one per week) per ten week term (Spring and Fall, 1973 and Spring, 1974). The information was presented directly via the graphics system using the scope as a blackboard and the instructor as the knowledgeable programmer. This course is open to students from most programs in the University.

It should be emphasized that with recent advances in computer hardware, and industry projection of further developments the availability of interactive graphics units is rapidly on the rise. Storage tubes can now be purchased for under $4,000.00. Some mini-computers for programming support are in the one to ten thousand dollar range. Refresh scopes are also becoming inexpensive and raster systems (TV like systems) are being employed for which the costs are within the $10,000 range per unit with mini-computers interfaced for support. These costs will continue to come down as the new miniturization technology, such as used in the new pocket calculators, advances. As this happens, more departmental requests for plotting support in terms of both hardware and software programming will be honored if need can be demonstrated.

PROGRAMS FOR CONCEPT DEMONSTRATION

The computer programs, MATRIX and SQWAV, are utilized in this paper to demonstrate the techniques under discussion. MATRIX demonstrates the geometric significance of matrices and vectors. SQWAV allows sine curve approximations to a square wave to be explored. The MATRIX program is intended to aid in developing understanding at the geometric and analytical levels, thus enabling the instructor or student to develop mathematical comprehension. The SQWAV program illustrates a mechanism whereby detailed analysis for comprehension in an application may be developed. The MATRIX and SQWAV programs provide a variety of visual options to choose from thus freeing the instructor to select the concept area desired within the framework of the program being processed. Through the use of such programs, the comprehension of the student is readily tested and the development of new understanding as well as the correction of weak areas can be accomplished by the instructor. In this manner we feel graphics is seen to be an important teaching aid to the classroom instructor and for the student.

In Figure 1 the interactive scope, plotter and line printer are contrasted at the pictorial level to emphasize realizable aspects of the thoughts presented. The other figures illustrate other concepts which could be viewed in a similar manner.

MATRIX PROGRAM

By selecting a transformation to be varied off a variable control dial or through a program input READ statement, such as rotation about the Z axis (in the XY plane), the variation of vector and matrix equation entries may be monitored dynamically as illustrated in Figure 1. It should be noted that the X,Y-planar rotation submatrix varies but that all Z rotation elements remain unchanged. This allows the two dimensional transformation (rotation) to be observed within the larger context of the three-dimensional environment and also indicates the mathematical independence of the third dimension. In a similar manner, other rotations, translations and scaling may be studied as well as how they effect each other in combination.

EFFECTS OF INDIVIDUAL MATRIX ENTRIES

Another important aspect illustrated in the matrix program is the effect on an individual matrix element on a geometric transformation. Figure 2 presents sequences of pictures which illustrate the changing of the rotational matrix element in row 1, column 1 of the rotation array to affect scaling in X. The scaling range is illustrated from zero to full scale. Figure 3 illustrates the individual and combined effects of skewing in X, then Y and then both X and Y. The last picture of this sequence illustrates viewing such a skewing operation in 3-dimensions. Figure 4 indicates a precision coordinate grid available to check on the transformation accuracy from a visual standpoint, while figure 5 indicates how the various mathematical representations are selectable and comparable on the scope face as different transformational aspects are being monitored. The input and control tables may optionally be viewed as illustrated. Note that in all of the scope illustrations, the primary value being monitored is shown at the bottom of the graphic image to three significant figures while the matrix items being monitored are shown only to two figures, thus illustrating the round off in effect on the system.

SQUARE WAVE PROGRAM

The blackboard problem is very pronounced in teaching Fourier Series. With simple sketching the instructor can illustrate how the terms are summed and possibly convince the student that the series will converge. The accuracy and speed of computer graphics, however, allow the instructor to illustrate more concepts convincingly. On a cathode ray tube one can watch the series converging and gain some meaningful insight into the rapidity of convergence. The Gibb's phenomenon which is almost impossible to sketch with any accuracy or to even convince the students that it exists, is beautifully illustrated (note Figure 6). That is one reason for using the split screen to show Gibb's phenomenon along with the next term to be added and the sum of the first terms of the series. The computer output can be used by each individual student or they can be used as visual aids for the instructor. Both techniques have been used by Mathematics Professors in teaching the Course Fourier Series and Partial Differential Equations to establish understanding which cannot be gained from the usual classroom exercises.

CONCLUSION

Through the transformation matrix and square wave examples we have indicated how programs at both the structure and application levels can be used to produce a symbiotic affect between theory and applications where the student develops understanding in both areas simultaneously. Programs such as this are intended to provide the knowledgeable instructor with a source of mathematical and geometrical information which makes available for selection different representations of the same concepts. A collection of such programs provides an interactive textbook-like data base for teaching fundamental concepts in an area, in this case two and three dimensional transformations contrasting mathematical models with both matrix and geometric representations. Sequences of output as shown in the figures can be obtained from the computer with the student supplying the input sequence for control of an entry in the matrices or an axis or axies about which the rotation is to be observed. The sequential output of pictures will then tell the story. Preferred sequences of values may be selected to develop course material of an instructor's own choosing for classroom work. In the case of an interactive system both the knowledgeable instructor and the novice student may dynamically manipulate the geometric imagery and/or the matrix entries to study the effects of modifying either the geometric structure or the analytical mathematical model which corresponds to the geometric image. In this manner the student develops a visual geometric understanding of the analytic model as well as an analytical understanding of the geometric structures and transformations.

Thus computer graphics can be used as a valuable

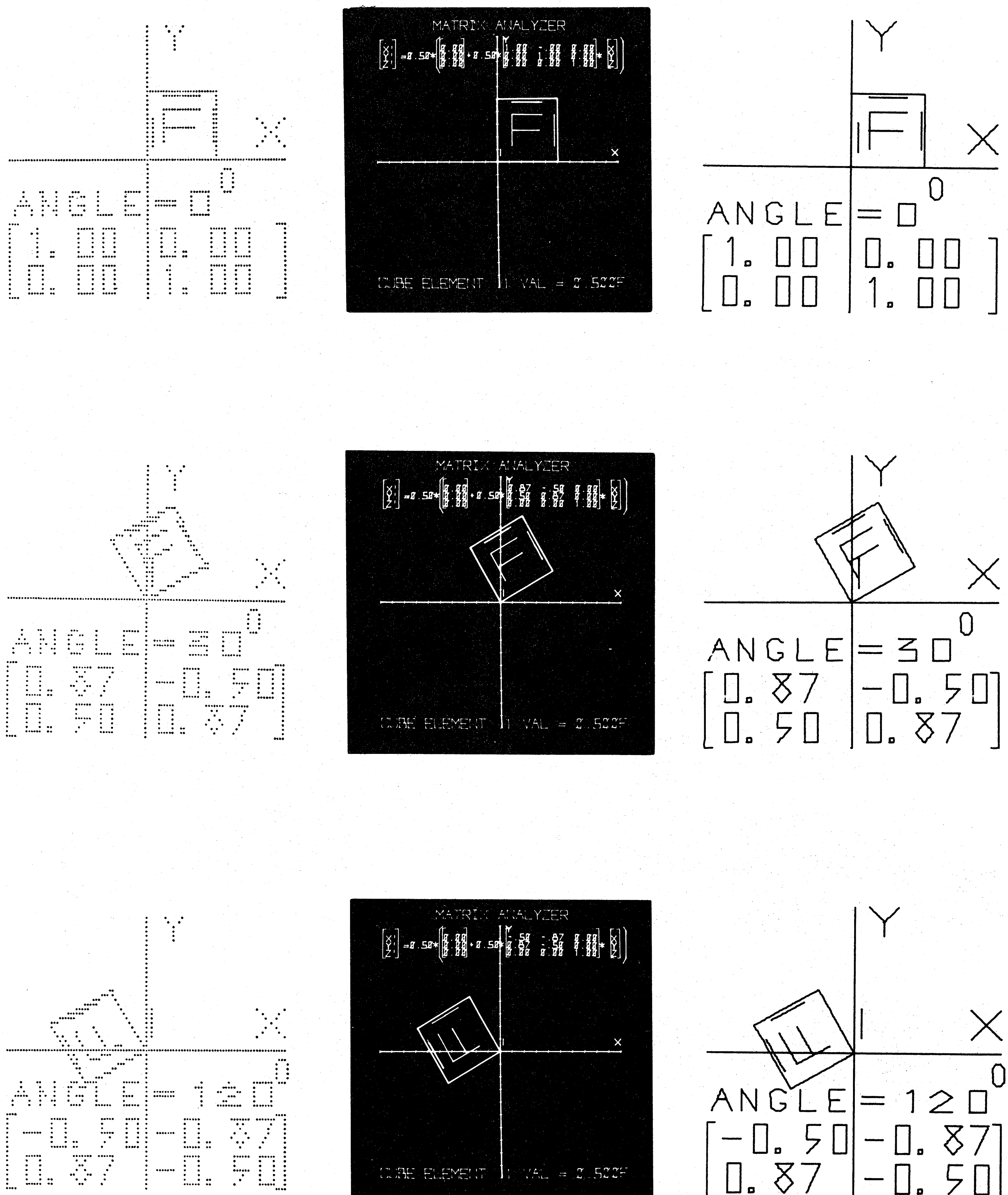

Y
X
ANGLE = 0°
1.00 0.00
0.00 1.00
MATRIX ANALYZER
CUBE ELEMENT 1
ANGLE = 30°
0.87 −0.50
0.50 0.87
ANGLE = 120°
−0.50 −0.87
0.87 −0.50

Figure 1

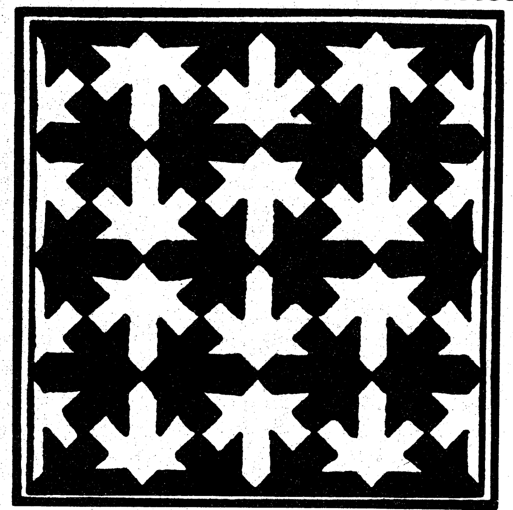
teaching aid at either the classroom level, the individual (1 to 1) confrontation level, or at the level where the student seeks information through direct program interaction. With a little imagination, such techniques as these can be implemented using any available high speed printers or plotters for output.

It should be noted that programs, such as those illustrated herein, make excellent student programming projects. They allow the use of as much programming skill and imagination as the student is capable of, because any project of this type is open ended and can grow from very simple to quite complex. The end results of a project are useful, giving the student a sense of accomplishment in not doing just another exercise.

REFERENCES

(1) William M. Newman and Robert F. Sproull, *Principles of Interactive Computer Graphics*. McGraw Hill, 1973

(2) *Adage Fortran Programming System*, Adage, Inc., 1079 Commonwealth Avenue, Boston, Mass. 1972



Fluctuation of attention Illusion
Gaze steadily and will change from white design on black background to black design on white background.

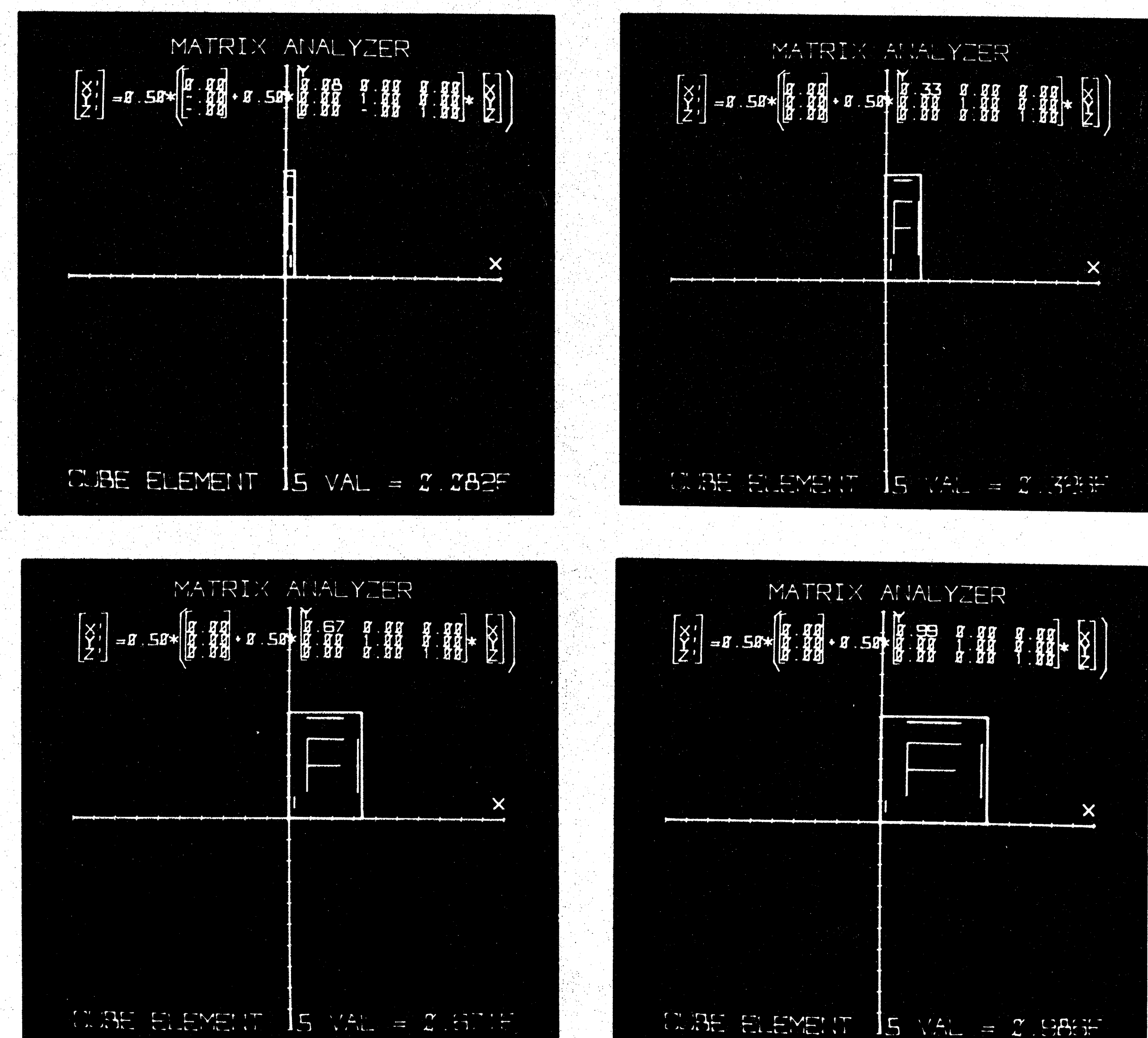


Figure 2

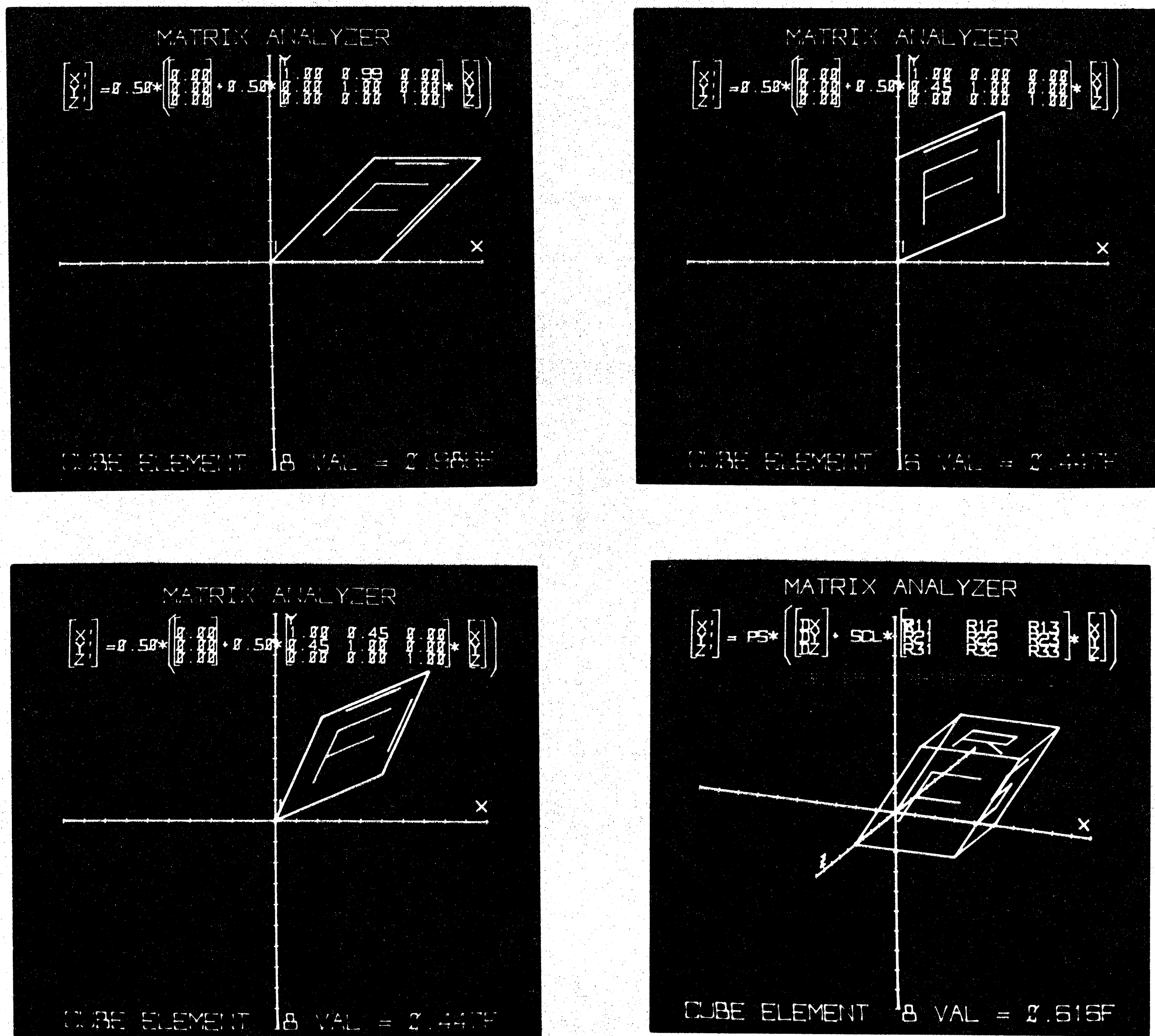


Figure 3

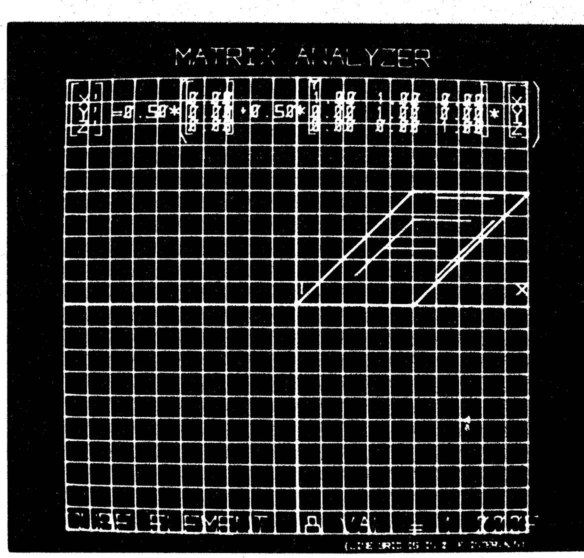


Figure 4

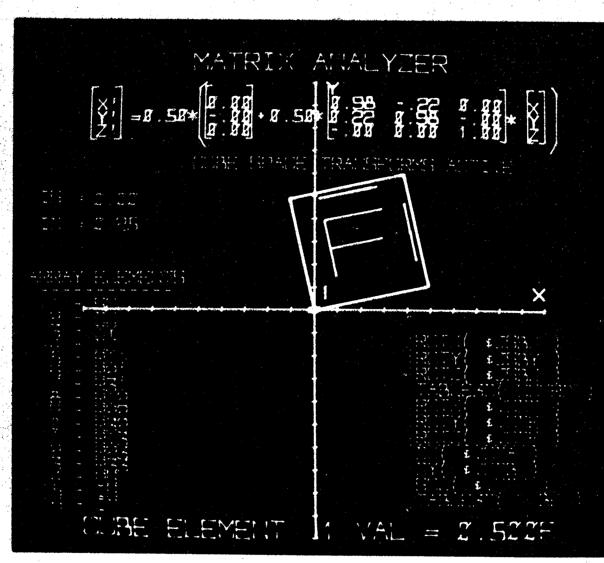


Figure 5

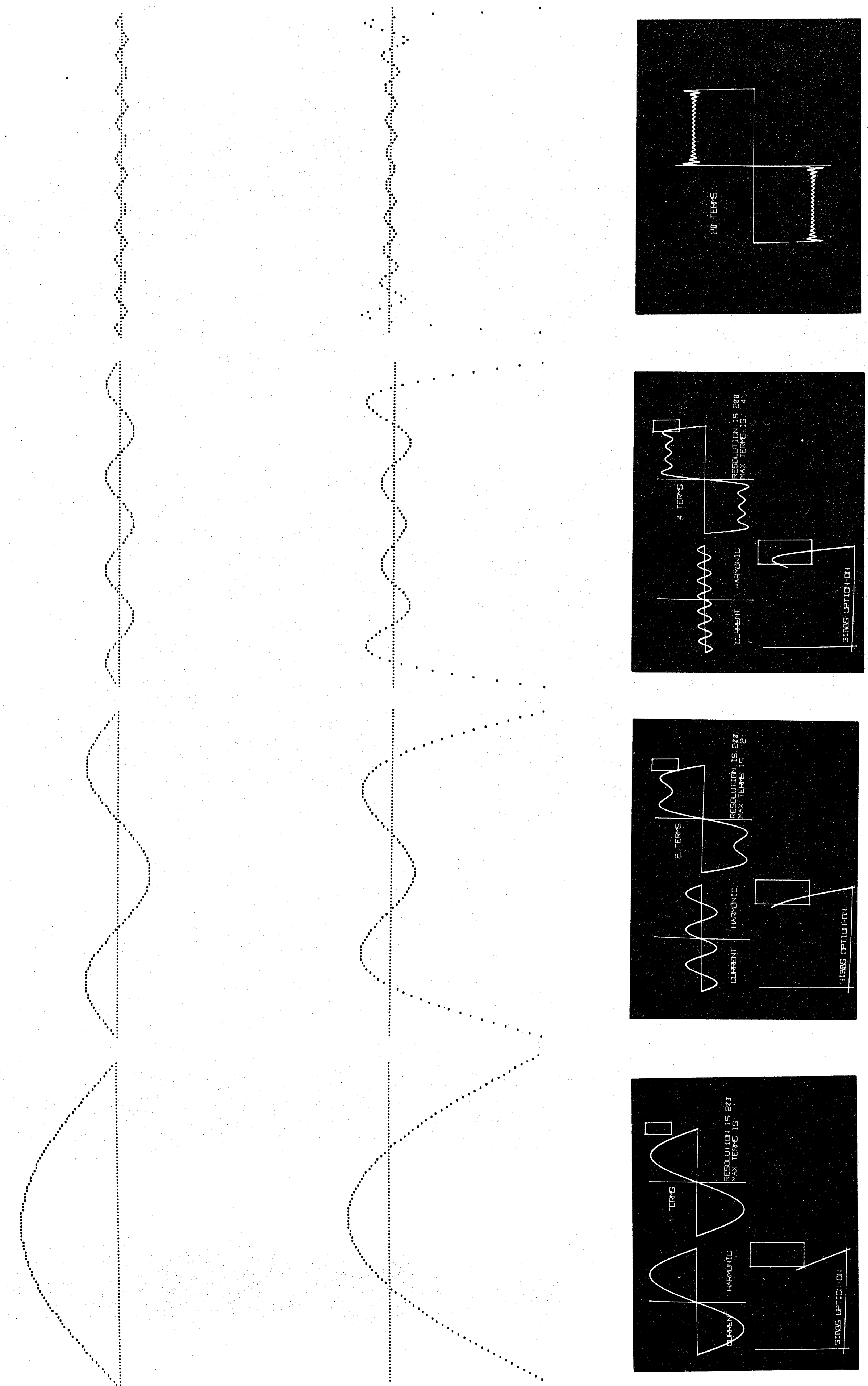

CURRENT
HARMONIC
1 TERMS
RESOLUTION IS 2ØØ
MAX TERMS IS 1
GIBBS OPTION-ON
CURRENT
HARMONIC
2 TERMS
RESOLUTION IS 2ØØ
MAX TERMS IS 2
GIBBS OPTION-ON
CURRENT
HARMONIC
4 TERMS
RESOLUTION IS 2ØØ
MAX TERMS IS 4
GIBBS OPTION-ON
2Ø TERMS

Figure 6

GEOMETRIC PROOFS

by Thomas J. Kelanic
Project SOLO and
Taylor-Allderdice High School

It has bothered me for over a decade that I cannot personally check each of the 300 or more geometric proofs my students submit to me as daily homework. Consider that any high school geometry teacher may have a hundred students to whom he may assign three or more proofs for homework. Can the teacher read and analyze each potentially unique theorem produced? Can the teacher spend enough time on each proof to diagnose and correct the errors he may find? Can each student see what he did wrong when one correct proof of each theorem is placed on the chalkboard? Can the student who has written an "off-beat" but valid proof receive approval? Does the student get enough feedback from his own paper as he writes his proof? Is the student aware of the alternatives open to him at each step in his proof? Does the student feel a sense of direction as he writes his proof?

I believe the program I have written is an important step forward in attempting to provide an affirmative answer to all of the questions above. My first conceptualization of the program was stimulated by an article describing pattern-matching of "relation-chains" using "keywords" in a man-machine dialogue by Rockart, Morton, and Zannetos (1971). The SITBOL improvements by Gimpbel (1972) of the SNOBOL4 programming language of Griswold, Poage, and Polonsky (1971) seemed to me to provide the best medium for expression of the concept.

The program provides "dynamic" proof construction capability which may enable a student to write three or four proofs in a 45-minute period. The run shown, for example, took 12 minutes of terminal time. By "dynamic," I mean that the program will perform the following functions:

1. Examine student input for proper vocabulary and word sequence. Faulty sentences are ignored; a diagnostic message is issued; and another opportunity for input is presented.
2. Examine student input statements for logical validity; that is, verify that the statement is a given assumption, a valid conclusion from previous statements, or a postulate or property which is accepted as true independently. Invalid statements are identified and rejected, and another opportunity for input is presented.
3. Validate reasons given by the student for each statement he inputs. If the reason is incorrect, the correct reason is output and reference is made to the previous statements which support the conclusion.
4. Output all valid conclusions which are implied by the new statement in combination with any of the previous statements. The student may then elect to input one of these conclusions as a new statement.
5. Provide for student control through the use of student commands such as the following:
   a. "REVIEW" to review all previous statements.
   b. "TOTAL" to review all previous conclusions.
   c. Backslash (\) to ignore faulty input.
   d. "PROVE" to exit the "givens" loop.
   e. "QUIT" to terminate program execution.
6. Output the proof in standard two-column format and give a tally of invalid statements and erroneous reasons, if any.

Although the program is incomplete, it works. The basic interactive concept of the program has been firmly established. The program functions "dynamically" and adds a kinetic element to the study of implications. The student can witness an implication as an event. The student can exercise his control over these events, and can pursue, in partnership with the program, many alternative routes during proof construction. The use of statement number references in reasons can leave little doubt in the student's mind as to which statements were the cause of the conclusions produced and why. The student will find it desirable to learn the postulates when he realizes that they are simply unstated "givens." The program presently uses 10 implications which represent a few definitions, theorems, and properties selected not on the basis of geometric precedence, but on the basis of programming challenge. In addition to these 10, one property that does not behave as an implication, the reflexive property of congruence, is used. Before testing the program on students, further development of the program will be continued.

EXAMPLE SHOWING USE OF THE PROGRAM

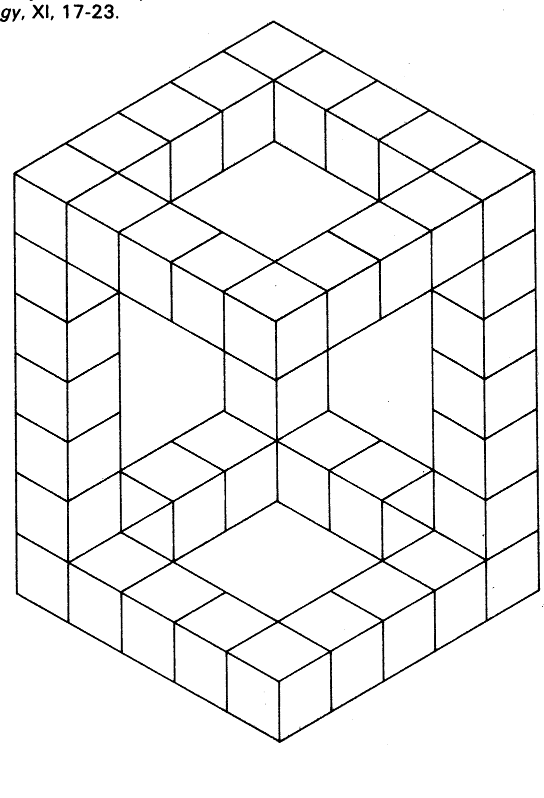
In the sample run which follows, all student input has been marked by a heavy arrow (←) to aid interpretation by the reader. The program itself, which was written in SNOBOL4, is not shown since not many schools have this language. (A future project will be to try to write the program in BASIC-PLUS).

REFERENCES

Gimpbel, J.F. 1972. *SITBOL* Version 1.0, Document S4D30, Bell Telephone Laboratories Inc., Holmdel, New Jersey.

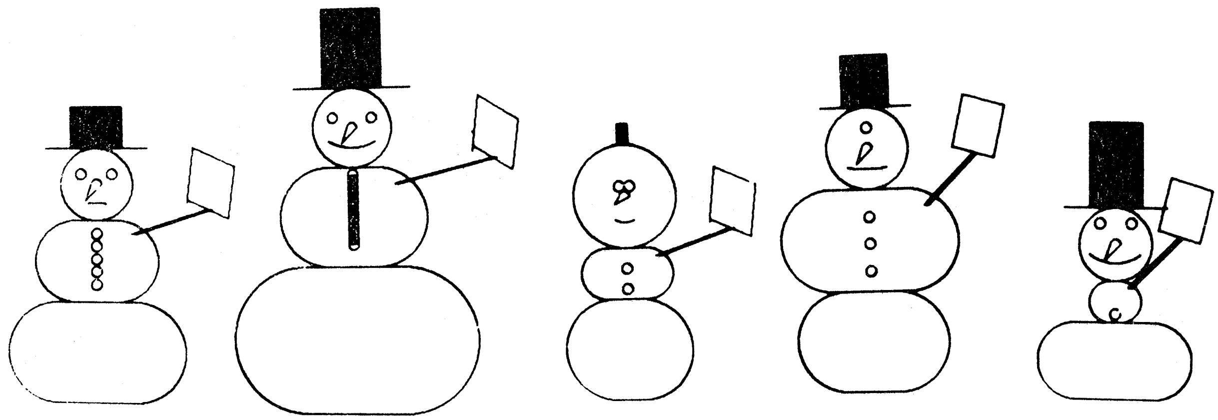
Griswold, R.E., J.F. Poage, and I.P. Polonsky 1971. *The SNOBOL4 Programming Language*, Second Edition, Prentice Hall, Englewood Cliffs, New Jersey.

Rockart, J.F., M.S. Scott Morton, and Z.S. Zannetos, "Associative Learning Project in Computer-Assisted Instruction," 1971 *Educational Technology*, XI, 17-23.



SAMPLE RUN

```
.EXECUTE PROOFS.SNO
---WHAT IS 'GIVEN'?
M-SEGMENT AC = M-SEGMENT AD.
---WHAT ELSE IS 'GIVEN'?
SEGMENT BC CONGRUENT SEGMENTBX\
---WHAT ELSE IS 'GIVEN'?
SEGMENT BC CONGRUENT SEGMENT BX.
---WHAT ELSE IS 'GIVEN'?
SEGMENT BD CONGRUENT SEGMENT BX.
---WHAT ELSE IS 'GIVEN'?
TRIANGLE ABC CORRESPONDS TRIANGLE ABD.
---WHAT ELSE IS 'GIVEN'?
PROVE
---WHAT IS 'TO BE PROVED'?
M-ANGLE ACB = M-ANGLE ADB.

-----------------------------------------------------------------
GIVEN: 0. M-SEGMENT AC = M-SEGMENT AD.
       0. SEGMENT BC CONGRUENT SEGMENT BX.
       0. SEGMENT BD CONGRUENT SEGMENT BX.
       0. TRIANGLE ABC CORRESPONDS TRIANGLE ABD.
PROVE: 0. M-ANGLE ACB = M-ANGLE ADB.
-----------------------------------------------------------------
STATEMENTS:                        REASONS:
-----------------------------------------------------------------

---WHAT IS YOUR STATEMENT NUMBER 1 ?
SEGMENT AC CONGRUENT SEGMENT AD.
---WHAT IS YOUR REASON NUMBER 1 ?
GIVEN.
---INVALID STATEMENT.
---WHAT IS YOUR STATEMENT NUMBER 1 ?
M-SEGMENT AC = M-SEGMENT AD.
---WHAT IS YOUR REASON NUMBER 1 ?
GIVEN.

1. M-SEGMENT AC = M-SEGMENT AD.    ! 1. GIVEN (VALID).

---NEW CONCLUSIONS FROM THE ABOVE ARE:
SEGMENT AC CONGRUENT SEGMENT AD.
---WHAT IS YOUR STATEMENT NUMBER 2 ?

---NULL STRING WAS INPUT AND IGNORED.
---MAY BE DUE TO SENSITIVE KEYBOARD AT TERMINAL.
---WHAT IS YOUR STATEMENT NUMBER 2 ?
SEGMENT AC CONGRUENT SEGMENT AD.
---WHAT IS YOUR REASON NUMBER 2 ?
GIVEN.

2. SEGMENT AC CONGRUENT SEGMENT    ! 2. DEFINITION CONGRUENCE AND
   AD.                             !    STATEMENT 1. NOT 'GIVEN'
                                   !    (STUDENT ERROR).

---NEW CONCLUSIONS FROM THE ABOVE ARE:
SEGMENT AD CONGRUENT SEGMENT AC.
M-SEGMENT AC = M-SEGMENT AD.
---WHAT IS YOUR STATEMENT NUMBER 3 ?
SEGMENT BC CONGRUENT SEGMENT BX.
---WHAT IS YOUR REASON NUMBER 3 ?
GIVEN.

3. SEGMENT BC CONGRUENT SEGMENT    ! 3. GIVEN (VALID).
   BX.                             !

---NEW CONCLUSIONS FROM THE ABOVE ARE:
SEGMENT BX CONGRUENT SEGMENT BC.
M-SEGMENT BC = M-SEGMENT BX.
---WHAT IS YOUR STATEMENT NUMBER 4 ?
SEGMENT BD CONGRUENT SEGMENT BX.
---WHAT IS YOUR REASON NUMBER 4 ?
GIVEN.

4. SEGMENT BD CONGRUENT SEGMENT    ! 4. GIVEN (VALID).
   BX.                             !

---NEW CONCLUSIONS FROM THE ABOVE ARE:
SEGMENT BX CONGRUENT SEGMENT BD.
M-SEGMENT BD = M-SEGMENT BX.---WHAT IS YOUR STATEMENT NUMBER 5 ?

---NULL STRING WAS INPUT AND IGNORED.
---MAY BE DUE TO SENSITIVE KEYBOARD AT TERMINAL.
---WHAT IS YOUR STATEMENT NUMBER 5 ?
SEGMENT BX CONGRUENT SEGMENT BD.
---WHAT IS YOUR REASON NUMBER 5 ?
DON'T KNOW

5. SEGMENT BX CONGRUENT SEGMENT    ! 5. SYMMETRIC PROPERTY CONGRUENCE
   BD.                             !    AND STATEMENT 4. IS VALID
                                   !    (STUDENT ERROR).

---NEW CONCLUSIONS FROM THE ABOVE ARE:
SEGMENT BD CONGRUENT SEGMENT BX.
SEGMENT BC CONGRUENT SEGMENT BD.
SEGMENT BD CONGRUENT SEGMENT BD.
M-SEGMENT BX = M-SEGMENT BD.
---WHAT IS YOUR STATEMENT NUMBER 6 ?
SEGMENT B\
---WHAT IS YOUR STATEMENT NUMBER 6 ?
TOTAL
SEGMENT AC CONGRUENT SEGMENT AD.
SEGMENT AD CONGRUENT SEGMENT AC.
M-SEGMENT AC = M-SEGMENT AD.
SEGMENT BX CONGRUENT SEGMENT BC.
M-SEGMENT BC = M-SEGMENT BX.
SEGMENT BX CONGRUENT SEGMENT BD.
M-SEGMENT BD = M-SEGMENT BX.
SEGMENT BD CONGRUENT SEGMENT BX.
SEGMENT BC CONGRUENT SEGMENT BD.
SEGMENT BD CONGRUENT SEGMENT BD.
M-SEGMENT BX = M-SEGMENT BD.
---WHAT IS YOUR STATEMENT NUMBER 6 ?
REVIEW
1. M-SEGMENT AC = M-SEGMENT AD.
2. SEGMENT AC CONGRUENT SEGMENT AD.
3. SEGMENT BC CONGRUENT SEGMENT BX.
4. SEGMENT BD CONGRUENT SEGMENT BX.
5. SEGMENT BX CONGRUENT SEGMENT BD.
---WHAT IS YOUR STATEMENT NUMBER 6 ?
SEGMENT BC CONGRUENT SEGMENT BD.
---WHAT IS YOUR REASON NUMBER 6 ?
DON'T KNOW

6. SEGMENT BC CONGRUENT SEGMENT    ! 6. TRANSITIVE PROPERTY CONGRUENCE
   BD.                             !    AND STATEMENTS 3,5. IS VALID
                                   !    (STUDENT ERROR).

---NEW CONCLUSIONS FROM THE ABOVE ARE:
SEGMENT BD CONGRUENT SEGMENT BC.
M-SEGMENT BC = M-SEGMENT BD.
---WHAT IS YOUR STATEMENT NUMBER 7 ?
TRIANGLE ABC CORRESPONDS TRIANGLE ABD.
---WHAT IS YOUR REASON NUMBER 7 ?
GIVEN.

7. TRIANGLE ABC CORRESPONDS        ! 7. GIVEN (VALID).
   TRIANGLE ABD.                   !

---NEW CONCLUSIONS FROM THE ABOVE ARE:
---NONE.
---WHAT IS YOUR STATEMENT NUMBER 8 ?
TRIANGLE ABC CONGRUENT TRIANGLE ABD.
---WHAT IS YOUR REASON NUMBER 8 ?
GIVEN.
---INVALID STATEMENT.
---WHAT IS YOUR STATEMENT NUMBER 8 ?
SEGMENT AB CONGRUENT SEGMENT AB.
---WHAT IS YOUR REASON NUMBER 8 ?
REFLEXIVE PROPERTY.

8. SEGMENT AB CONGRUENT SEGMENT    ! 8. REFLEXIVE PROPERTY CONGRUENCE
   AB.                             !    AND STATEMENT 8. IS VALID
                                   !    (STUDENT ERROR).

---NEW CONCLUSIONS FROM THE ABOVE ARE:
TRIANGLE ABC CONGRUENT TRIANGLE ABD.
M-SEGMENT AB = M-SEGMENT AB.
---WHAT IS YOUR STATEMENT NUMBER 9 ?
TRIANGLE ABC CONGRUENT TRIANGLE ABD.
---WHAT IS YOUR REASON NUMBER 9 ?
DON'T KNOW.

9. TRIANGLE ABC CONGRUENT          ! 9. SIDE SIDE SIDE AND STATEMENTS
   TRIANGLE ABD.                   !    2,6,7,8. IS VALID (STUDENT
                                   !    ERROR).

---NEW CONCLUSIONS FROM THE ABOVE ARE:
TRIANGLE ABD CONGRUENT TRIANGLE ABC.
M-TRIANGLE ABC = M-TRIANGLE ABD.
SEGMENT AB CONGRUENT SEGMENT AB.
SEGMENT BC CONGRUENT SEGMENT BD.
SEGMENT AC CONGRUENT SEGMENT AD.
ANGLE ABC CONGRUENT ANGLE ABD.
ANGLE BAC CONGRUENT ANGLE BAD.
ANGLE ACB CONGRUENT ANGLE ADB.
---WHAT IS YOUR STATEMENT NUMBER 10 ?
ANGLE ACB CONGRUENT S\
---WHAT IS YOUR STATEMENT NUMBER 10 ?
ANGLE ACB CONGRUENT ANGLE ADB.
---WHAT IS YOUR REASON NUMBER 10 ?
DEFINITION CONGRUENT TRIANGLE\S AND STATEMENT 9.

10. ANGLE ACB CONGRUENT ANGLE ADB. ! 10. DEFINITION CONGRUENT
                                   !     TRIANGLES AND STATEMENT 9. IS
                                   !     VALID (STUDENT ERROR).

---NEW CONCLUSIONS FROM THE ABOVE ARE:
ANGLE ADB CONGRUENT ANGLE ACB.
M-ANGLE ACB = M-ANGLE ADB.
---WHAT IS YOUR STATEMENT NUMBER 11 ?
M-ANGLE ACB = M-ANGLE ADB.
---WHAT IS YOUR REASON NUMBER 11 ?
DEFINITION CONGRUENCE AND STATEMENT 10.

-----------------------------------------------------------------
GIVEN: 0. M-SEGMENT AC = M-SEGMENT AD.
       0. SEGMENT BC CONGRUENT SEGMENT BX.
       0. SEGMENT BD CONGRUENT SEGMENT BX.
       0. TRIANGLE ABC CORRESPONDS TRIANGLE ABD.
PROVE: 0. M-ANGLE ACB = M-ANGLE ADB.
-----------------------------------------------------------------
STATEMENTS:                          REASONS:
-----------------------------------------------------------------
1. M-SEGMENT AC = M-SEGMENT AD.    ! 1. GIVEN (VALID).
2. SEGMENT AC CONGRUENT SEGMENT    ! 2. DEFINITION CONGRUENCE AND
   AD.                             !    STATEMENT 1. NOT 'GIVEN'
                                   !    (STUDENT ERROR).
3. SEGMENT BC CONGRUENT SEGMENT    ! 3. GIVEN (VALID).
   BX.                             !
4. SEGMENT BD CONGRUENT SEGMENT    ! 4. GIVEN (VALID).
   BX.                             !
5. SEGMENT BX CONGRUENT SEGMENT    ! 5. SYMMETRIC PROPERTY CONGRUENCE
   BD.                             !    AND STATEMENT 4. IS VALID
                                   !    (STUDENT ERROR).
6. SEGMENT BC CONGRUENT SEGMENT    ! 6. TRANSITIVE PROPERTY CONGRUENCE
   BD.                             !    AND STATEMENTS 3,5. IS VALID
                                   !    (STUDENT ERROR).
7. TRIANGLE ABC CORRESPONDS        ! 7. GIVEN (VALID).
   TRIANGLE ABD.                   !
8. SEGMENT AB CONGRUENT SEGMENT    ! 8. REFLEXIVE PROPERTY CONGRUENCE
   AB.                             !    AND STATEMENT 8. IS VALID
                                   !    (STUDENT ERROR).
9. TRIANGLE ABC CONGRUENT          ! 9. SIDE SIDE SIDE AND STATEMENTS
   TRIANGLE ABD.                   !    2,6,7,8. IS VALID (STUDENT
                                   !    ERROR).
10. ANGLE ACB CONGRUENT ANGLE ADB. ! 10. DEFINITION CONGRUENT
                                   !     TRIANGLES AND STATEMENT 9. IS
                                   !     VALID (STUDENT ERROR).
11. M-ANGLE ACB = M-ANGLE ADB.     ! 11. DEFINITION CONGRUENCE AND
                                   !     STATEMENT 10. (VALID).
-----------------------------------------------------------------
2 INVALID STATEMENTS, 6 ERRONEOUS REASONS. ---Q.E.D.
-----------------------------------------------------------------

EXIT
```



Computer-Planned Snowmen

Robert S. McLean
Ontario Institute for Studies in Education
Toronto, Canada

Computer models are useful for testing theories. If you have a model of a process or structure expressed as a computer program, you have a powerful tool. A computer can then produce many different instances of the modeled process through the use of a range of parameter values. It will produce the results of the model without being influenced by extraneous notions of what the outcome should be. In addition to investigating the range of applicable parameter values, one can even push those values beyond "reasonable" limits and observe the results. The computer tirelessly shows the results of the chosen conditions without requiring very many input resources. If the result of the model is a picture, so much the better; its adequacy can often be judged visually by the user very rapidly. What is more, any pictorial outputs can be very entertaining when they illustrate weaknesses of the model or data, thereby producing very unusual pictures.

Computer modeling at first seems like such a high-powered idea that it would be hard to apply for fun. Maybe it ought to be reserved for sending men to the moon, finding petroleum resources, or managing large construction projects. Although these projects use modeling, simple things that an individual does can also benefit from these techniques. Consider building a snowman, for instance. Here is an important problem for the individual which can be solved with the power of modern computing.

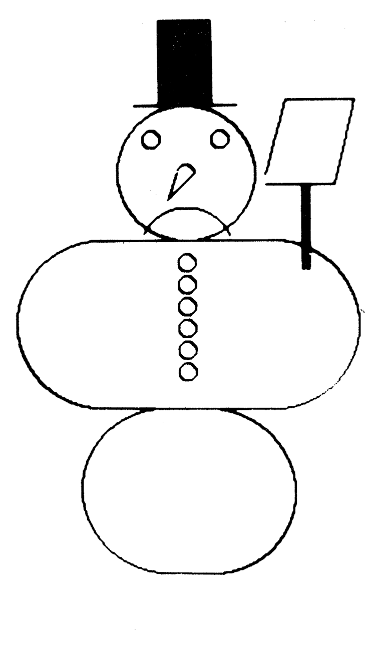
By choosing the proper aspects of a snowman as parameters for our model, we can use the computer to draw pictures of the resulting snowmen. We can observe the shape of hundreds of combinations of parameter values and select the most pleasing one for implementation with real materials. In addition to increasing the range of choice available, this procedure has several other advantages. Most of the design process as well as some of the construction effort is no longer at the mercy of the weather. Much of the work can be done in locations not previously suitable for this activity (home, office, school room, etc. were not suitable places for snowman construction activity previously). Valuable resources are not squandered in making real prototypes since one will have a better idea of the outcome of the process before starting to use these materials. Thus, computer-planned snowmen become feasible in areas where snow is in short supply. One can appreciate that the benefits are many.

What are the parameters of use in modeling a snowman? The reader may wish to propose his own set; for purposes of illustration, we provide a suggested set that were used to design the accompanying illustration. Our standard snowman will consist of three balls of snow, referred to as B1 (on bottom), B2 (middle), and head. In addition to the features of these components, we will add a hat and a broom. This is the economy version, since it could be given many more accessories, such as scarves, cigars, glasses, etc. These are left as an exercise for the reader.

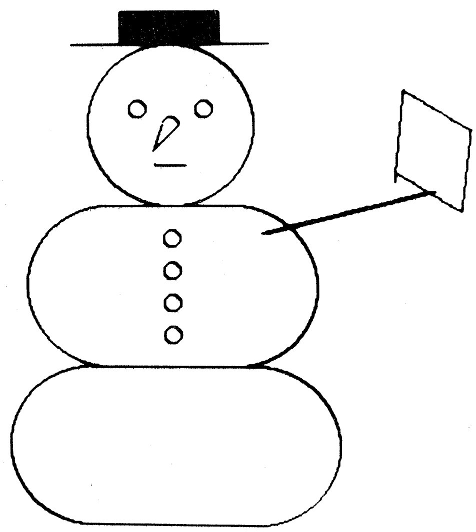
Snowballs 1 and 2 are not necessarily round; they often become somewhat flattened by the load above. Thus, two parameters specify these balls: the radius of the two half circles that form the curved part and the length of the flat parts on top and bottom of the ball, RAD and SIZ respectively, giving the first four parameters, B1SIZ, B1RAD, B2SIZ, and B2RAD. The middle snowball has, in addition, some number of buttons down its front (NBUT). These are spaced out over the vertical extent of B2.

In the world of economy snowmen, heads are always round; hence, there is one head size parameter, HEAD, the radius of the head. Five other parameters are used to place the eyes, nose and mouth on the head. In this version of the model, the nose is always placed in the center of the head and always has the peculiar (carrot-like) shape shown. The sole parameter available here is the length of the nose, NOSE. The eyes are located symmetrically above the center

B1SIZ	50
B1RAD	100
B2SIZ	200
B2RAD	100
N BUT	6
HEAD	80
NOSE	30
SMILE	50
SMIL2	55
EYEHT	40
EYEWD	40
HATWD	60
HATHT	50
BROOM	100
BDISP	0



```
B1SIZ 200
B1RAD 100
B2SIZ 150
B2RAD 100
N BUT   4
HEAD  100
NOSE   30
SMILE  20
SMIL2 200
EYEHT  20
EYEWD  40
HATWD 120
HATHT  20
BROOM  50
BDISP 200
```



of the head along a horizontal line EYEHT units above the center of the head. They are separated by EYEWD units, the interoccular distance.

The mouth is specified by two parameters, where SMILE gives the length of an arc used to denote mouth, and SMIL2 gives the radius of curvature of that arc. Since models should strive for generality in their parameters, SMIL2 may be negative as well as positive. If positive, the center of the arc is above the mouth, and if negative, it is below the mouth. Thus, we obtain the relationship that a frown is a negative smile! The mouth is always placed halfway between the nose and the chin, again in aid of simplicity.

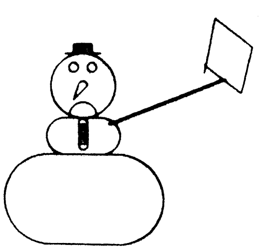
The remaining four parameters specify the two accessories, the hat and the broom. HATWD and HATHT specify the width and height of the hat; the brim is always twice as wide as the hat. The broom is perhaps the most difficult part of the model and as can be seen looks more like a shovel (a kind interpretation) or a strike sign (less kind). It surely leaves room for the reader to improve the model. The parameters used here are BROOM, giving the length of the handle, and BDISP, giving the distance that the broom handle upper end is displaced to the right from the lower end. Some very crude modeling results in the funny quadrilateral that looks more like a hoe in some cases.

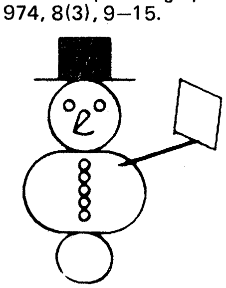
The reader can, no doubt, suggest refinements and is encouraged to do so. The inspiration for this program came from Chernoff's faces (1971) and an adaptation of them that is used to adjust the parameters of oral surgery and to illustrate the various ways in which facial features might be rebuilt (Eisenfeld, et al, 1974). But that's pretty serious business. For the amateur simulator, there are many other possibilities.

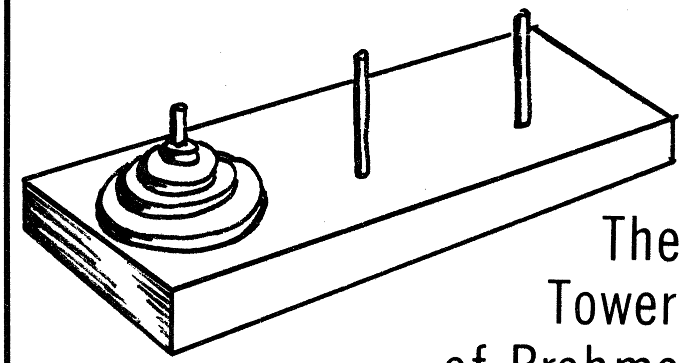
REFERENCES

Chernoff, H. The use of faces to represent points in n-dimensional space graphically. Tech. Report No. 71. Department of Statistics, Standford University, Stanford, California, December 1971.

Eisenfeld, J., Barker, D. R., and Mishelevich, D. J. Iconic representation of the human face with computer graphics. *Computer Graphics* (SIGGRAPH—ACM) 1974, 8(3), 9–15.







The Tower of Brahma

In the great temple at Benares beneath the dome which marks the center of the world, rests a brass plate in which are fixed three diamond needles, each a cubit high and as thick as the body of a bee. On one of these needles, at the creation, God placed sixty-four discs of pure gold, the largest disc resting on the brass plate and the others getting smaller and smaller up to the top one. This is the Tower of Brahma. Day and night unceasingly, the priests transfer the discs from one needle to another, according to the fixed and immutable laws of Brahma, which require that the priest on duty must not move more than one disc at a time and that he must place this disc on a needle so that there is no smaller disc below it. When the sixty-four discs shall have been thus transferred from the needle on which, at the creation, God placed them, to one of the other needles, tower, temple, and Brahmans alike will crumble into dust, and with a thunderclap, the world will vanish.

If the priests were to effect one transfer every second, and work twenty-four hours a day for each day of the year, it would take them 58,454,204,609 decades plus slightly more than 6 years to perform the feat, assuming they never made a mistake—for one small slip would undo all their work.

How many transfers are required to fulfill the prophecy?

A. Set up a program which allows the user to move disks by hand. You can try your ingenuity at drawing the result by some sort of plot or graph.

B. At least verbally, indicate how one would proceed in any arbitrary case (5 disks, 6 disks, etc. 64 is too much to try!).

C. The monks, like monks everywhere, never eat, sleep, rest or die. If they have been moving one disk per second since the world began, how long will the total age of the universe be on Thunderclap Day?

Note: Prove that a game of N disks can be played in 2 -1 (2-to-the-N, minus 1) moves.

D. Analyze the problem in this way: to move 5 disks, what kind of 4-disk moves are required? How do the "from" and "to" of these moves relate to the "from" and "to" of the 5-disk level?

DISCS	MOVES	DISCS	MOVES
1	1	6	63
2	3	7	127
3	7	8	255
4	15	9	511
5	31	10	1023

Roses are red, Computers are blue

by David H. Ahl

There must be some things a computer can't do, but — aside from pole vaulting — it's getting harder and harder to name them. And now the electronic monster is also a poet! Maybe a good one, even. What is poetry, anyway, or some kinds of poetry at least, but words strung together on a page to a musical, surprising, emotional, or otherwise interesting effect? And this a computer can do like crazy. All it requires is a programmer to program its grammar and vocabulary and a man of taste to winnow its rhapsodic outpourings. And though the computer may not yet be the greatest of poets, it is certainly among the freshest and most original. Would you believe an anthology of the future featuring the works of IBM 360? Click.

"Only once in every generation is there a computer that can write poetry like this."

© DATAMATION ®

EXERCISE 1

Try to discover poetry in discarded magazines. Cut out interesting words and phrases and arrange them for meaning or to convey an impression or feeling. Then paste them on construction paper for an interesting display. A typical result:

Now! Run to Mexico
Now you're free
after 30 days of driving
on a single tank of gas
Free!

EXERCISE 2

Run the computer program POET for a page or two of output. This is heavily random except that phrases from four groups generally follow in order. The phrases, for the most part, are similar to those that might be used by Edgar Allen Poe. Some of the stanzas will seem to make sense. Overall the impression is one of evil darkness and impending doom.

EXERCISE 3

Divide the class into small groups. Have each group select one of the groups of words below and use them in the computer program BARD. This program is the same as POET except it allows you to input your phrases instead of using those already in the computer. (You can also use POET by retyping it.)

AIRLINE PHRASES

Giants of the skies
Soaring high
On the world's rim
Breathtaking
The wings of man

Distant lands
New vistas
Luring me
Up, up and away
Fasten your seatbelt

Shades of blue
Silver bird
Time shortens and melts
Entrancing me
Billowing clouds

Endless horizon
Deep red sunset
See the world
Like a bird
Fly high . . . .

PROGRAM POET

The programs are both written in BASIC-PLUS. You'll have to convert them to BASIC for your computer. A backslash (\) separates multiple statements on one line. Generally, you can substitute a backslash for "ELSE" if your BASIC doesn't permit IF-THEN-ELSE. If your BASIC doesn't allow multiple statements on one line, then you have a little more work to do.

```
90 RANDOMIZE
100 IF I<>1 THEN 101 ELSE PRINT "MIDNIGHT DREARY";
101 IF I<>2 THEN 102 ELSE PRINT "FIREY EYES";
102 IF I<>3 THEN 103 ELSE PRINT "BIRD OR FIEND";
103 IF I<>4 THEN 104 ELSE PRINT "THING OF EVIL";
104 IF I<>5 THEN 210 ELSE PRINT "PHOPHET";
105 GOTO 210
110 IF I<>1 THEN 111 ELSE PRINT "BEGUILING ME";
111 IF I<>2 THEN 112 ELSE PRINT "THRILLED ME";
112 IF I<>3 THEN 113 ELSE PRINT "STILL SITTING..."\GOTO 212
113 IF I<>4 THEN 114 ELSE PRINT "BURNED.  "\GOTO 212
114 IF I<>5 THEN 210 ELSE PRINT "NEVER FLITTING";
115 GOTO 210
120 IF I<>1 THEN 121 ELSE IF U=0 THEN 210 ELSE PRINT "SIGN OF PARTING";
121 IF I<>2 THEN 122 ELSE PRINT "AND MY SOUL";
122 IF I<>3 THEN 123 ELSE PRINT "DARKNESS THERE";
123 IF I<>4 THEN 124 ELSE PRINT "SHALL BE LIFTED";
124 IF I<>5 THEN 210 ELSE PRINT "QUOTH THE RAVEN";
125 GOTO 210
130 IF I<>1 THEN 131 ELSE PRINT "NOTHING MORE";
131 IF I<>2 THEN 132 ELSE PRINT "YET AGAIN";
132 IF I<>3 THEN 133 ELSE PRINT "SLOWLY CREEPING";
133 IF I<>4 THEN 134 ELSE PRINT "...NEVERMORE";
134 IF I<>5 THEN 210 ELSE PRINT "EVERMORE.";
210 IF U=0 THEN 212 ELSE IF RND>.19 THEN 212 ELSE PRINT ",";\U=2
212 IF RND>.65 THEN 214 ELSE PRINT " ";\U=U+1\GOTO 215
214 PRINT\U=0
215 I=INT(5*RND+1)
220 J=J+1\K=K+1
230 IF U>0 THEN 240 ELSE IF INT(J/2)<>J/2  THEN 240 ELSE PRINT "     ";
240 ON J GOTO 100,110,120,130,250
250 J=0\PRINT\IF K>20 THEN 270 ELSE GOTO 215
270 PRINT\U=0\K=0\GOTO 110
999 END
```

PROGRAM BARD

```
10 REMNANTS OF ANOTHER PROGRAM, POET, APPEAR IN THIS
20 REMARKABLE PROGRAM ,BARD, WRITTEN BY DAVE AHL.

30 PRINT "BARD WRITES RANDOM POETRY WITH YOUR WORDS OR PHRASES."
40 PRINT "YOU SUPPLY 20 WORDS OR PHRASES UP TO 16 LETTERS LONG."
50 RANDOMIZE \ DIM A$(20) \ FOR I=1 TO 4
60 PRINT \ PRINT "FIVE PHRASES FOR LINE"I"PLEASE"
70 FOR J=1 TO 5 \ S=(I-1)*5+J \ INPUT A$(S)
80 IF LEN(A$(S))>16 THEN INPUT "TOO LONG.  AGAIN...";A$(S) \ GOTO 80
90 NEXT J \ NEXT I
100 PRINT \ INPUT "WHERE DO YOU WANT OUTPUT (KB:, LP:, OR LP1:)";B$
110 OPEN B$ FOR OUTPUT AS FILE 1 \ J=0 \ GOTO 330
200 PRINT #1,A$(I); \ GOTO 300
210 PRINT #1,A$(I+5); \ GOTO 300
220 PRINT #1,A$(I+10); \ GOTO 300
230 PRINT #1;A$(I+15); \ GOTO 300
300 IF U=0 THEN 310 ELSE IF RND>.19 THEN 310 ELSE PRINT #1,","; \ U=2
310 IF RND>.65 THEN 320 ELSE PRINT #1," "; \ U=U+1 \ GOTO 330
320 PRINT #1 \ U=0
330 I=INT(5*RND+1) \ J=J+1 \ K=K+1 \ L=L+1
340 IF U>0 THEN 350 ELSE IF INT(J/2)<>J/2 THEN 350 ELSE PRINT #1,"     ";
350 ON J GOTO 200,210,220,230,360
360 J=0 \ PRINT #1 \ IF K>20 THEN 370 ELSE GOTO 330
370 PRINT #1 \ U=0 \ K=0 \ IF L<200 THEN 200
999 CLOSE 1 \ END
```

This limits a phrase to 16 characters. 20 might be better.

This allows printing on the line printer or your terminal. If you only have a terminal, simply use standard PRINT statements.

NATURE PHRASES

Carpet of ferns
Morning dew
Tang of dawn
Swaying pines
The song of nature

Entrances me
Soothing me
Rustling leaves
Gently caresses
Radiates calm

Mighty oaks
Grace and beauty
Silently singing
Nature speaking
Captures my senses

Untouched, unspoiled
Shades of green
Tranquility
. . . . Evermore
So peaceful . . . .

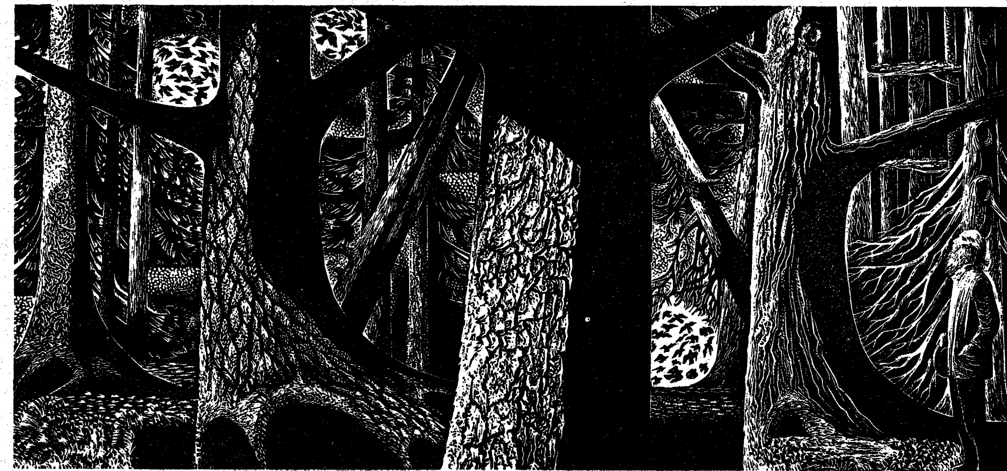
INDUSTRY PHRASES

Harnessing energy
Industrial might
Steel, oil, timber, nylon
Furnaces roaring
Around the clock

Mining, casting, refining
Computer tapes whir
Hammers pounding
Throbbing pulsating
Rubber, zinc, glass

(You write these
5 phrases)

Machines and people
Improving life
Conquering poverty
Energy . . . .
Technology . . . .



3-YEAR-OLD PHRASES

Romper stompers
"Look at me!"
Dancing and singing
Sparkling eyes
Bouncing curls

Devilishly spirited
Unbounded energy
Supremely happy
Inquiring, curious
Looking, discovering

"Tie my shoes"
"I can do it myself"
Skipping and jumping
Radiating joy
Oh, so lively!

Always active
And gumballs
. . . . Delightful
. . . . So lovable
Adventure!

EXERCISE 4

Choose a subject and make up 20 phrases about the subject. Use these phrases in the program BARD.

EXERCISE 5

Read the computer generated poetry below. Do you like it more or less than poetry written by humans? Why?

Since the computer was programmed by a human maybe we should regard the computer as an extension of the hand or the intellect of the writer. In this case the poetry is still really a product of the writer. Do you agree? Why or why not?

ASTRONOMY

TASTE AND TOUCH BRIGHTEN SPRING SNOW
WEEDS BROUGHT THE REFUSAL
UNBEARABLY

WEEDS LOITERED WESTERN STARS
THE EARTH CATWALKS THE EAST RIM
LESS VIVID

TO REST THE EARTH
TO HAVE FOG-BREATH
THE LITTLE WESTERN STARS FOUGHT

A LIONESS

UPON THE SHORE
ALL NIGHT
GROTESQUELY
FRIGID FINAL A LIONESS SAYS

AFTER A RIVER
TOO SOON
TO FEEL
SHINING MEAT-EATING EMOTION GRINNED

THE STEEL

VEINS ROUSE THE STEEL
JOY PROTRACTS THE DIRECTION

MISTS UTTER THE STATUES
RED FIRE WEAVES LOVERS

NAKED PLEASURES BEGIN
THE STEEL

OPTIONAL PROJECT 1

Write a program to compose any kind of poetry you wish.

HAIKU GENERATOR

Paul J. Emmerich
Dana College, Blair, Nebraska

This program is a rather trivial example of the symbiotic relationship of man and computer forecast by John Kemeny as the direction of future development. In it the operator interacts with the computer to produce seventeen syllable, three line statements which, with luck and perseverance, may be termed haiku. You'll no doubt remember that a haiku is a Japanese form of poetry that consists of three lines containing five, seven, and five syllables. The main idea is to present a striking image.

Program Characteristics

The program is composed of four lists of words (nouns, verbs, adjectives, and adverbs), one punctuation mark (the ellipsis), and a method of selecting words at random from these lists to be combined in a particular pattern. The pattern of word categories is determined by the operator at the time he runs the program.

Two inputs are required from the operator — a limit for choosing a number of random numbers (referred to by the program as a "seed") and the pattern of words to be used in composing the haiku. The "seed" is used to run through a series of random numbers produced by the computer's random number generator. The random numbers determine which particular words are chosen from the word lists in the pattern specified by the operator. The word categories are entered as numbers — 1 for noun, 2 for verb, 3 for adjective, 4 for adverb, and 5 for an ellipsis. Up to eighteen categories can be entered for this pattern.

Syllables are counted by identifying *a, e, i, o, u,* and terminal *y*. Terminal *e* does not count as a syllable. Thus *boot* would count as two syllables and *the* would not count at all.

The lists need not be single words since the strings are combined on the basis of length regardless of internal blanks. By following the spelling conventions described above, you can modify the lists to include anything that you desire. If you do not follow the rules (e.g. you use words with adjacent vowel letters that are not pronounced separately), the worst that will happen is that the machine will not count syllables correctly.

Lexical items must be 26 or fewer characters, thus permitting phrases and clauses as well as individual words in the lexicon of the program. Since a haiku contains only seventeen syllables, the number of word categories never needs to be longer than that. In fact, one or two categories can be used alone since the program repeats the input pattern as many times as necessary to come up with at least seventeen syllables. Each haiku is also limited to a maximum of 255 characters.

Currently, there are eighteen of each — noun, verb, adjective, and adverb. From these 72 entries and the one punctuation mark permitted, there are thousands of different combinations possible. Changing the vocabulary, of course, increases the number of different haiku that can be written by the program. If your form of BASIC allows character strings in DATA statements, you can work out a nicer lexicon by using the RND function and reading from DATA statements.

Operation

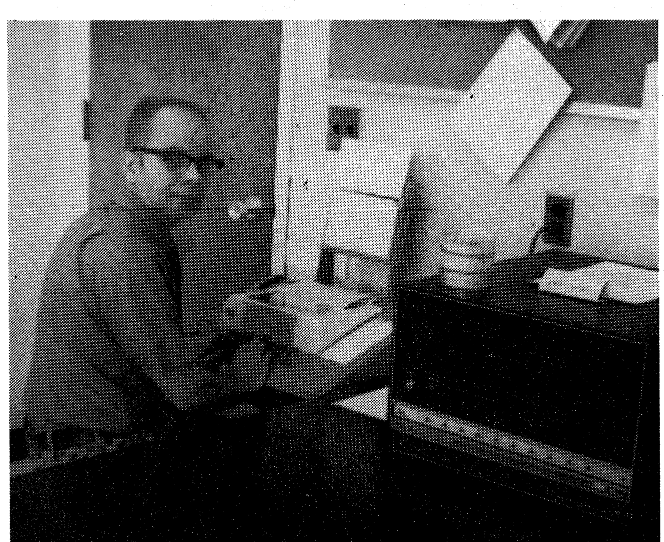
Type the word RUN, followed as usual by RETURN. The program will respond SEED? Enter a number. When the program has produced that many random numbers, it will ask for your list of word categories. Enter numbers from 1 to 5, pressing RETURN after each. The numbers mean:

1 choose a noun
2 choose a verb
3 choose an adjective
4 choose an adverb
5 put in an ellipsis.

At the end of your pattern, enter 0 (zero) to tell the machine to start composing the haiku.

If the first part of a haiku is satisfying, modifications can be made by choosing the same seed, repeating the word category pattern as far as it is satisfactory, and then changing the remaining entries in the pattern.

The program and a sample run appear below.



Author Emmerich and Nova computer produce Haiku together.

"HAIKU" LISTING

```
LIST
0010 REM *****HAIKU GENERATØR
0020 REM *****D.C. PRØGRAM LIBRARY FILE NØ. 850-1
0030 REM *****WRITTEN BY PAUL J. EMMERICH, FALL, 1973
0040 REM *****DANA CØLLEGE, BLAIR, NEBRASKA 68008
0050 DIM C[18],P$[255],A$[26]
0060 PRINT "SEED";
0070 INPUT N
0080 PRINT
0090 REM        *****WIND RANDØM NUMBER GENERATØR
0100 FØR I=1 TØ N
0110    LET X=RND(1)
0120 NEXT I
0130 LET C1=1
0140 REM        *****PATTERN INPUT RØUTINE
0150 PRINT "WØRD CATEGØRIES"
0160 INPUT C[C1]
0170 IF C[C1]=0 GØTØ  0210
0180 IF C[C1]>5 GØTØ  0130
0190 LET C1=C1+1
0200 GØTØ 0160
0210 LET I=0
0220 LET C1=C1-1
0230 PRINT
0240 REM        *****CØNSTRUCT STRING
0250 FØR J=1 TØ C1
0260    LET A=INT(RND(J)*100)+1
0270    IF A=0 GØTØ  0260
0280    IF A>18 GØTØ 0260
0290    ØN C[J] GØSUB  0670, 1100, 1510, 1920, 2330
0300    LET P$=P$,A$
0310    LET I=I+1
0320    IF I=17 GØTØ 0360
0330 NEXT J
0340 LET J=1
0350 GØTØ 0250
0360 LET J=1
0370 REM        *****BLANK REMAINDER ØF LINE FØR PRINTING
0380 LET P$[LEN(P$)+1,250]=" "
0390 LET S2=5
0400 LET L=L+1
0410 REM        *****CØUNT SYLLABLES
0420 FØR I=J TØ LEN(P$)
0430    IF P$[I,I]="A" GØTØ  0500
0440    IF P$[I,I]="E" GØTØ  0490
0450    IF P$[I,I]="I" GØTØ  0500
0460    IF P$[I,I]="Ø" GØTØ  0500
0470    IF P$[I,I]="Y" GØTØ  0520
0480    IF P$[I,I]<>"U" GØTØ 0540
0490    IF P$[I+1,I+1]=" " GØTØ  0540
0500    LET S1=S1+1
0510    GØTØ 0530
0520    IF P$[I+1,I+1]=" " GØTØ  0500
0530    IF S1=S2 GØTØ  0570
0540 NEXT I
0550 REM        *****END ØF WØRD?
0560 LET I=I+1
0570 IF P$[I,I]<>" " GØTØ 0560
0580 REM        *****PRINT LINE
0590 PRINT P$[J,I]
0600 LET J=I
0610 LET S1=0
0620 IF L=2 GØTØ  0390
0630 LET S2=7
0640 IF L=1 GØTØ  0400
0650 PRINT
0660 STØP
0670 REM        *****NØUN RØUTINE
0680 IF A>9 GØTØ  0700
0690 ØN A GØTØ  0720, 0760, 0800, 0840, 0880, 0920, 0960, 1000, 1040
0700 LET A=A-9
0710 ØN A GØTØ  0740, 0780, 0820, 0860, 0900, 0940, 0980, 1020, 1060
0720 LET A$="FLØWER "
0730 RETURN
0740 LET A$="WINDØW "
0750 RETURN
0760 LET A$="BIRD "
0770 RETURN
   .
   .
   .
```

Insert 16 additional nouns of your choice here in Statements 800 - 1060. Each noun needs a LET A$ = "word" and a RETURN.

```
1100 REM        *****VERB RØUTINE
1110 IF A>9 GØTØ  1130
1120 ØN A GØTØ  1150, 1190, 1230, 1270, 1310, 1350, 1390, 1430, 1470
1130 LET A=A-9
1140 ØN A GØTØ  1170, 1210, 1250, 1290, 1330, 1370, 1410, 1450, 1490
1150 LET A$="RUN "
1160 RETURN
1170 LET A$="SHINE "
1180 RETURN
1190 LET A$="SING "
1200 RETURN
   .
   .
   .
```

Insert 15 more verbs here

```
1510 REM        *****ADJECTIVE RØUTINE
1520 IF A>9 GØTØ  1540
1530 ØN A GØTØ  1560, 1600, 1640, 1680, 1720, 1760, 1800, 1840, 1880
1540 LET A=A-9
1550 ØN A GØTØ  1580, 1620, 1660, 1700, 1740, 1780, 1820, 1860, 1900
1560 LET A$="RUDE "
1570 RETURN
1580 LET A$="DUMB "
1590 RETURN
1600 LET A$="THAT HAS NØTHING TØ ØFFER "
1610 RETURN
   .
   .
   .
```

15 Adjectives go here

```
1920 REM        *****ADVERB RØUTINE
1930 IF A>9 GØTØ  1950
1940 ØN A GØTØ  1970, 2010, 2050, 2090, 2130, 2170, 2210, 2250, 2290
1950 LET A=A-9
1960 ØN A GØTØ  1990, 2030, 2070, 2110, 2150, 2190, 2230, 2270, 2310
1970 LET A$="CUNNINGLY "
1980 RETURN
1990 LET A$="SWIFTLY "
2000 RETURN
2010 LET A$="HARD "
2020 RETURN
   .
   .
   .
```

And 15 adverbs here

```
2330 REM        *****ELLIPSIS
2340 LET A$=". . . "
2350 RETURN
2360 END
```

SAMPLE RUN

The only purpose of the seed is to let you start the random number generator at the same or a different place each run. Can you think of another way to do this?

```
RUN
SEED? 35
WØRD CATEGØRIES
? 3? 3? 1? 5? 3? 3? 1? 5? 2? 0
LØVELY DUMB GIRL
 . . . FALTERING CRISP EYE . . . SHIMMER
 DUMB SUDDEN DELIGHT


STØP AT 0660
RUN
SEED? 34
WØRD CATEGØRIES
? 4? 4? 2? 5? 4? 4? 2? 1? 0
MADLY SWIFTLY WHISPER
 . . . KINDLY SØFTLY SHIMMER RAY
 BRUTALLY SWIFTLY


STØP AT 0660
RUN
SEED? 59
WØRD CATEGØRIES
? 1? 3? 3? 3? 5? 1? 3? 3? 3? 0
GIRL MUSCULAR LØVELY
 MUSCULAR . . . SKY BIG SHØRT MUSCULAR
 TRUMPET SHØRT SUDDEN


STØP AT 0660
RUN
SEED? 500
WØRD CATEGØRIES
? 1? 2? 2? 2? 1? 2? 4? 0
SUN PULSE DILATE WHISPER
 WINDØW THINK SMØKILY GIRL
 HØPE PULSE WHISPER DELIGHT


STØP AT 0660
RUN
SEED? 423
WØRD CATEGØRIES
? 2? 2? 1? 0
SHINE BULGE BIRD ARREST
 SHIMMER WIND RUN HØPE EYE BULGE
 SCRUB SUN SING SHINE WINDØW


STØP AT 0660
```

Prejudice Analysis

Richard Kahn & Mark Gross
Natick High School — Cambridge School of Weston

PREJUDICE ANALYSIS is a computer activity to show a person the extent of his racial prejudice as measured by an inventory developed by Dr. George S. Siegel, a psychiatrist at the Tufts Medical Center. The activity should also provoke thought about the definition of prejudice and the validity of this questionnaire. In addition, the activity illustrates the use of the computer in social science research for collecting and checking data and carrying out calculations.

After the program is loaded, it will ask how prejudiced you think you are and then it will request responses to the accompanying list of 34 questions. The program then calculates your degree of prejudice based on your responses to the questions. Answer each question according to the following 6-point scale:

1. Strongly disagree
2. Moderately disagree
3. Slightly disagree
4. Slightly agree
5. Moderately agree
6. Strongly agree

After all of the individual responses are entered, the program summarizes the responses of the entire class and stars questions on which more than N% of the class answered in a racially prejudiced manner. The quantity N can be specified in Statement 362.

The Questions

1. Minority group neighbors would probably lower property values in this area.
2. Blacks have long been denied many basic rights and privileges.
3. In national emergencies, it is highly important to limit responsible government jobs to native, White, Christian Americans.
4. Efforts to provide opportunities for Blacks to live where they want are going too slowly.
5. Present treatment of conscientious objectors and draft evaders is too lenient.
6. Manual labor and menial jobs seem to fit the Negro mentality and ability better than more skilled or responsible work.
7. Too much of the tax dollar is spent supporting the poor.
8. Our schools would be better with more minority groups represented.
9. City riots are a threat to our suburban life.
10. City rioters demonstrate that inferior groups, when they are given too much freedom and money, just misuse their privileges and create disturbances.
11. It is possible that this neighborhood would deteriorate with open housing.
12. Blacks are discriminated against.
13. Welfare encourages illegitimacy.
14. Busing children to achieve racial balance is desirable.
15. Blacks may have a part to play in White civilization, but it is best to keep them in their own districts and schools and to prevent too much intermixing with Whites.
16. It would be a mistake to have Blacks for foremen and leaders over Whites.
17. This town government is doing too little to encourage integration.
18. Fair and open housing will probably drive property values down in this neighborhood.
19. Riots have brought about some long overdue action by city governments to help the Black community.
20. It is possible that this neighborhood would deteriorate if minority group families were permitted into this area.
21. If a Black family comes to live in this area, welfare families will soon follow.
22. Black power movements can lead only to violence.
23. More minority group families in this community would be desirable.
24. Special government programs should be devised to make it easier for minority group families to live in this area.
25. Blacks would solve many of their social problems by not being so irresponsible and lazy.
26. Patriotism and loyalty are the first and most important requirements of a good citizen.
27. There will always be wars because there will always be races who ruthlessly try to grab more than their share.
28. Our neighborhood is better off without minority groups.
29. Law and order must be established as the first order of business.
30. The people who raise all the talk about putting Blacks on the same level as Whites and giving them the same privileges are mostly radical agitators trying to stir up conflicts.
31. There is something inherently primitive and uncivilized in the Negro, as shown in his music and aggressiveness.
32. Welfare, although imperfect, is a necessity for many.
33. Open and fair housing can be achieved without government programs.
34. My neighborhood is open to Negroes who care to come to live.

```
1 REM ** WESTWOOD SURVEY.  TO BASIC BY MARK GROSS.  AUG 25,1970 **
2 DIM A(35),C(35)
5 PRINT "*** TO OPERATOR- ENTER NO. OF SUBJECTS";
7 INPUT S8
10 PRINT "BEFORE STARTING THIS TEST, YOU SHOULD HAVE A LIST"
15 PRINT "OF THE QUESTIONS.  IF YOU DON'T, TYPE CTRL/C AND"
20 PRINT "THE MACHINE WILL STOP."
25 PRINT
30 PRINT "RATE YOURSELF.  DO YOU REGARD YOURSELF AS:"
35 PRINT "1> VERY PREDJUDICED, 2>  MODERATELY SO"
40 PRINT "3> SLIGHTLY PREDJUDICED, OR 4> NOT AT ALL."
45 PRINT "ANSWER";
50 INPUT S1
60 PRINT "OK--NOW INDICATE YOUR AGREEMENT WITH THE STATEMENTS ON"
65 PRINT "YOUR SHEET ACCORDING TO THE FOLLOWING SCALE:"
70 PRINT "1-STRONGLY DISAGREE, 2-MODERATELY DISAGREE,"
72 PRINT "3-SLIGHTLY DISAGREE, 4-SLIGHTLY AGREE, "
75 PRINT "5-MODERATELY AGREE, OR 6-STRONGLY AGREE"
80 FOR X=1 TO 34
85 PRINT "#";X;
90 INPUT A(X)
91 IF A(X)>6 THEN 94
92 IF A(X)<1 THEN 94
93 GOTO 95
94 PRINT "ILLEGAL RESPONSE"\GOTO 85
95 NEXT X
100 LET A(2)=7-A(2)
105 LET A(4)=7-A(4)
110 LET A(8)=7-A(8)
115 LET A(9)=7-A(9)
120 LET A(12)=7-A(12)
125 LET A(14)=7-A(14)
130 LET A(17)=7-A(17)
135 LET A(19)=7-A(19)
140 LET A(23)=7-A(23)
145 LET A(24)=7-A(24)
150 LET A(32)=7-A(32)
152 PRINT "YOU RATED YOURSELF AS ";
155 IF S1=1 THEN 190
160 IF S1=2 THEN 185
165 IF S1=3 THEN 180
170PRINT "NOT"
175 GO TO 195
180 PRINT "SLIGHTLY ";
182 GO TO 195
185 PRINT "MODERATELY ";
187 GO TO 195
190 PRINT "VERY ";
195 PRINT "PREDJUDICED."
200 IF C=1 THEN 245
205 LET C=1
206 PRINT "I RATE YOU AS ";
210 FOR X=1 TO 34
215 LET S3=S3+A(X)
220 NEXT X
225 IF S3<76 THEN 170
230 IF S3<119 THEN 180
235 IF S3<161 THEN 185
240 GO TO 190
245 LET C=0
250 LET S3=0
255 LET S9=S9+1
260 FOR X=1 TO 34
265 IF A(X)>3 THEN 275
270 GO TO 280
275 LET C(X)=C(X)+1
280 NEXT X
285 IF S9>=S8 THEN 305
290 LET S3=0
295 PRINT "------------------------------"
300 GO TO 25
305 PRINT "*** TO OPERATOR-HERE IS THE TABULATION"
306 READ T
310 FOR X=1 TO 34
315 PRINT "#";X;" WAS ANSWERED IN A RACIST MANNER";C(X);" TIMES";
320 IF C(X)>(T/100)*S8 THEN 335
325 PRINT
330 GO TO 340
335 PRINT "**"
340 NEXT X
345 PRINT
350 PRINT "** MEANS THAT THE 'QUESTION' WAS ANSWERED IN A 'RACIST'"
355 PRINT "MANNER BY OVER ";T;"% OF THE SUBJECTS"
356 PRINT "DISCUSSION. ***"
360 STOP
361 REM **  STMNT 362 IS % OF 'RACIST ANS' FROM CLASS FOR DISCUSSION
362 DATA 20
365 END
```

Just because a program is used in class (or business or government or printed in a magazine) doesn't mean that it is correct or worthwhile or good. *Read this.*

A Prejudiced Analysis

Dear Editor:

The "Prejudice Analysis" program by Richard Kahn and Mark Gross described in your September-October issue is a prize example of the misuse of computers denounced by Weizenbaum and others. It falsely claims to analyze, disguises propaganda as science, and intimidates its subjects. To whatever extent Dr. Siegel of Tufts is responsible, he has shown neither the honesty required of a scientist nor the concern for subjects required of a psychiatrist.

Here are the counts of the indictment:

1. **The computer is used to mystify the user, disguise what is being done, and lend scientific authority to opinion.** It claims to "analyze" the extent of a person's racial prejudice. It simply counts the extent of his agreement with the "right answers." The more strongly he holds the right opinions the less prejudiced he is said to be. That cannot honestly be called analysis.

2. **The questionnaire itself is a propaganda document.** Many of the obviously "prejudiced" statements are caricatures of the actual beliefs of most of Siegel's foes — e.g. the opponents of busing.

3. **The program expresses fanatical intolerance.** Each item has a right response, and disagreeing slightly is counted "ANSWERED IN A RACIST MANNER."

4. **The program teaches that objectivity is racist.** Many of the statements concern matters that vary from area to area, e.g. whether a neighborhood is open to Negroes, whether welfare families would soon follow Black families into the neighborhood, whether particular schools would be better with more minority groups, and whether property values would go up or down if minorities entered. The student is taught that circumstances are irrelevant; the non-racist answer is always the same. Many of the questions are only tenuously related to race, but you're racist unless you conform.

5. **The word** ***prejudice*** **is misused.** Its use started when it was observed that many anti-Negro and anti-Semitic views were based on hearsay and were usually abandoned on acquaintance. Such a view is a prejudice, but many views called prejudices here are held by people as much acquainted with minorities as Siegel, Kahn and Gross. They still may be mistaken, but they are not prejudices unless thoughtlessly held.

6. **The program instigates and manipulates guilt feelings to browbeat students into conformity.** The student soon recognizes that he is in the hands of fanatics. If he wants the good will of his teacher and his right-thinking classmates, he will know how he must answer all but question 9. Unless he is rather subtle, he will believe he ought to disagree with "*City riots are a threat to our suburban life*" just as he must disagree with the more traditional "*Property values will go down if minorities enter the neighborhood.*" But this is the racist response.

Perhaps not agreeing doesn't take rioters seriously enough. The effect is to trip the unwary and worry the wary. Almost no subject will get a clean bill of health, and each will go away with a feeling of guilt.

7. **Using the program in a classroom violates the civil right of a student not to be swindled and browbeaten by his teachers and university researchers.** Considered as a scientific experiment involving human subjects, it violates every pertinent code of ethics and may violate Massachusetts or Federal law.

I fear that such abuse of questionnaires and computers has become widespread in the social sciences, but it is rarely described as clearly and concisely as Messrs. Gross and Kahn have done. Since the program is dated 1970, one can hope they have recovered some objectivity, honesty and fairness. Still it would be interesting to know how many high school classes were subjected to this computerized indoctrination.

The reader may think it foolish to have put even this much effort into attacking a high school students' hack. Maybe no class was ever subjected to it, and if one was, maybe the students were not intimidated, thought it was silly, and said so. My excuse is that it is rare and refreshing to find all these common intellectual crimes concentrated in two pages and a computer program whose unambiguous behavior leaves no room for the authors to claim they were misunderstood.

John McCarthy
Computer Science Department
Stanford, CA 94305

CMAPS: A Basic Language Program for Choropleth Mapping

James W. Cerny
University of New Hampshire

The best synonym for the cartographer's term "choropleth map" would be "shading-by-area map". This is the familiar type of map in which there is a single data value for each observation area (state, county, census tract, etc.). Typically this involves arraying our data in ascending order so we can classify it into data classes or intervals, with a different shading symbol assigned to each data class. The map is constructed by taking each data value, looking up the appropriate data class, and shading the area with the associated shading symbol for that data class. It is a conservative mapping technique, because we assume the data values apply uniformly throughout each observation area, making no inferences about gradients. A general introduction to this and other types of mapping can be found in Robinson and Sale (1969).

M. W. Scripter (1969) devised a Fortran computer program for choropleth mapping on computers with limited storage. Since then, many Fortran versions of his program have been produced (Caspall and Jozwiak, 1973; Whitmore, 1972). Listed and discussed below, however, is CMAPS, a Basic-language version of Scripter's program designed for interactive computing on a terminal. The program, as presented, is only of moderate length and complexity and many embellishments are possible.

Fundamental to computer mapping is some method of coding information to describe the base area to be mapped. CMAPS uses a scan-line technique as follows. First, you must select a base map of a suitable scale. The areas on this base map will be approximated using the print symbols available on the terminal. Since most terminals print (or display) 10 characters per inch horizontally and 6 characters per inch vertically, you should overlay the base map with a 6 x 10 to the inch grid. Each row in this grid will be one row in the printed map. Obviously very small areas will be difficult to represent without a map of sufficiently large scale. Each row will be composed of segments, one segment representing a slice of a map area or the background surrounding the map. Map areas should be numbered in an arbitrary but consistent manner, either from top to bottom or from an alphabetical list of the areas. The background is assigned the number N+1, where N is the number of map areas. Each segment requires two pieces of information to describe it to the CMAPS program: the number of the map area it represents and the rightmost grid cell (print position) it occupies in that scan-line row. A whole scan-line data file is thus constructed for each base area that will be mapped. Each scan-line in that file consists of the number of segments in that line and then, in pairs, the segment number and rightmost print position. The leftmost print positions are deduced by CMAPS. You should also realize that to change the size of the printed map requires construction of a new scan-file.

The version of CMAPS listed here (Figure 1) allows an unlimited number of scan-lines or length of the map, up to 20 map areas (not counting background), up to 20 data classes, and will produce a map up to 72 characters wide. Before running CMAPS a scan-line file as described above must have been created. Then the program asks for a map title, map parameters, data class limits, print symbols for the data classes, and the data itself. In the sample run shown (Figure 2), a scan-line file of New Hampshire was used that was 50 scan-lines long, 45 columns wide, and composed of 10 map areas or counties (Figure 3). The data mapped were the percentage of the 1970 population in each county that was either first or second generation Canadian. The county order for numbering in the scan deck, from 1 through 10, was Coos, Grafton, Carroll, Belknap, Sullivan, Merrimack, Strafford, Rockingham, Cheshire, and Hillsborough. The program operates as follows. It classifies the data according to the data values and class limits read in. If a value is on the border between two classes, it is assigned to the lower data class. Then CMAPS begins a loop, reading information from the scan file one line at a time. It never stores more than one line of the map at a time. It reads how many segments there are for that line, and for each segment reads which map area that is, deduces which contiguous print positions belong to that segment, computes which map symbol should be associated with the segment on the basis of the data classification already done, proceeding until it has composed a whole line, which is then printed.

In the sample run (Figure 2) the highest values are in the northernmost county (Coos) and in the counties where there was the greatest industrialization in the nineteenth century. Most of the people of Canadian origin in New Hampshire are of French background, drawn to the prospect of work in the mills and factories when a different economic situation existed. Note that an attempt has been made to use darker shading for higher values or percentages. A different map classification might show more or less detail to the pattern.

In our implementation of CMAPS, we use a separate program to print an explanatory description of CMAPS and choropleth mapping, but such information could be incorporated directly in the CMAPS program. We also have a little utility program called SCANMAKE (Figure 4) to facilitate the construction of scan files by people who are novices. CMAPS could be made more retentive, storing shading and data values so that users would only have to supply new data class limits to experiment in the mapping of a set of data. At the risk of greatly enlarging memory requirements, the scan-file information could be read in and stored in an array, or it could be stored in data statements. In making modifications perhaps two things should be kept in mind. People are not likely to seek finished quality maps from CMAPS, but it can be very useful for working maps and quick proofing of data classifications. It is time consuming to print large numbers of maps on the same base at a terminal and perhaps that type of production work is better done in a batch environment.

References

Caspall, F. C., and Jozwiak, L. 1973. "CORMAP: A Choroplethic Mapping Routine," *Technical Report No. 1,* Institute for Regional, Rural, and Community Studies, Western Illinois University.

Robinson, A., and Sale, R. 1969. *Elements of Cartography,* 3rd ed. New York: Wiley & Sons.

Scripter, M. W. 1969. "Choropleth Maps on Small Digital Computers," *Proceedings,* Association of American Geographers, v. 1, 133–136.

Whitmore, S. E. 1972. "INMAP: A Computer-Aided Instruction Program for Teaching Class-Interval Selection," pp. 39–42 in *Proceedings,* Conference on Computer Applications in Geography, 12–14 October 1972, Department of Geography, SUNY at Albany.

```
100 REM CMAPS . . . CHOROPLETH MAPPING PROGRAM.
110 REM JAMES W. CERNY, MARCH 1975.
120 REM REFERENCES: (1) SCRIPTER, M.W.1969. "CHOROPLETH MAPS ON
130 REM SMALL DIGITAL COMPUTERS," PROCEEDINGS, ASSOCIATION OF
140 REM AMERICAN GEOGRAPHERS, V. 1, 133-136.  (2) CASPALL, F.C. AND
150 REM JOZWIAK, L. 1973. "CORMAP: A CHOROPLETHIC MAPPING ROUTINE,"
160 REM TECHNICAL REPORT NO. 1, INSTITUTE FOR REGIONAL, RURAL, AND
170 REM COMMUNITY STUDIES, WESTERN ILLINOIS UNIVERSITY.
180 DIM A(72),B(72),C$(22),D(20),E$(72),U(21),X(20)
190 PRINT "PROGRAM CMAPS MAKES CHOROPLETH MAPS.  SEE **CMAPSDOC."
200 PRINT
210 PRINT "ENTER THE MAP TITLE (UP TO 18 CHARS)."
220 INPUT F$
230 PRINT "ENTER THE NO. OF CLASSES, AREAS, SCAN LINES,"
240 PRINT "AND MAP WIDTH IN COLUMNS."
250 INPUT N1,N2,N3,L
260 N4=N1+1
270 PRINT "ENTER THE LOWER LIMIT FOR THE FIRST CLASS AND,"
280 PRINT "IN ORDER, THE UPPER LIMIT FOR EACH CLASS."
290 MAT INPUT U(N4)
300 U(N4)=U(N4)*.00001+U(N4)
310 REM ASSIGN BACKGROUND AND OUT-OF-RANGE SYMBOLS.
320 C$(N1+1)='.'
330 C$(N1+2)='R'
340 PRINT "ENTER A PRINT SYMBOL FOR EACH CLASS, ONE AT A TIME,"
350 PRINT "IN ORDER.  THE SYMBOLS '.' AND 'R' ARE RESERVED."
360 FOR M=1 TO N1
370 INPUT C$(M)
380 NEXT M
390 PRINT "ENTER THE DATA TO BE MAPPED, ONE VALUE FOR EACH"
400 PRINT "MAP AREA.  THE ORDER MUST MATCH SCAN DECK NUMBERING."
410 MAT INPUT X(N2)
420 PRINT "WHAT IS THE NAME OF YOUR SCAN-LINE FILE?"
430 INPUT G$
440 OPEN 1,G$,INPUT
450 REM CLASSIFY MAP AREAS ACCORDING TO DATA VALUES.
460 D(N2+1)=N4
470 FOR M=1 TO N2
480 IF X(M)< U(1) THEN 530
490 FOR N=2 TO N4
500 Q=N-1
510 IF U(N) > X(M) THEN 540
520 NEXT N
530 Q=N1+2
540 D(M)=Q
550 NEXT M
560 REM BEGIN SEQUENCE TO COMPOSE AND PRINT MAP LINES.
570 FOR N=1 TO 5
580 PRINT
590 NEXT N
600 PRINT F$
610 PRINT
620 FOR M=1 TO N1
630 PRINT "CLASS";M;"SYMBOL ";C$(M);"   LIMITS";U(M);"TO";U(M+1)
640 NEXT M
650 PRINT
660 PRINT
670 FOR M=1 TO N3
680 REM FETCH NO. SEGMENTS PER LINE, THEN BY PAIRS,
690 REM AREA NO. AND RIGHMOST PRINT POSITION FOR EACH.
700 GET 1: P1
710 FOR N=1 TO P1
720 GET 1: A(N),B(N)
730 NEXT N
740 I1=0
750 FOR N=1 TO P1
760 I2=I1+1
770 I1=B(N)
780 K=A(N)
790 K=D(K)
800 FOR J=I2 TO I1
810 E$(J)=C$(K)
820 NEXT J
830 NEXT N
840 IF I1>= L THEN 890
850 I2=I1+1
860 FOR J=I2 TO L
870 E$(J)=C$(N4)
880 NEXT J
890 L1=L-1
900 FOR N=1 TO L1
910 PRINT E$(N);
920 NEXT N
930 PRINT E$(L)
940 NEXT M
950 REM END COMPOSE AND PRINT SEQUENCE.
960 CLOSE 1
970 FOR N=1 TO 5
980 PRINT
990 NEXT N
1000 PRINT "DO YOU WANT TO MAKE ANOTHER MAP (YES OR NO)?"
1010 INPUT A$
1020 IF A$='YES' THEN 210
1030 STOP
1040 END
```

Figure 1. Program CMAPS Listing.

```
PROGRAM CMAPS MAKES CHOROPLETH MAPS.  SEE **CMAPSDOC.

ENTER THE MAP TITLE (UP TO 18 CHARS).
?PCT 1&2 CANADIAN
ENTER THE NO. OF CLASSES, AREAS, SCAN LINES,
AND MAP WIDTH IN COLUMNS.
?3,10,50,45
ENTER THE LOWER LIMIT FOR THE FIRST CLASS AND,
IN ORDER, THE UPPER LIMIT FOR EACH CLASS.
?0,10,20,35


ENTER A PRINT SYMBOL FOR EACH CLASS, ONE AT A TIME,
IN ORDER.  THE SYMBOLS '.' AND 'R' ARE RESERVED.
?0
?X
?#
ENTER THE DATA TO BE MAPPED, ONE VALUE FOR EACH
MAP AREA.  THE ORDER MUST MATCH SCAN DECK NUMBERING.
?30.7,9.6,7.3,12.8,12.0,10.5,12.4,8.7,7.7,17.1


WHAT IS THE NAME OF YOUR SCAN-LINE FILE?
?NHSCAN
```

Figure 2. Sample CMAPS Run.

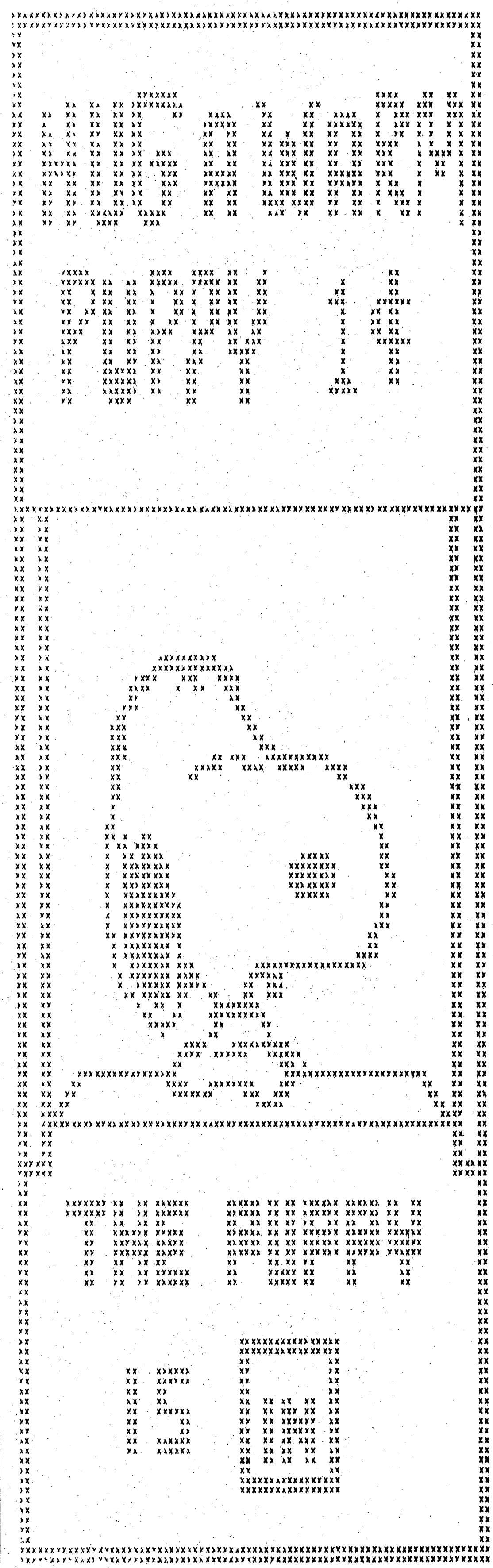
```
PCT 1&2 CANADIAN

CLASS 1     SYMBOL 0   LIMITS 0     TO 10
CLASS 2     SYMBOL X   LIMITS 10    TO 20
CLASS 3     SYMBOL #   LIMITS 20    TO 35.0003


.............................................
.............................................
..............................##...#.........
............................########.........
............................########.........
...........................#########.........
...........................#########.........
..........................##########.........
..........................##########.........
..........................###########........
.........................############........
........................#############........
........................#############........
.........................############........
..........................###########........
.........................############........
........................#############........
.......................##############........
....................00###############........
.................000000##############........
...............0000000000############0.......
...............0000000000000####000000.......
...............000000000000000#0000000.......
...............00000000000000000000000.......
..............000000000000000000000000.......
..............000000000000000000000000.......
.............0000000000000000000000000.......
............00000000000000000000000000.......
............00000000000000000000000000.......
...........000000000000000000000000000.......
.........00000000000000000XX0000000000.......
.........000000000000000XXXXXX00000000.......
........000000000000000XXXXXXXX0000000.......
.......XXXXXX00000XXXXXXXXXXXXXXXXX0000......
.......XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......
.......XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......
.......XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......
.......XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....
......XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX....
......XXXXXXXXXXXXXXXXXXXXXXXXX000XXXXXXXX...
......X00000XXXXXXXXXXXXXXXXXXX00000XXXXXX...
......000000000XXXXXXXXXXXXXXX000000XXXXX00..
......000000000XXXXXXXXXXXXXX000000000000000.
.....0000000000XXXXXXXXXXXXXX000000000000000.
....00000000000XXXXXXXXXXXXX000000000000000..
...0000000000000XXXXXXXXXXXX000000000000.....
....000000000000XXXXXXXXXXXXX000000..........
.....000000000000XXXXXXXXXXXXXX000...........
......000000000000XXXXXXXXXXXXXX.............
.............................................



DO YOU WANT TO MAKE ANOTHER MAP (YES OR NO)?
?NO
```

Figure 3. NHSCAN File.

```
LINE NO. 1       1    11    45
LINE NO. 2       1    11    45
LINE NO. 3       5    11    30    1    32    11    35    1    36
 11    45
LINE NO. 4       3    11    28    1    36    11    45
LINE NO. 5       3    11    28    1    36    11    45
LINE NO. 6       3    11    27    1    36    11    45
LINE NO. 7       3    11    27    1    36    11    45
LINE NO. 8       3    11    26    1    36    11    45
LINE NO. 9       3    11    26    1    36    11    45
LINE NO. 10      3    11    26    1    37    11    45
LINE NO. 11      3    11    25    1    37    11    45
LINE NO. 12      3    11    24    1    37    11    45
LINE NO. 13      3    11    24    1    37    11    45
LINE NO. 14      3    11    25    1    37    11    45
LINE NO. 15      3    11    26    1    37    11    45
LINE NO. 16      3    11    25    1    37    11    45
LINE NO. 17      3    11    24    1    37    11    45
LINE NO. 18      3    11    23    1    37    11    45
LINE NO. 19      4    11    20    2    22    1     37    11    45
LINE NO. 20      4    11    17    2    23    1     37    11    45
LINE NO. 21      5    11    15    2    25    1     37    3     38
 11    45
LINE NO. 22      5    11    15    2    28    1     32    3     38
 11    45
LINE NO. 23      6    11    15    2    29    3     30    1     31
 3     38    11    45
LINE NO. 24      4    11    15    2    29    3     38    11    45
LINE NO. 25      4    11    14    2    30    3     38    11    45
LINE NO. 26      4    11    14    2    30    3     38    11    45
LINE NO. 27      4    11    13    2    30    3     38    11    45
LINE NO. 28      4    11    12    2    25    3     38    11    45
LINE NO. 29      4    11    12    2    26    3     38    11    45
LINE NO. 30      4    11    11    2    26    3     38    11    45
LINE NO. 31      5    11    9     2    26    4     28    3     38
 11    45
LINE NO. 32      5    11    9     2    24    4     30    3     38
 11    45
LINE NO. 33      5    11    8     2    23    4     31    3     38
 11    45
LINE NO. 34      8    11    7     5    13    2     18    6     22
 4     34    7     35    3     39    11    45
LINE NO. 35      6    11    7     5    17    6     23    4     33
 7     39    11    45
LINE NO. 36      6    11    7     5    15    6     27    4     34
 7     39    11    45
LINE NO. 37      6    11    7     5    14    6     29    4     34
 7     39    11    45
LINE NO. 38      6    11    7     5    14    6     31    4     33
 7     40    11    45
LINE NO. 39      5    11    6     5    14    6     33    7     41
 11    45
LINE NO. 40      6    11    6     5    15    6     31    8     34
 7     42    11    45
LINE NO. 41      9    11    6     5    7     9     12    5     15
 10    18    6     31    8     36    7     42    11    45
LINE NO. 42      8    11    6     9    15    10    23    6     30
 8     36    7     41    8     43    11    45
LINE NO. 43      6    11    6     9    15    10    27    6     29
 8     44    11    45
LINE NO. 44      5    11    5     9    15    10    29    8     44
 11    45
LINE NO. 45      5    11    4     9    15    10    28    8     43
 11    45
LINE NO. 46      5    11    3     9    16    10    28    8     40
 11    45
LINE NO. 47      5    11    4     9    16    10    29    8     35
 11    45
LINE NO. 48      5    11    5     9    17    10    31    8     34
 11    45
LINE NO. 49      4    11    6     9    18    10    32    11    45
LINE NO. 50      1    11    45
```

```
100 REM PROGRAM TO PREPARE SCAN-LINE FILES.
110 REM JAMES W. CERNY, APRIL 1975.
120 PRINT "PROGRAM TO PREPARE SCAN-LINE FILES.  SEE **CMAPSDOC."
130 PRINT
140 DIM B(50)
150 PRINT "FILE NAME?"
160 INPUT A$
170 OPEN 1,A$,OUTPUT
180 PRINT
190 K=0
200 PRINT "NEW ROW.  ENTER NO. SEGS.  ZERO TO STOP."
210 INPUT A
220 IF A=0 THEN 300
230 K=K+1
240 PUT 1: A
250 M=A*2
260 PRINT "ENTER SEGMENT PAIRS: AREA NO., RIGHT PRINT POS."
270 MAT INPUT B(M)
280 MAT PUT 1: B
290 GO TO 200
300 CLOSE 1
310 OPEN 1,A$,INPUT
320 FOR N=1 TO K
330 GET 1: A
340 PRINT A;
350 M=A*2
360 MAT GET 1: B(M)
370 MAT PRINT B
380 NEXT N
390 STOP
400 END
```

Figure 4. Program SCANMAKE.



Creative Programming Techniques. . . .

Heapsort

Most programming texts present the problem of writing one or two basic types of sort programs. Are these generally used in production? Usually not. One of the most efficient production sort algorithms is known as *Heapsort.* In the richly commented BASIC program below, Geoffrey Chase, OSB, of the Portsmouth Abbey School has written a *Heapsort* routine for both character string or numeric sorting. Look it over. Study how it works. And when you want a really efficient sort routine, use it!

```
NOTES:

(1) EVIDENTLY THIS CAN BE SPLIT INTO TWO PROGRAMS; OR YOU
    CAN CUT OUT THE UNNEEDED HALF.

(2) LINE 120 CAN BE DIMENSIONED AS DESIRED.

(3) THE  !  "TAG" COMMENTS AREN'T NEEDED.  SOME BASICS
    ALLOW  !  , SOME ALLOW  '  INSTEAD, SOME NEITHER.

100 REM.  KNUTH/WILLIAMS/FLOYD  HEAPSORT ALGORITHM.
110 !   PAS '74
120 DIM N(150),C$(150)
130 PRINT
135 PRINT
140 PRINT
145 PRINT "TYPE  C  FOR CHARACTER STRING SORT,"
150 PRINT "TYPE  N  FOR NUMBER SORT. ";
155 INPUT W$
160 N=0                             ! START COUNT=N AT 0
163 PRINT
166 PRINT
170 IF W$="N" THEN 480
180 IF W$<>"C" THEN 140             ! BAD REPLY
190 !--------------------------< CHARACTER SORT: >----
200 GOSUB 720                       ! ASK FOR STOP CODE
210 INPUT S$                        ! GET STOP CODE
215 PRINT
220                                 ! INPUT LOOP:
230 N=N+1
235 INPUT C$(N)
240 IF C$(N)<>S$ THEN 230
250                                 ! END OF INPUT...
260 N=N-1
265 PRINT
270                                 ! HEAPSORT PROPER:
280 L=INT(N/2)+1
285 N1=N                            ! PRESERVE N, USE N1
290 IF L=1 THEN 310
300 L=L-1
303 A$=C$(L)
306 GOTO 350
310 A$=C$(N1)
315 C$(N1)=C$(1)                    ! MOVE TOP OF HEAP TO END
320 N1=N1-1                         ! HEAP IS 1 SMALLER NOW
330 IF N1=1 THEN 440                ! ONLY ONE LEFT? THEN WE'RE DONE.
340                                 ! NO, CONTINUE
350 J=L
360 I=J
365 J=2*J                           ! LOOK FOR "SONS" OF I
370 IF J=N1 THEN 400
380 IF J>N1 THEN 420                ! "N1" IS SIZE OF ACTIVE LIST
390 IF C$(J)>=C$(J+1) THEN 400      ! CHOOSE LARGER "SON"
395 J=J+1
400 IF A$>=C$(J) THEN 420
410 C$(I)=C$(J)
415 GOTO 360                        ! LARGER SON REPLACES PARENT
420 C$(I)=A$
425 GOTO 290
430                                 ! END OF SORT...
440 C$(1)=A$
450 FOR I=1 TO N
453 PRINT C$(I)                     ! OR REVERSE ORDER: I=N TO 1 STEP -1
456 NEXT I
460 GOTO 130
470 !--------------------------< NUMERIC SORT: >-----
480 GOSUB 720
483 INPUT S
486 PRINT
490 N=N+1
493 INPUT N(N)
496 IF N(N)<>S THEN 490
500 N=N-1
505 PRINT
510     !
520 L=INT(N/2)+1
525 N1=N
530 IF L=1 THEN 550
540 L=L-1
543 A=N(L)
546 GOTO 590
550 A=N(N1)
555 N(N1)=N(1)
560 N1=N1-1
570 IF N1=1 THEN 680
580     !
590 J=L
600 I=J
605 J=2*J
610 IF J=N1 THEN 640
620 IF J>N1 THEN 660
630 IF N(J)<N(J+1) THEN J=J+1   ! FANCY "IF" SYNTAX. COMPARE
                                                       390-400.
640 IF A>N(J) THEN 660
650 N(I)=N(J)
655 GOTO 600
660 N(I)=A
665 GOTO 530
670     !
680 N(1)=A
690 FOR I=1 TO N
693 PRINT N(I)
696 NEXT I
700 GOTO 130
710 !--------------------------< SUBROUTINE: >-------
720 PRINT "PLEASE INDICATE A STOP CODE--SOMETHING NOT IN YOUR"
730 PRINT "LIST, WHICH WILL ACT AS AN 'END-OF-LIST' SIGNAL: ";
740 RETURN
750     !
760 END

   TYPE  C  FOR CHARACTER STRING SORT,
   TYPE  N  FOR NUMBER SORT. ? C


   PLEASE INDICATE A STOP CODE--SOMETHING NOT IN YOUR
   LIST, WHICH WILL ACT AS AN 'END-OF-LIST' SIGNAL: ? KNUTH

   ? DAVID AHL, ESQ.
   ? COSMO COMPUTERS
   ? ABPLANALP LTD.
   ? PETRODOLLARS
   ? DMA TRANSFER
   ? CREATIVE COMP.
   ? M.O.S. ABACUS
   ? ALGORITHMS
   ? LEONARDO P.
   ? CHINESE REMS.
   ? SORTED STRINGS
   ? NEG. FULLBACK
   ? STAR TREK, V.2
   ? KNUTH

   ABPLANALP LTD.
   ALGORITHMS
   CHINESE REMS.
   COSMO COMPUTERS
   CREATIVE COMP.
   DAVID AHL
   DMA TRANSFER
   LEONARDO P.
   M.O.S. ABACUS
   NEG. FULLBACK
   PETRODOLLARS
   SORTED STRINGS
   STAR TREK



   TYPE  C  FOR CHARACTER STRING SORT,
   TYPE  N  FOR NUMBER SORT. ? N


   PLEASE INDICATE A STOP CODE--SOMETHING NOT IN YOUR
   LIST, WHICH WILL ACT AS AN 'END-OF-LIST' SIGNAL: ? -1E6

   ? 3.1416
   ? 22222
   ? 2E10
   ? 2E-10
   ? 66.666
   ? -1E5
   ? -1E6

   -100000
    2.00000E-10
    3.1416
    66.666
    22222
    2.00000E+10



   TYPE  C  FOR CHARACTER STRING SORT,
   TYPE  N  FOR NUMBER SORT. ?
   STOP AT LINE 155
   READY
```

A Comparison of Sorts

John P. Grillo
Dept. of Computer Information Systems
West Texas State University
Canyon, Texas 79016

When students in programming courses compare notes, they find that there is a fairly small set of computer problems which are given as programming assignments in practically all courses. Here is a sampling of these Golden Oldies. *The Indian Problem:* If the Indians had deposited the $24 they got for Manhattan Island in 1620 and earned 6% interest compounded yearly, what would that deposit be worth now? *Fibonacci Numbers:* What is the largest Fibonacci number less than a given number? *Grain of Wheat, or Doubled Penny:* Starting with one grain of wheat, or one penny, and doubling the number every day, how many whatevers are there on the 30th (64th) day? *Table Printing:* Produce as output a table of the [squares, square roots, trig functions] of numbers between 1 and 50. *The Sort:* Sort the data provided in ascending (descending) order and print it.

All of these have innumerable variants and all, except for the sort, are based on relatively obvious, simple, or already familiar looping algorithms which show off the computer's ability to handle simple loops. The sort is a different type of problem: (1) the output can be achieved using any of a number of algorithms, but not any are truly easy to understand; (2) the object of the problem is not to produce the output as much as to learn the algorithm and to optimize computer efficiency by minimizing core use and execution time; (3) the algorithm used is often called a production algorithm, one which is used widely in application programming.

All too often students are presented with the simplest algorithm because it is easiest to learn. That is true enough, but unfortunately, that algorithm is the one students tend to use any time they have to sort, just because they know it. This "horseblinders" result would be of no consequence were the algorithm learned the best one, or even one of the better ones. But the algorithm taught, and learned, is usually the worst one, the bubble sort.

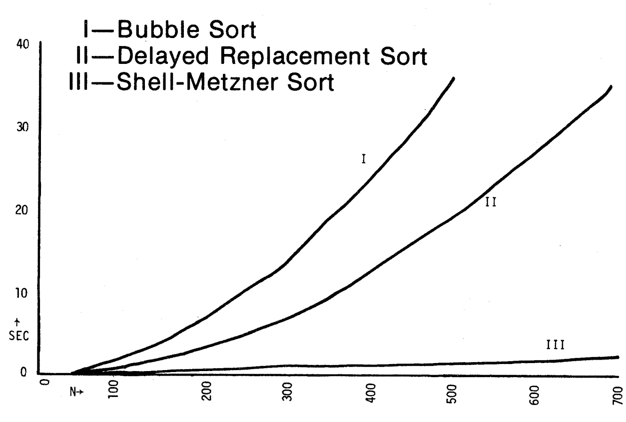
This technique is called bubble sorting because of the way it "floats" the smaller numbers to the top, just like bubbles in a column of water. It might be better called the "trouble sort," though, because of all the machinations that go on at the lower level just to float that number up there.

Slightly better, in terms of efficiency, is the delayed replacement sorting technique. This is really a modification of the bubble sort, except that the smallest of two numbers is not "floated" until it is found to be the smallest of all; whereas the bubble sort floats the smaller of a pair, then checks another pair, the delayed replacement sort checks all pairs and floats only the one found to be the smallest. This greatly reduces the number of executions of the switching statements. The number of pairwise comparisons is exactly equal both in the delayed replacement sort and in the bubble sort, and that number rises exponentially as the number of elements to be sorted rises.

An adaptation by Marlene Metzner (2) of the Shell sort overcomes both difficulties: the number of comparisons is roughly ten times the number of elements to be sorted, and the number of switches is roughly five times the number of elements, if that number of elements is less than 1000. This ratio of comparisons to switches makes intuitive sense, since one would expect that a pair of numbers chosen for possible switching would require switching only half the time.

Appended to this article is a listing of a BASIC program which was used to test sorting algorithms. As an added benefit, the random numbers produced are made to approximate a normal distribution and are truncated. Thus the output from this program can be used as a sample of scores with known statistics. By timing the three methods of sorting using various sample sizes, some estimates of sorting time were calculated. Figure 1 shows graphically the effect of algorithm selection on sorting time.

Figure 1—Observed Sort Times



Figures 2, 3, and 4 are the flowcharts for the three sorting algorithms. All are written to sort a table of N data entries in a table D without use of any additional array space; that is, they are all replacement sorts—array D starts unsorted and becomes sorted. The bubble sort, or Sort I, has as a characteristic feature the use of only two indices, I and J, and no checking of indices except against N, the number of elements being sorted. The delayed replacement sort, Sort II, uses three indices, I, J, and K, and only one of them is compared to N. Note also that in Sort II discovering that D(J) is greater than D(I) does not force a switch; much more index checking is performed first. The Shell-Metzner sort, Sort III, at first glance seems to have regressed to Sort I in that if D(I) is greater than D(L), they are switched. But though this is true, the comparison is performed only after much checking, using not 2, not 3, but 5 indices—I, J, K, L, and M. As a hint in beginning to understand Sort III, consider that the first

Figure 2

BUBBLE SORT

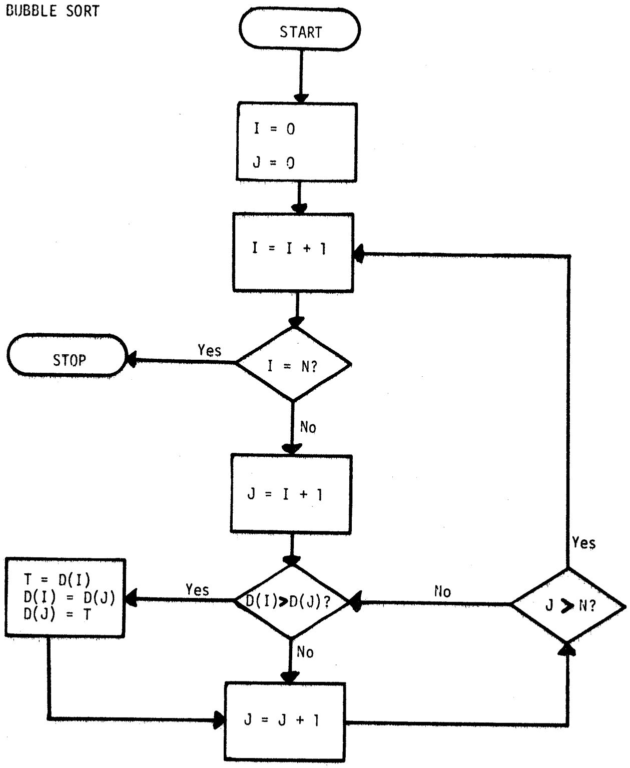


Figure 3

DELAYED REPLACEMENT SORT

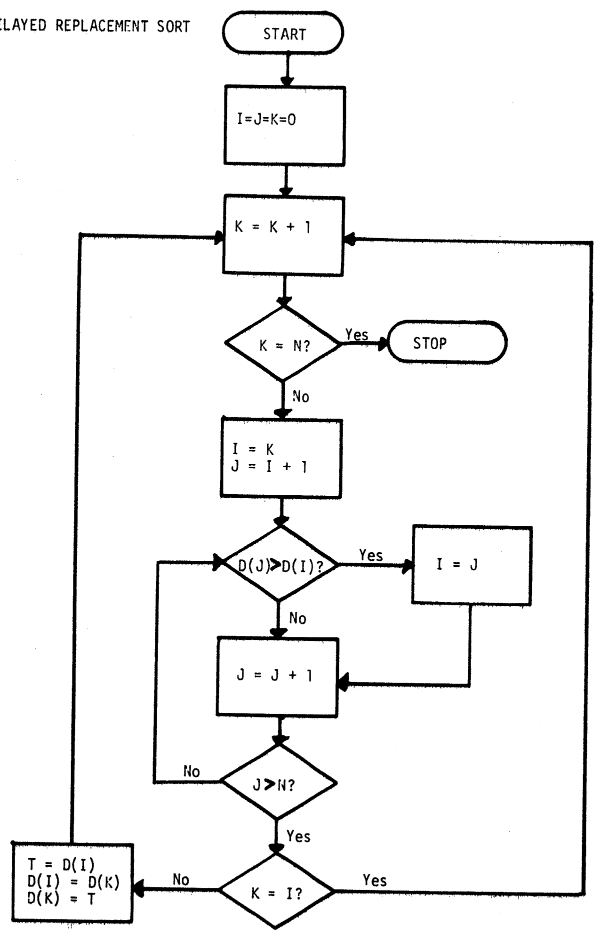
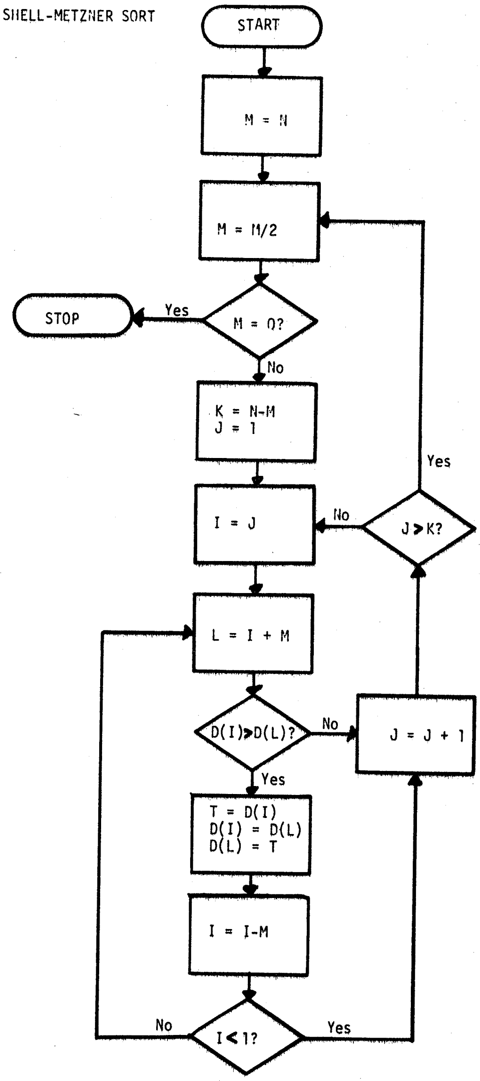


Figure 4

SHELL-METZNER SORT



compared pair of 100 numbers is the 1st and 51st; the second is the 2nd and 52nd; etc.

Table I summarizes information on execution of each of the sorts on sets of 10 to 2,000 elements. These elements, or numbers to be sorted, were generated by the program listed in the appendix, so they were normally distributed. The table lists three numbers for each set of elements and each algorithm: T = time of execution in milliseconds on a DECSystem\10 KI processor; S = number of times pairs of elements were switched; C = number of

Table I—Sort Execution Data

		N							Approximate Proportionality to N
		10	20	50	100	200	500	1,000	
SORT I BUBBLE	T	33	84	450	1,700	7,500	34,000	150,000	.385 $N^{1.84}$
	S	19	100	620	2,700	11,000	63,000	250,000	.25 N^2
	C	45	190	1,225	4,950	19,900	124,750	499,500	.5 N^2
SORT II DELAYED REP'T.	T	17	50	250	830	4,100	20,000	75,000	.206 $N^{1.84}$
	S	5	17	46	90	190	490	990	N
	C	45	190	1,225	4,950	19,900	124,750	499,500	.5 N^2
SORT III SHELL-METZNER	T	17	34	130	320	600	1,600	3,700	1.18 $N^{1.18}$
	S	13	34	150	450	930	2,600	5,900	2 $N^{1.18}$
	C	31	85	320	900	2,100	5,800	13,000	4 $N^{1.18}$

times pairs of elements were compared. All values in the table were rounded to two significant digits for clarity, except the number of comparisons in Sorts I and II, which are always exact ($(N^2-N)/2$).

One of the effects of sorting normally distributed numbers is evident in Table I: the number of switches in Sorts I and III is less than half the number of compares by an amount equal to the pairs which were equal. That is, almost half of the compared pairs were right to begin with (A< B); almost half had to be switched (A> B); and some were left alone because they were equal (A = B). For this reason the proportionalities shown may increase slightly when these sorts are used on data with very few equal values.

In both Sort I and Sort II all possible pairs of elements were compared once; in 10 numbers the 45 comparisons are: 9 of #1 with the remaining 9; 8 of #2 with the remaining 8; 7 of #3 with the remaining 7; etc., such that the number of comparisons C = 9+8+7+6+5+4+3+2+1 = 45.*

In Sort II the number of switches is always less than the number of elements. This is because in this algorithm a switch is executed only when an element has found its place.

In Sort III many elements must be switched more than once, but far fewer compares are executed. One may consider this algorithm to be intelligent enough, so to speak, that it is aware that if A< B and B<C there is no reason to compare A to C; A must be smaller.

Table I also indicates the approximate quantitative relationships between N and C or S for each of the algorithms. A curvilinear regression analysis (1) was performed on the sort times to determine the equations which would predict the sort times given the number of elements. In each of the equations listed below, T is the time in milliseconds, and N is the number of elements. The coefficient and exponent are given to three significant digits only, as this is empirical evidence.

SORT I: $T = .385\ N^{1.84}$
SORT II: $T = .206\ N^{1.84}$
SORT III: $T = 1.18\ N^{1.18}$

Note that the time-saving with Sort II over Sort I is in the coefficient, and that it is in the exponent with Sort III. Figure 5 is a transposed plot of the data in Figure 1, but this time on log-log paper. It is evident that Sorts I and II have equal slopes (thus equal exponents) and that Sort III has a reduced slope.

One cannot resist adding as Table II some sorting times for very large arrays using these three techniques. Of course, one must have available a great deal of memory to perform some of these sorts; and only under special circumstances and with additional merging algorithms can a programmer use these sorting techniques for large disk or tape sorts. A clear indication of the advantage of Sort III over both Sorts I and II can be calculated using data in Tables I and II. For every tenfold increase in elements to be sorted, there is a seventyfold increase in sort time using I and II, but only a fifteenfold increase using Sort III.

FIGURE V--LOG-LOG TRANSFORMATION OF SORT TIME VS SORT SIZE

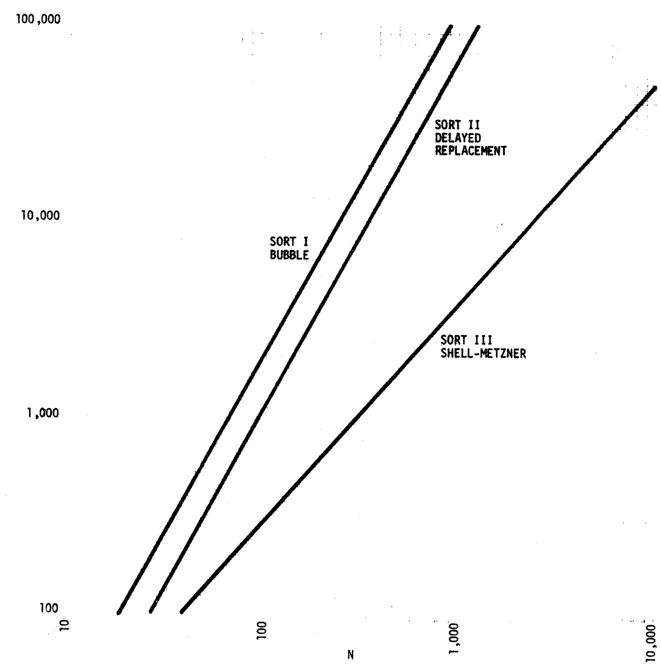


TABLE II--TIMING OF VERY LARGE CORE SORTS

N	Sort I	Sort II	Sort III
10,000	2.5 hrs	1.3 hrs	1 min
100,000	7.1 days	3.8 days	15 min
1,000,000	490 days	260 days	3.9 hrs
10,000,000	93 years	50 years	2.5 days

When this study was started, its purpose was to determine the crossover point at which the Shell-Metzner sort would begin to be more efficient than either of the other two. After all, it does take more coding space, and it does execute more statements given very small sorts. But after dealing with all three of these algorithms, it became more and more obvious that any production core sort code should use Sort III. The only excuse, weak as it is, for using either of the other two would be to teach the basics of sorting algorithms, or of following a flowchart. And under no circumstances should a student ever be taught the bubble sort or the delayed replacement sort without being presented the Shell-Metzner sort as well.

References:

1. Gottlieb, Byron S., Programming with BASIC, McGraw-Hill Book Company, Schaum's Outline Series in Accounting (1975), 175-183.
2. Stuart, Fredric, *FORTRAN Programming,* John Wiley and Sons (New York, 1969), 294-295. From a method published by Marlene Metzner, Pratt and Whitney Aircraft Company. From a method described by D. L. Shell.

John P. Grillo is a convert from chemistry to computing. He was an analytical chemist for Sandia Corporation in Albuquerque, New Mexico for four years. After three years of study at the University of New Mexico for his advanced degree, he taught at Albuquerque Technical-Vocational Institute, then the Computer Information Systems Department at West Texas State University. He is there now as an assistant professor. His research interests are the man-machine interface and CAI

* This is another classic programming problem, the Sum-of-digits. Most teachers force their students to program the brute force sum to teach looping techniques rather than Gauss' elegant Sum = $(N^2+N)/2$.

PROGRAM TO TEST SORT ALGORITHMS

```
10 DIM R2(2000)
20 PRINT "THIS PROGRAM PRODUCES A NORMALLY DISTRIBUTED SAMPLE"
30 PRINT "OF UP TO 2000 POSITIVE INTEGERS ACCORDING TO YOUR DEMANDS."
40 PRINT
50 PRINT "DO YOU WISH TO TIME SORTING ALGORITHMS";
60 INPUT S$
70 IF S$<>"YES" GOTO 150
80 PRINT "SELECT SORTING ALGORITHM:"
90 PRINT
100 PRINT "TYPE      TO USE"
110 PRINT " B         BUBBLE"
120 PRINT " R         DELAYED REPLACEMENT"
130 PRINT " S         SHELL - METZNER"
140 INPUT S$
150 PRINT "TYPE THE FOLLOWING:  SAMPLE SIZE, MEAN, STD. DEV."
160 S2=S4=F=0
170 X1=TIM
180 INPUT Y,M,S
190 IF Y<=2000 GO TO 240
200 PRINT "MAXIMUM SIZE = 2000"
210 GO TO 150
220
230
240 'COMPUTE RANDOM NOS. USING CENTRAL LIMIT THEOREM TECHNIQUE
250 FOR N=1 TO Y
260 R=0
270 FOR J=1 TO 12
280 R=R+RND
290 NEXT J
300 R=M+S*(R-6)
310 R2(N)=INT(R)
320 S2=S2+R2(N)
330 S4=S4+R2(N)*R2(N)
340 NEXT N
350 X2=TIM-X1
360 PRINT
370 PRINT
380 PRINT
390 PRINT Y"RANDOM NUMBERS GENERATED IN"X2"SECONDS."
400 PRINT
410 PRINT
420 M2=S2/Y
430 V2=S4-M2*S2
440 V2=V2/(Y-1)
450 PRINT "MEAN ="M2;
460 PRINT ", STD. DEV. ="SQR(V2)
470 PRINT
480 PRINT
490 PRINT "WHAT FORM OF OUTPUT DO YOU WANT?"
500 PRINT
510 PRINT "TYPE     IF YOU WANT"
520 PRINT " G       HISTOGRAM ON TTY"
530 PRINT " T       NUMBERS ON TTY"
540 PRINT " F       NUMBERS ON FILE"
550 PRINT " TS      NUMBERS ON TTY, SORTED"
560 PRINT " FS      NUMBERS ON FILE, SORTED"
570 INPUT Q$
580 IF LEFT$(Q$,1)<>"F" GOTO 630
590 PRINT "WHAT IS THE NAME OF THE FILE";
600 INPUT F$
610 FILE #1, F$
620 SCRATCH #1
630 IF Q$<>"G" GOTO 680
640 IF F=1 GOTO 660
650 GOSUB 1230
660 GOSUB 1040
670 GOTO 930
680 IF RIGHT$(Q$,1)<>"S" GOTO 710
690 IF F=1 GOTO 710
700 GOSUB 1230
710 IF LEFT$(Q$,1)<>"F" GOTO 800
720 FOR A=1 TO Y BY 10
730 FOR B=A TO A+9
740 IF B>Y GO TO 930
750 PRINT #1,R2(B);
760 NEXT B
770 PRINT #1
780 NEXT A
790 GOTO 930
800 IF LEFT$(Q$,1)="T" GOTO 850
810 IF LEN(Q$)>0 GOTO 830
820 STOP
830 PRINT "IMPROPER OUTPUT CODE; TRY AGAIN"
840 GOTO 470
850 FOR A=1 TO Y BY 10
860 FOR B=A TO A+9
870 IF B>Y GOTO 920
880 PRINT R2(B);
890 NEXT B
900 PRINT
910 NEXT A
920 PRINT
930 PRINT "DIFFERENT OUTPUT";
940 INPUT Q$
950 IF Q$ = "YES" GOTO 510
960 IF Q$<>"NO" GOTO 580
970 PRINT "DO YOU WANT ANOTHER SET OF NUMBERS";
980 INPUT T$
990 IF T$="YES" GO TO 150
1000 STOP
1010
1020
1030 'GRAPHING ROUTINE
1040 L=R2(1)
1050 H=R2(Y)
1060 I=(H-L)/30
1070 PRINT "GRAPH OF"Y"NUMBERS PRODUCED, FROM"L"TO"H"BY"I
1080 PRINT
1090 PRINT
1100 B=1
1110 FOR A=L TO H BY I
1120 PRINT INT(A),
1130 IF R2(B)<R2(B-1) GOTO 1200
1140 IF R2(B)>A GOTO 1180
1150 PRINT "*";
1160 B=B+1
1170 GOTO 1130
1180 PRINT
1190 NEXT A
1200 PRINT
1210 PRINT
1220 RETURN
1230 'SORTING ROUTINE
1240 F=1
1250 X1=TIM
1260 IF S$="R" GOTO 1500
1270 IF S$="B" GOTO 1320
1280 GOTO 1710
1290
1300
1310 'BUBBLE SORT
1320 PRINT "BUBBLE SORT ALGORITHM:"
1330 N7=C7=0
1340 FOR A=1 TO Y-1
1350 FOR B=A+1 TO Y
1360 C7=C7+1
1370 IF R2(A)<R2(B)GOTO 1420
1380 N7=N7+1
1390 T=R2(A)
1400 R2(A)=R2(B)
1410 R2(B)=T
1420 NEXT B,A
1430 X2=TIM-X1
1440 PRINT X2"SECONDS SORTING TIME."
1450 PRINT N7"SWITCHES EXECUTED."
1460 PRINT C7"COMPARISONS EXECUTED."
1470 RETURN
1480
1490
1500      'DELAYED REPLACEMENT SORT
1510      PRINT "DELAYED REPLACEMENT SORT ALGORITHM:"
1520 N7=C7=0
1530 J7=K7=L7=0
1540 L7=L7+1
1550 IF L7=Y GOTO 1430
1560 J7=L7
1570 K7=J7+1
1580 C7=C7+1
1590 IF R2(K7)>R2(J7) GOTO 1610
1600 J7=K7
1610 K7=K7+1
1620 IF K7<=Y GOTO 1580
1630 IF L7=J7 GOTO 1540
1640 N7=N7+1
1650 T=R2(J7)
1660 R2(J7)=R2(L7)
1670 R2(L7)=T
1680 GOTO 1540
1690
1700
1710 'SHELL - METZNER SORT
1720 PRINT "SHELL - METZNER SORT:"
1730 N7=C7=0
1740 M6=Y
1750 M6=INT(M6/2)
1760 IF M6=0 GOTO 1430
1770 K6=Y-M6
1780 J6=1
1790 I6=J6
1800 L6=I6+M6
1810 C7=C7+1
1820 IF R2(I6)<=R2(L6) GOTO 1890
1830 N7=N7+1
1840 T=R2(I6)
1850 R2(I6)=R2(L6)
1860 R2(L6)=T
1870 I6=I6-M6
1880 IF I6>=1 GOTO 1800
1890 J6=J6+1
1900 IF J6>K6 GOTO 1750
1910 GOTO 1790
1920 END

READY
```

```
THIS PROGRAM PRODUCES A NORMALLY DISTRIBUTED SAMPLE
OF UP TO 2000 POSITIVE INTEGERS ACCORDING TO YOUR DEMANDS.

DO YOU WISH TO TIME SORTING ALGORITHMS ?YES
SELECT SORTING ALGORITHM:

TYPE      TO USE
 B         BUBBLE
 R         DELAYED REPLACEMENT
 S         SHELL - METZNER
 ?R
TYPE THE FOLLOWING:  SAMPLE SIZE, MEAN, STD. DEV.
 ?100,100,15


 100 RANDOM NUMBERS GENERATED IN 0.316 SECONDS.


MEAN = 100.81 , STD. DEV. = 15.5879


WHAT FORM OF OUTPUT DO YOU WANT?

TYPE     IF YOU WANT
 G       HISTOGRAM ON TTY
 T       NUMBERS ON TTY
 F       NUMBERS ON FILE
 TS      NUMBERS ON TTY, SORTED
 FS      NUMBERS ON FILE, SORTED
 ?T
 101  149  98  88  69  81  104  115  131  111
 115  105  101  82  120  113  88  107  108  115
 95  117  80  89  111  97  115  95  105  103
 109  102  84  127  99  113  96  77  98  93
 87  97  74  125  114  135  83  103  90  100
 108  104  93  93  112  106  85  76  112  123
 103  110  73  88  104  91  120  107  133  106
 105  80  106  76  110  82  82  83  92  92
 100  102  121  103  72  101  108  88  114  119
 120  111  71  89  95  106  94  88  90  120

DIFFERENT OUTPUT ?TS
DELAYED REPLACEMENT SORT ALGORITHM:
 0.967 SECONDS SORTING TIME.
 96 SWITCHES EXECUTED.
 4950 COMPARISONS EXECUTED.
 69   71   72   73   74   76   76   77   80   80
 81   82   82   82   83   83   84   85   87   88
 88   88   88   88   89   89   90   90   91   92
 92   93   93   93   94   95   95   95   96   97
 97   98   98   99   100  100  101  101  101  102
 102  103  103  103  103  104  104  104  105  105
 105  106  106  106  106  107  107  108  108  108
 109  110  110  111  111  111  112  112  113  113
 114  114  115  115  115  115  117  119  120  120
 120  120  121  123  125  127  131  133  135  149

DIFFERENT OUTPUT ?G
GRAPH OF 100 NUMBERS PRODUCED, FROM 69 TO 149 BY 2.66667


 69          *
 71          *
 74          ***
 77          ***
 79
 82          ******
 85          ****
 87          *
 90          *********
 93          ******
 95          ****
 98          *****
 101         ******
 103         ******
 106         **********
 109         ******
 111         *****
 114         ******
 117         *****
 119         *
 122         *****
 125         **
 127         *
 130
 133         **
 135         *
 138
 141
 143
 146
```

DAYS AND DATES

James Reagan
Mathematics Teacher
Stevenson High School
Sterling Heights, Michigan

Dates become important and remembered because of their importance. You remember your birthday, that perfect date, a confirmation or bar mitzvah, a marriage, divorce, death, birth, or graduation date of yourself or your love. These are personal. Remember the dates? Sure. Remember the day of the week? No? I didn't think so. But now, to take you back in your memory lane to that fond or dreaded day there is a find-the-day-of-the-week formula known as Zeller's Congruence.

If you don't care about your personal past, how knowledgeable are you about your historical past? Try the quiz to see.

An Illustrative Quiz

Provide the date and the day of the week for each of the following events.

1. The stock market crashes beginning the Great Depression.
2. The Second Continental Congress adopts the Declaration of Independence.
3. Japan attacks Pearl Harbor.
4. President Lincoln is assssinated at Ford's Theater.
5. The bombardment of Fort Sumter begins the Civil War.
6. General Custer makes his "last stand" at Little Big Horn.
7. Russia launches Sputnik I, the first artificial satellite.
8. The United States of America drops an atomic bomb on Hiroshima, Japan.
9. President Kennedy is slain by an assassin's bullet in Dallas, Texas.
10. The oceanliner Lusitania is sunk by German U-boat torpedoes killing 1198 persons including 124 Americans.
11. German armies invade Poland starting World War Two.
12. The United States Supreme Court rules in the case of *Brown v Board of Education* that separate schools based upon skin color are inherently unequal.
13. South Korea is invaded by North Korean troops.
14. Richard M. Nixon resigns as President of the United States of America.
15. D-day. Allied troops land in Normandy, France.

Scoring: Count 1 point each for month, day of the month, and year; count 5 points for correct day of the week. There are a possible 8 points for each event with 120 possible points for the quiz. If you scored 0-10 points you are about average; 11-20 points above average; 21-40 points superior; 41-80 points unbelievable; 81-120 points an historical nut — congratulations!!!

For those of you who need help 1) reread history books for the date and 2) utilize Zeller's Congruence to determine the day of the week for any particular date. The formula is:

$$F = (\lfloor 2.6m - 0.2 \rfloor + k + d + \frac{d}{4} + \frac{c}{4} - 2c) \bmod 7.$$

In this formula, F will have a value 0, 1, 2, 3, 4, 5, or 6; the corresponding day of the week is Sunday, Monday, Tuesday, Wednesday, ..., or Saturday. The modulus 7 can be thought of as the remainder when the value of the parenthetical expression is divided by 7.

The righthand side of the congruence contains the variables described as follows:

- k is the day of the month,
- c is the number of hundreds in the year,
- d is the year in the century, and
- m is the month number, but not the layman's month number. January and February are month numbers 11 and 12 of the preceding year (affecting *d* and possibly *c* described above), March is month number 1, April is month 2, May is 3, ..., and December is month number 10.

The square brackets, ⌊ ⌋, indicate that the "greatest integer value" is to be applied to the included expression. A specific example follows.

Example

The date is October 12, 1956. In layman's terms the date is expressed as 10, 12, 1956. For Zeller's Congruence we use m = 8, k = 12, c = 19, and d = 56.

Substituting these values into the right side of the congruence we have

$$F = (\lfloor 2.6*8 - 0.2 \rfloor + 12 + 56 + \frac{56}{4} + \frac{19}{4} - 2*19) \bmod 7$$

$$= (\lfloor 20.8 - 0.2 \rfloor + 12 + 56 + \lfloor 14 \rfloor + \lfloor 4.75 \rfloor - 38) \bmod 7$$

$$= (20 + 12 + 56 + 14 + 4 - 38) \bmod 7$$

$$= (68) \bmod 7$$

$$= 5 \bmod 7.$$

Thus, we conclude that the day of the week is Friday.

In the application of the formula the following mapping may be a helpful study guide.

LAYMAN'S NOTATION	FORMULA REQUIRES	F VALUE COMPUTED	DAY OF THE WEEK
10-12-1956	8,12,19,56	5	Friday
9-18-1963	7,18,19,63	3	Wednesday
12-25-1972	10,25,19,72	1	Monday
3- 9-1929	1, 9,19,29	6	Saturday
2- 6-1976	12, 6,19,75	5	Friday
1-13-1970	11,13,19,69	2	Tuesday
1- 1-2000	11, 1,19,99	6	Saturday

It might be helpful to understand that the month numbers for the application of the congruence begin with March = 1 and continue to the following February = 12; in this way any leap year day is placed at the end of the formula year.

The First Problem

Write a program that will accept any date in layman's terms and print the corresponding day of the week. The program must provide the translation for the application of the variables used in the congruence. For example, if one types 1,13,1974 the program must translate these values to 11,13,19,73 for the corresponding values of *m, k, c* and *d,* respectively. Using this program you may verify the days of the week for the dates of the Illustrative Quiz.

The Second Problem

Superstitions have developed over the history of man. Many people are superstitious of certain events; those who are not superstitious have some knowledge of the superstitions. Some of the events associated with "bad luck" are: walking under a ladder, having a black cat cross one's path, and breaking a mirror. Perhaps the most well known of all superstitions involves "Black Friday," the description of Friday the Thirteenth.

This year, 1976, has two Friday the Thirteenths; one occurred in February and the other in August. This may be verified by a search of the calendar or by observation of a perpetual calendar.

The second problem becomes one of modifying the program produced to solve the first problem: produce a list of Friday the Thirteenths over a given interval of years. For example, produce a list of Friday the Thirteenths for the years from 1977 to 1980.

The Third Problem

This third problem might be investigated using the computer program produced for the second problem. However, there is also a rigorous mathematical proof of the conjectures motivated by the computer investigation.

The problem is stated in the form of two questions:

1. What is the most number of Friday the Thirteenths in any given year?
2. Is there any year that does not have at least one Friday the Thirteenth?

The Fourth Problem

Some workers are paid bi-weekly, that is they are paid every-other week. The traditional payday is Friday. In a given year there are some months that have 5 Fridays; two of these months occur so that there are 3 paydays in that month, one on each of the first, third and fifth Fridays. The month of February has 4 of each day of the week except in years that are leap years; then one day occurs five times. If that day that occurs five times is Friday, there is a possibility that three paydays may occur in that month.

In what years will February have five Fridays? How often does this occur? If one has bi-weekly pay-periods and one of them does occur on the first Friday of a leap year February beginning on Friday, will the same situation occur again in the worker's lifetime?

Answers To Illustrative Quiz

1. October 24, 1929: Thursday
2. July 4, 1776: Sunday
3. December 7, 1941: Sunday
4. April 14, 1865: Tuesday
5. April 12, 1861: Friday
6. June 25, 1876: Sunday
7. October 4, 1957: Friday
8. August 6, 1945: Monday
9. November 22, 1963: Friday
10. May 7, 1915: Friday
11. September 1, 1939: Friday
12. May 17, 1954: Monday
13. June 25, 1950: Monday
14. August 9, 1974: Friday
15. June 6, 1944: Tuesday

ANSWERS TO COMPUTER LITERACY QUIZ

1. F	7. F	13. 4	19. F	25. F
2. F	8. F	14. T	20. T	26. T
3. 1	9. T	15. T	21. F	27. T
4. T	10. 1	16. 2	22. F	28. 4
5. 5	11. T	17. 3	23. F	29. 4
6. 2	12. 3	18. T	24. 4	30. F

CREATIVE PROGRAMMING TECHNIQUES....

In this regular column, ***Creative Computing*** *will publish original programming techniques, hints, and tricks. We're not looking for material from textbooks, but we are seeking material from readers that has proved helpful and effective. Send contributions to Editor,* ***Creative Computing.*** *The techniques presented below are from the Advisory Unit for Computer Based Education, Hertfordshire County Council, England.*

CONDITIONAL STATEMENTS

It is probably a fair generalisation to say that the comparison statement in any high level language is one to be avoided wherever possible. It usually translates into a sequence of arithmetic operations followed by a comparison with zero, which is usually the basic comparison operation available in the machine code.

As, contrariwise, the comparison instruction is fundamental to any serious programming, it is natural to look at some ways of using it as little as possible. We offer some general hints:

In a language which allows compound 'logical' statements, such as FORTRAN, it is often wise to re-write such a statement using simpler statements, particularly where "OR" is being used. The following is probably transparent to readers without any knowledge of FORTRAN.

Instead of:

```
IF (COST.EQ.0.0 .OR. TIME.LT.9.5 .OR.
   DAYS.GT.5.0) GO TO 9995
```

it is better to write:

```
IF (COST.EQ.0.0) GO TO 9995
IF (TIME.LT.9.5) GO TO 9995
IF (DAYS.GT.5.0) GO TO 9995
```

and best to order these three conditional statements so that the one most likely to be satisfied is tried first. The philosophy behind this is clear; if the COST is zero, we do not need to try the other two conditionals, but it is often the case with some systems that the full logical value of the compound conditional statement will be computed before branching.

THERE'S ALWAYS
AN EASY SOLUTION
TO EVERY PROBLEM

NEAT ... PLAUSIBLE ... AND WRONG

117

SEARCHING A LIST

Another interesting technique is concerned with searching an unordered list for a particular entry. Imagine we have a list of names:

GILLIAN, JOSEPHINE, CINDY, ANGELA, DAPHNE,......, JEAN

stored in the one-dimensional array A$(1), A$(2),......, A$(n), and we wish to find out whether "PHOEBE" is one of the names on this list. An 'obvious' approach is contained in this program fragment:

```
300 LET I =O
310 LET I = I+1
320 IF I> N THEN 500
330 IF A$ (I) = "PHOEBE" THEN 400
340 GO TO 310
•
•
400 PRINT "PHOEBE FOUND AS THE
";I;"-TH NAME ON THE LIST"
•
•
500 PRINT "NAME NOT FOUND ON LIST"
```

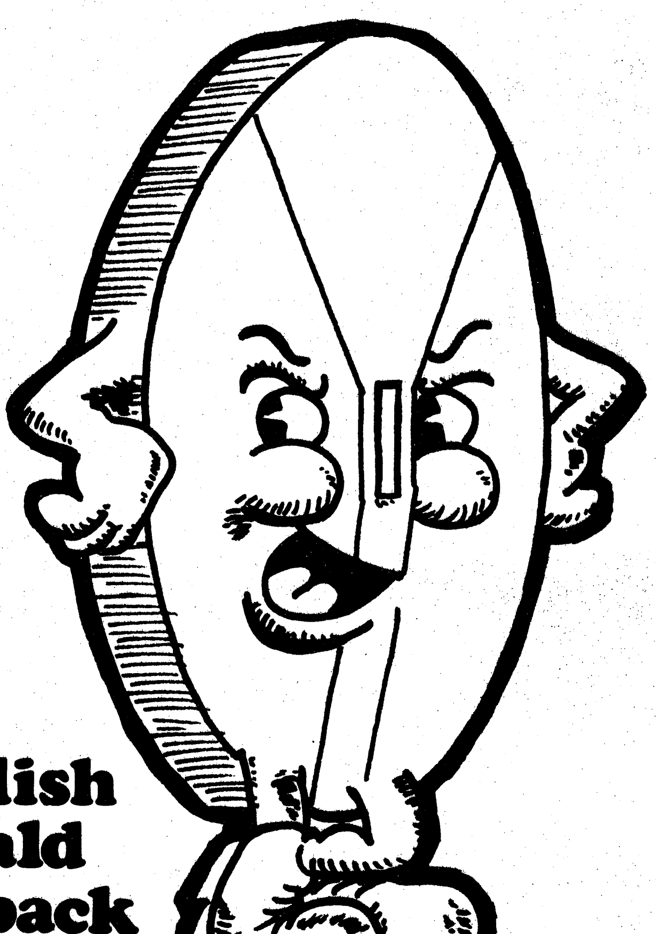
The comparison at line 320 is necessary because we must detect the end of the list in the event of not finding "PHOEBE".

However, if we adjust the list, in order to ensure that "PHOEBE" is always found, by adding it as the N+1 th element, we can avoid this comparison. The program then becomes:

```
300 LET A$(N+1) = "PHOEBE"
310 LET I = O
320 LET I = I+1
330 IF A$ (I) = "PHOEBE" THEN 400
340 GOTO 320
400 IF I = N + 1 THEN 500
410 PRINT " PHOEBE FOUND........&c"
•
•
500 PRINT "NOT FOUND &c...."
```

Now the comparison at line 400 will be executed once instead of N+1 times. This might not be a saving, however. As in all 'short-cuts' of this nature, it may happen that the overheads of time or core-space used by arranging the short-cut are more expensive than the saving. In general, the longer the list, the better the saving.

Computer Games



Devilish
Donald
Diskpack

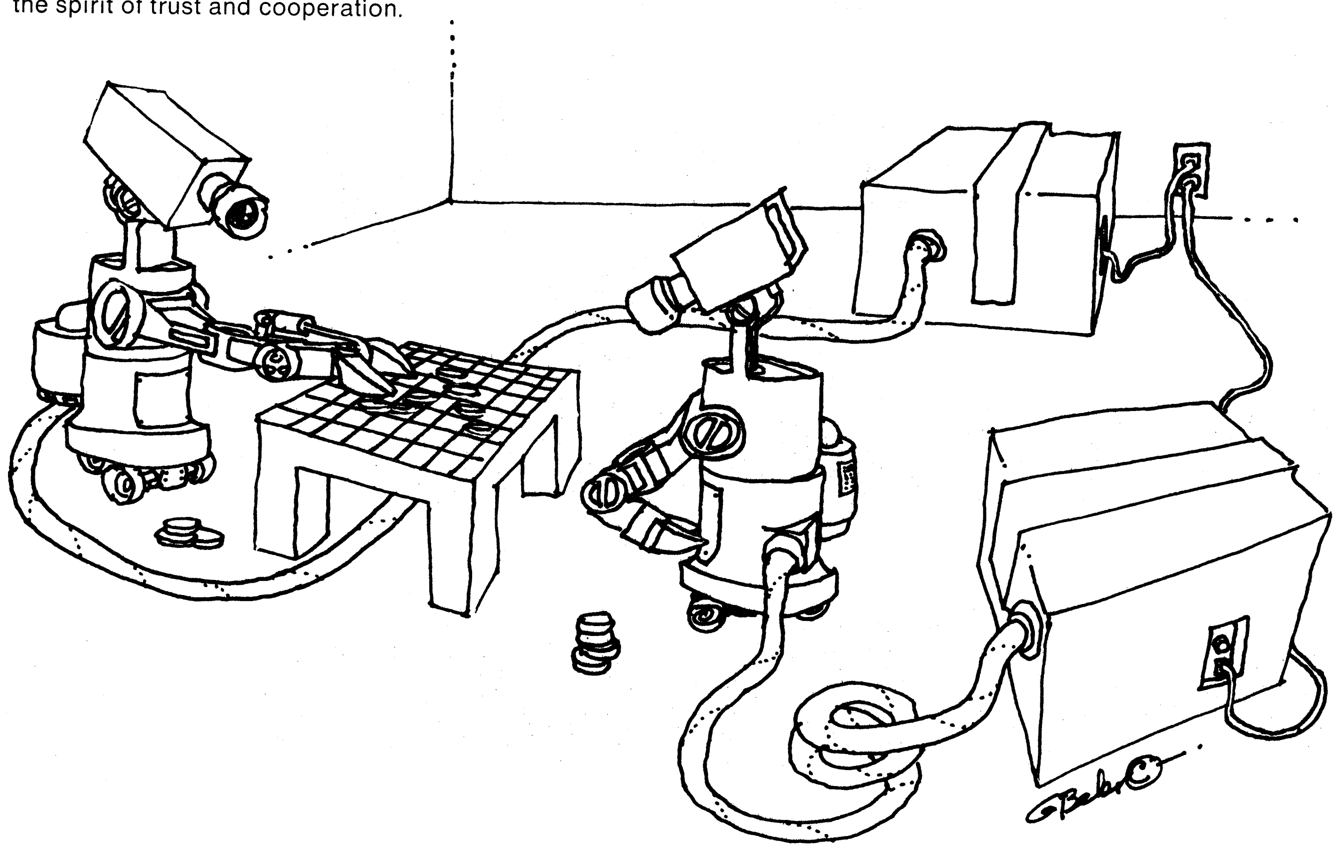
Learning With Computer Games

Brief History of Sports and Games
Sports and games have been with us almost since the dawn of man. The ancient Chinese and Egyptians devised an astonishing array of mathematical and logical games (Nim, Towers of Hanoi, Awari or Kalah, etc.). The Greeks originated a great many physical games in their Olympic competitions. Medieval Europe was responsible for many games played for recreation in nobles' courts (Chess, Hi-Q, etc.). Team games are a phenomenon of the last few centuries, with many originating in England and the United States.

Throughout history, the common thread running through all sports and games is that their intended function has been purely recreational. Games were viewed as a diversion from the realities of life. Secondly, sports and games were and are a leisure-time activity. In early Europe, only the nobles, who had leisure time, played games. Even today, except for professional athletes, sports and games are generally considered outside, extra-curricular activities.

Nevertheless, sports and games have generally had some redeeming virtues. Merely serving as a break from the realities of life is probably enough to assure games a place in history. But there is more. Chinese philosophers spoke of games as sharpening one's wit. Sports build the body just as card, board and mathematical games build the mind. Team games heighten the spirit of trust and cooperation.

Games as an Educational Tool
Not until the last 10 or 15 years of educational innovation, however, have games ever been used primarily as an educational tool where learning is the primary purpose of playing. Today, there is a growing body of evidence which indicates that a combination of learning games and student teams may well be one of the most effective approaches to learning ever devised. In a study by Edwards, DeVries, and Snyder (1972), the games-teams combination in seventh grade classes resulted in a significant increase on several dimensions of mathematics achievement. A follow-up study in 1973 found that both games and teams represent useful techniques for restructuring classroom processes. Their effects are complementary in that they create very different classroom experiences for students. Games cause students to translate their increased interpersonal interaction into increased informal peer-tutoring on the subject material at hand. They are also likely to view their class in a much more positive light. The addition of teams to a traditional classroom meets a rather different set of objectives. Attending the class and encountering the subject matter is not necessarily made more fun as it is with games. However, students work together on traditional classroom tasks to a much greater degree, resulting in an increased level of mutual concern among the students.



Other studies echo the results above. The learning effectiveness of games has been frequently cited (Allen, Allen, and Ross, 1970; Boocock and Schild, 1968; Fletcher, 1971). Learning games generally create an intense and often enjoyable interpersonal experience. This is due in part to the interdependent task structure which requires interaction among the players (Inbar, 1968).

Teams have been in use longer as an educational technique. Generally it has been found that students in a team outperform students working as individuals. A key reason for the effect of teams has been the high rate of peer tutoring (Wodarski, Hamblin, Buckholdt, and Ferritor, 1971).

As mentioned earlier, because games and teams positively affect different classroom processes, combining the two techniques creates an even more powerful effect on the learning environment. This in turn is translated into greater academic growth as well as increased trust and cooperation.

Computer Games and Simulations

People have written games for the computer almost since its birth. Many tended to be copies of real life sports and games (football, poker, tic-tac-toe, etc.). Other games were written as simulations of other real processes with a recreational quality (lunar landing, parachute jump, artillery weapon firing, etc.). Still other computer games were devised mainly for their recreational and mind-challenging value (game of life, bulls and cows, bagels, mugwump, etc.).

The educational value of both playing and writing computer games is substantial. They make ideal supplemental learning experiences when teaching about probability, statistics, logic, problem solving, decision making, and value clarification.

David H. Ahl

BIBLIOGRAPHY

Allen, L. E., Allen, R. W., & Ross, J. "The Virtues of Nonsimulation Games." *Simulation and Games,* 1970, *1,* 319-326.

Boocock, S. S. & Schild, E. O. (Eds.) *Simulation Games in Learning.* Beverly Hills, CA: Sage, 1968.

DeVries, D. S. & Edwards, K. J. "Learning Games and Student Teams: Their Effects on Classroom Process." *American Education Research Journal,* 1973, *10,* 307-318.

Edwards, K. J., DeVries, D. L., & Snyder, J. P. "Games and Teams: A Winning Combination." *Simulation and Games,* 1972, *3,* 247-269.

Fletcher, J. L. "The Effectiveness of Simulation Games as Learning Environments." *Simulation and Games,* 1971, *2,* 425-454.

Inbar, M. "Individual and Group Effects on Enjoyment and Learning in a Game Simulating a Community Disaster." In S. S. Boocock & E. O. Schild (Eds.), *Simulation Games in Learning,* Beverly Hills, CA: Sage, 1968, Pp. 169-190.

Wodarski, J. S., Hamblin, R. L., Buckholdt, D. & Ferritor, D. "Effects of Individual and Group Contingencies On Arithmetic Performance." Paper presented at the meeting of the American Educational Association, New York, January, 1971.

DEFINITIONS

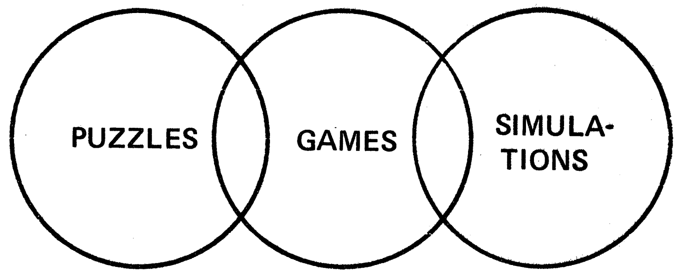
In the world of computer education and recreation a number of terms are used which are sometimes confused with one another.

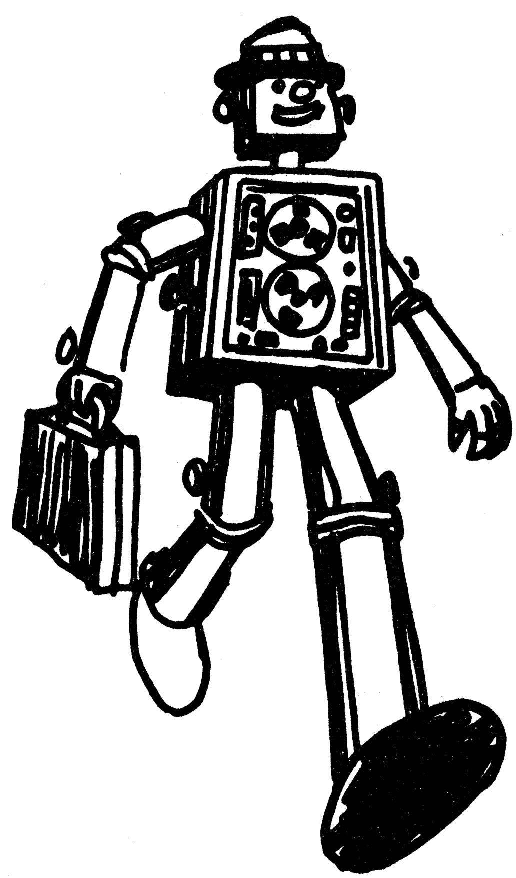
A *Puzzle* is a problem that has a baffling quality or great intricacy, that one must exercise substantial mental ingenuity and thought to solve.

A *Game* is an amusement or sport involving competition under a specific set of rules. The competition may be against oneself or against laws of chance such as in solitaire or roulette.

A *Simulation* is a model or representation of a real-life process, situation, or event, frequently on a computer.

One might think of puzzles, games and simulations as having a certain overlap as follows:





Wumpus 2 by Gregory Yob

Hark!! The weary Wumpus hunter, wan from 50 days in the Terminal Caverns, exhausted and with all of his arrows expended — — — (A groaning Teletype roars at a sleepy student. Maps litter the floor covered with circles and integers. With callused fingers, the immortal Wumpus player looks up with bloodshot eyes and implores: "How do I get out of here?")

I suspected that the dodecahedron may prove a bit boring after a few thousand games, so I wrote Wumpus 2 to extend your pleasure. Some of the more mathematical minded may have noticed there are lots of ways to link caves with three tunnels apiece. Some of these patterns are topologically interesting . . . .

Wumpus 2 is the same old Wumpus* in different settings — including those of your own design. As you play in the different caves, you will notice that the game changes in difficulty and strategy. Now to a description of the various caves in Wumpus 2.

CAVE 0 (Dodecahedron)
This is the same old Wumpus with which you are familiar.

CAVE 1 (Mobius Strip)
Since my original vision was topological, here is the first wonder of topology, the Mobius strip. Take a strip of paper, give it a half-twist and join the ends into a loop. The result has just one side and one edge (if you disbelieve, take a pencil and go around the thing).

A perceptive player will note that the placement of the pits influence the game. Two pits placed just right (around 5% of the games have this) will force a detour back around the strip in certain cases. Getting around is slower than in Cave 0, but it is easier to search the place.

CAVE 2 (String of Beads)
See the diagram for this one. Here, placement of the pits will often make parts of the caves inaccessible except by bat-express. (Can you see why?) Play in this cave is frustrating until you have gone to the trouble of making a reference map; otherwise you keep coming back to your starting point. (Look at the diagram and see how this may be so.)

CAVE 3 (Hex Network)
This is my attempt at a torus (doughnut). If you can visualise a hexagon net like a honeycomb or a tile floor and stretch it onto a doughnut, you've got it!! The drawing tries to show this, but if you prefer, think of it as a complicated molecule of some sort. Play in this one is very similar to CAVE 0.

CAVE 4 (Dendrite)
Up to now, each tunnel leads to another cave and only one tunnel connects a pair of caves. This need not be a strict rule and the next two caves illustrate variations on this. The dendrite is a branching pattern like a tree or a plant. At the ends of the plant are "leaves" which are caves leading to themselves or multiple tunnels. This cave is especially susceptible to severance by pits and getting stuck in corners near the wumpus. A nice thing is that you often will know exactly where the Wumpus is when you come near him.

CAVE 5 (one way streets)
This is the extreme example of all tunnels are one-way. You will find that getting about this cave is like travel in Los Angeles — much going to get to the neighbor's house. If you overshoot, you must travel all the way around, just like missing a freeway offramp.

CAVE 6 (Do Your OWN)
Draw up a map of caves, each cave with tunnels GOING TO three (exactly three) caves (same or different). Then the computer will ask you for the numbers of the destination tunnels for each of the 20 caves in Wumpus. When you have it entered, play Wumpus on your own caves. Let me know of your favorite ones, and your most frustrating ones!!!

If you are a programming fiend, how would you arrange a Wumpus game with lots of caves and tunnels — and legitimate moves may include changing how the caves connect to each other?? Send me any versions which may happen.

WUMPUS 3

Around the PCC center, a Howard lives. He enjoys suggesting "improvements" to my games and I enjoy telling him how useless his ideas are. One day, he got around me and came up with Wumpus 3, with earthquakes, bat migrations and the incredible "tumareo" (don't let your fantasy get away from you now!!).

Have a look at the run and see if you like it. My personal opinion is that the changes and reshufflings happen too often for comfort — what do you think?

We didn't have room for Wumpus 3. Send to Gregory for a tape.

FINIS
In any case, Wumpus has spawned several versions and spread about the computer games-dom really nicely. For myself, the soul of the game is in the idea and fun of it rather than the program or the computer which hosts it. I feel that all really good games will turn programmers on enough for them to write it for their system from the idea alone and encourage games writers to think carefully on the art and esthetics of their games before writing a line of code.

WUMPUS TAPES, ETC.
I can be found at:

Gregory Yob
PO Box 354
Palo Alto, Calif. 94301

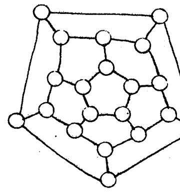
Paper tapes of Wumpus, Wumpus 2 and Wumpus 3 are available and cost $5.00 each.

May your arrows remain straight. —Gregory Yob.

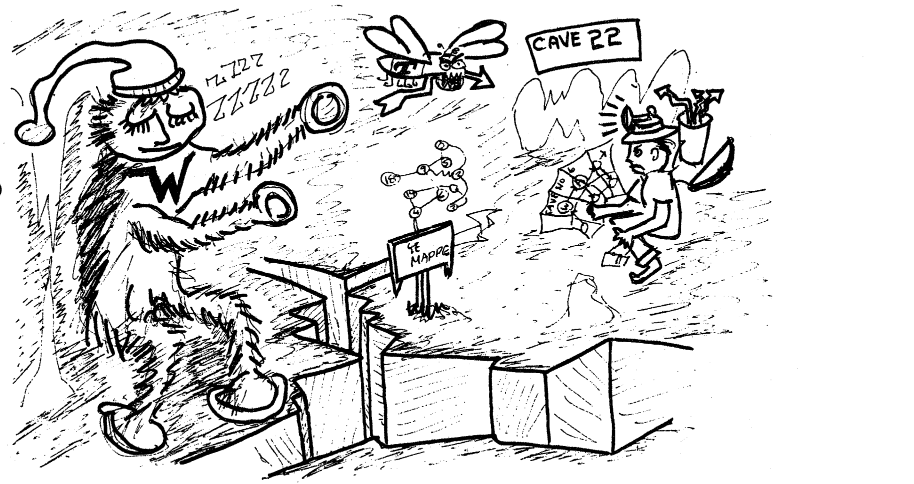
Sample run, listings, and cave diagrams on the next 2 pages.

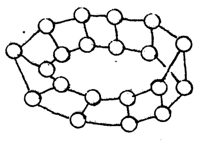
* Wumpus 1 appeared in <u>Creative Computing</u>, Vol 1, No 5 (Sep-Oct '75).

THE CAVES OF WUMPUS 2

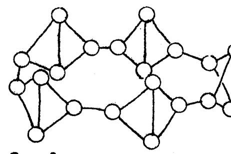


Cave 0
(the usual)

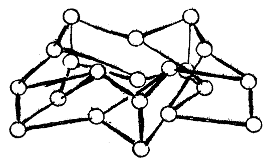




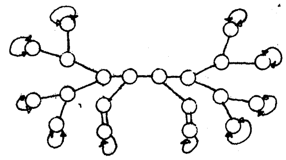
Cave I
(Mobius Strip)



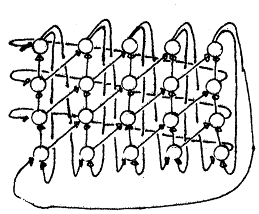
Cave 2
(String of Beads)



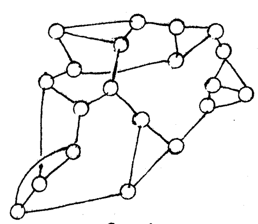
Cave 3
(Toroidal Hex Net)



Cave 4
(Dendrite)



Cave 5
(One Way Only)



Cave 6
(Anything You Like)

WUMPUS 2 SAMPLE RUN

We didn't print the instructions on the run. They are in the listing in Statements 700–1210.

```
HUNT THE WUMPUS

I SMELL A WUMPUS!
I FEEL A DRAFT
BATS NEARBY!
YOU ARE IN ROOM  11     TUNNELS LEAD TO  9      10      12

SHOOT OR MOVE? ?S

NO. OF ROOMS (1-5)? ?3

ROOM #?9

ROOM #?10

ROOM #?11


OUCH! ARROW GOT YOU!
HA HA HA - YOU LOSE!
PLAY AGAIN? ?Y

SAME SET-UP? ?Y

HUNT THE WUMPUS

I SMELL A WUMPUS!
I FEEL A DRAFT
BATS NEARBY!
YOU ARE IN ROOM  11     TUNNELS LEAD TO  9      10      12

SHOOT OR MOVE? ?M

WHERE TO? ?9


ZAP--SUPER BAT SNATCH! ELSEWHEREVILLE FOR YOU!
... OOPS! BUMPED A WUMPUS!

I SMELL A WUMPUS!
I FEEL A DRAFT
YOU ARE IN ROOM  12     TUNNELS LEAD TO  10     11      13

SHOOT OR MOVE? ?M

WHERE TO? ?13


YOU ARE IN ROOM  13     TUNNELS LEAD TO  12     14      15

SHOOT OR MOVE? ?S

NO. OF ROOMS (1-5)? ?2

ROOM #?12

ROOM #?11


AHA! YOU GOT THE WUMPUS!  HE WAS IN ROOM 11
HEE HDE HEE - THE WUMPUS'LL GETCHA NEXT TIME!!
```

WUMPUS 2 LISTING — PART 1

```
0001  REM: WUMPUS II BY GREGORY YOB REV MAY 1975
0002  REM: GREGORY YOB PO BOX 354 PALO ALTO, CALIF 94301
0003  REM: PHONE (415) 326-4039
0004  REM: MODIFIED FROM HP BASIC VERSION. NOTE THAT:
0005  REM: THE EXPRESSION I$(1,1) IS REPLACED BY I$(1:1)
0006  REM: IN THIS VERSION.
0010  DIM I$(6)
0020  REM- WUMPUS VERSION II
0050  PRINT "INSTRUCTIONS? ";
0060  INPUT I$
0070  PRINT
0090  IF I$(1:1) <> "Y" THEN 130
0100  GOSUB 700
0110  REM-CHOOSE & SET UP CAVE
0120  DIM S(20,3)
0130  GOSUB 2530
0140  DEF FNA(X)=INT(20*RND(0))+1
0150  DEF FNB(X)=INT(3*RND(0))+1
0160  DEF FNC(X)=INT(4*RND(0))+1
0170  REM-LOCATE L ARRAY ITEMS
0180  REM-1-YOU,2-WUMPUS,3&4-PITS,5&6-BATS
0190  DIM L(6)
0200  DIM M(6)
0210   FOR J=1 TO 6
0220   L(J)=FNA(0)
0230   M(J)=L(J)
0240   NEXT J
0250  REM-CHECK FOR CROSSOVERS (IE L(1)=L(2),ETC)
0260   FOR J=1 TO 6
0270    FOR K=J TO 6
0280    IF J=K THEN 300
0290    IF L(J)=L(K) THEN 210
0300    NEXT K
0310   NEXT J
0320  REM-SET# ARROWS
0330  A=5
0340  L=L(1)
0350  REM-RUN THE GAME
0360  PRINT "HUNT THE WUMPUS"
0370  REM-HAZARD WARNINGS & LOCATION
0380  GOSUB 1230
0390  REM-MOVE OR SHOOT
0400  GOSUB 1400
0410  GOTO 0 OF 430,470
0420  REM-SHOOT
0430  GOSUB 1550
0440  IF F=0 THEN 400
0450  GOTO 490
0460  REM-MOVE
0470  GOSUB 2150
0480  IF F=0 THEN 380
0490  IF F>0 THEN 540
0500  REM-LOSE
0510  PRINT "HA HA HA - YOU LOSE!"
0520  GOTO 550
0530  REM-WIN
0540  PRINT "HEE HDE HEE - THE WUMPUS'LL GETCHA NEXT TIME!!"
0550   FOR J=1 TO 6
0560   L(J)=M(J)
0570   NEXT J
0580  PRINT "PLAY AGAIN? ";
0590  INPUT I$
0600  PRINT
0620  IF I$(1:1) <> "Y" THEN 3310
0640  PRINT "SAME SET-UP? ";
0650  INPUT I$
0660  PRINT
0680  IF I$(1:1) <> "Y" THEN 130
0690  GOTO 330
0700  REM-INSTRUCTIONS
0710  PRINT "WELCOME TO WUMPUS II"
0720  PRINT "THIS VERSION HAS THE SAME RULES AS 'HUNT THE WUMPUS'."
0730  PRINT "HOWEVER, YOU NOW HAVE A CHOICE OF CAVES TO PLAY IN."
0740  PRINT "SOME CAVES ARE EASIER THAN OTHERS. ALL CAVES HAVE 20"
0750  PRINT "ROOMS AND 3 TUNNELS LEADING FROM ONE ROOM TO OTHER ROOMS."
0760  PRINT "THE CAVES ARE:"
0770  PRINT "  0  -  DODECAHEDRON      THE ROOMS OF THIS CAVE ARE ON A"
0780  PRINT "           12-SIDED OBJECT, EACH SIDE FORMING A PENTAGON."
0790  PRINT "           THE ROOMS ARE AT THE CORNERS OF THE PENTAGONS,"
0800  PRINT "           EACH ROOM HAVING TUNNELS LEADING TO 3 OTHER ROOMS.
"
0810  PRINT "  1  -  MOBIUS STRIP      THIS CAVE IS TWO ROOMS"
0820  PRINT "           WIDE AND 10 ROOMS AROUND (LIKE A BELT)"
0830  PRINT "           YOU WILL NOTICE THERE IS A HALF-TWIST"
0840  PRINT "           SOMEWHERE."
0850  PRINT
```

WUMPUS 2 LISTING
PART 2

```
0860 PRINT "  2  -  STRING OF BEADS    FIVE BEADS IN A CIRCLE."
0870 PRINT "          EACH BEAD IS A DIAMOND WITH A VERTICAL"
0880 PRINT "          CROSS-BAR.  THE RIGHT & LEFT CORNERS LEAD"
0890 PRINT "          TO NEIGHBORING BEADS.(THIS ONE IS DIFFICULT"
0900 PRINT "          TO PLAY)"
0910 PRINT
0920 PRINT "  3  -  HEX NETWORK    IMAGINE A HEX TILE FLOOR. TAKE"
0930 PRINT "         A RECTANGLE WITH 20 POINTS (INTERSECTIONS)"
0940 PRINT "         INSIDE (4X4). JOIN RIGHT & LEFT SIDES TO MAKE A"
0950 PRINT "         CYLINDER. THEN JOIN TOP & BOTTOM TO FORM A"
0960 PRINT "         TORUS (DOUGHNUT)."
0970 PRINT "         HAVE FUN IMAGINING THIS ONE!!"
0980 PRINT
0990 PRINT "  CAVES 1-3 ARE REGULAR IN THE SENSE THAT EACH ROOM"
1000 PRINT "  GOES TO THREE OTHEER ROOMS & TUNNELS ALLOW TWO-"
1010 PRINT "  WAY TRAFFIC.  HERE ARE SOME 'IRREGULAR' CAVES:"
1020 PRINT
1030 PRINT "  4  -  DENDRITE WITH DEGENERACIES   PULL A PLANT FROM"
1040 PRINT "          THE GROUND.  THE ROOTS & BRANCHES FORM A"
1050 PRINT "          DENTRITE - IE, THERE ARE NO LOOPING PATHS"
1060 PRINT "          DEGENERACY MEANS  A) SOME ROOMS CONNECT TO"
1070 PRINT "          THEMSELVES AND B) SOME ROOMS HAVE MORE THAN"
1080 PRINT "          ONE TUNNEL TO THE SAME OTHER ROOM IE, 12 HAS"
1090 PRINT "          TWO TUNNELS TO 13."
1100 PRINT
1110 PRINT "  5  -  ONE WAY LATTICE     HERE ALL TUNNELS GO ONE"
1120 PRINT "         WAY ONLY.  TO RETURN, YOU MUST GO AROUND THE"
1130 PRINT "         CAVE (ABOUT 5 MOVES)."
1140 PRINT
1150 PRINT "  6  -  ENTER YOUR OWN CAVE     THE COMPUTER WILL ASK"
1160 PRINT "         YOU THE ROOMS NEXT TO EACH ROOM IN THE CAVE."
1170 PRINT "         FOR EXAMPLE:"
1180 PRINT "           ROOM # 1     ?2,3,4     - YOUR REPLY OF 2,3,4"
1190 PRINT "              MEANS ROOM 1 HAS TUNNELS GOING TO ROOMS:"
1200 PRINT "              2, 3 & 4."
1210 PRINT "  HAPPY HUNTING!"
1220 RETURN
1230 REM-PRINT LOCATION & HAZARD WARNINGS
1240 PRINT
1250  FOR J=2 TO 6
1260   FOR K=1 TO 3
1270   IF S(L(1),K)#L(J) THEN 1340
1280   GOTO J-1 OF 1290,1310,1310,1330,1330
1290   PRINT "I SMELL A WUMPUS!"
1300   GOTO 1340
1310   PRINT "I FEEL A DRAFT"
1320   GOTO 1340
1330   PRINT "BATS NEARBY!"
1340   NEXT K
1350  NEXT J
1360 PRINT "YOU ARE IN ROOM ";L(1);
1370 PRINT " TUNNELS LEAD TO ";S(L,1);S(L,2);S(L,3)
1380 PRINT
1390 RETURN
1400 REM-CHOOSE OPTION
1410 GOTO 1450
1420 PRINT "ERROR   ";
1430 INPUT Z9
1440 PRINT "";
1450 PRINT "SHOOT OR MOVE? ";
1460 INPUT I$
1470 PRINT
1490 IF I$(1:1) <> "S" THEN 1520
1500 O=1
1510 RETURN
1520 IF I$(1:1) <> "M" THEN 1420
1530 O=2
1540 RETURN
1550 REM-ARROW ROUTINE
1560 F=0
1570 REM-PATH OF ARROW
1580 DIM P(5)
1590 GOTO 1630
1600 PRINT "ERROR    ";
1610 INPUT Z9
1620 PRINT "";
1630 PRINT "NO. OF ROOMS (1-5)? ";
1640 INPUT J9
1650 PRINT
1670 IF J9<1 OR J9>5 OR INT(J9) <> ABS(J9) THEN 1600
1680  FOR K=1 TO J9
1690  PRINT "ROOM #";
1700  INPUT P(K)
1710  PRINT
1730  IF P(K)>0 AND P(K)<21 AND INT(P(K))=ABS(P(K)) THEN 1780
1740  PRINT "ERROR   ";
1750  INPUT Z9
1760  PRINT "";
1770  GOTO 1690
1780  NEXT K
1790 PRINT
1800 REM-SHOOT ARROW
1810 A=A-1
1820 A9=L(1)
1830  FOR K=1 TO J9
1840   FOR K1=1 TO 3
1850   IF S(A9,K1)=P(K) THEN 1990
1860   NEXT K1
1870  REM-NO TUNNEL FOR THE ARROW
1880  A9=S(A9,FNB(1))
1890  GOTO 2000
1900  NEXT K
1910 PRINT "MISSED"
1920 REM-MOVE WUMPUS
1930 GOSUB 2070
1940 REM-AMMO CHECK
1950 IF A>0 THEN 1970
1960 F=-1
1970 RETURN
1980 REM-SEE IF ARROW IS AT L[1] OR L[2]
1990 A9=P(K)
2000 IF A9 <> L(2) THEN 2040
2010 PRINT "AHA! YOU GOT THE WUMPUS!  HE WAS IN ROOM";L(2)
2020 F=1
2030 RETURN
2040 IF A9 <> L(1) THEN 1900
2050 PRINT "OUCH! ARROW GOT YOU!"
2060 GOTO 1960
2070 REM-MOVE WUMPUS ROUTINE
2080 K=FNC(0)
2090 IF K=4 THEN 2140
```

```
2100 L(2)=S(L(2),K)
2110 IF L(2)#L THEN 2140
2120 PRINT "TSK TSK TSK- WUMPUS GOT YOU!"
2130 F=-1
2140 RETURN
2150 REM- MOVE ROUTINE
2160 F=0
2170 GOTO 2210
2180 PRINT "ERROR    ";
2190 INPUT Z9
2200 PRINT "";
2210 PRINT "WHERE TO? ";
2220 INPUT L
2230 PRINT
2240 IF L<1 OR L>20 OR INT(L) <> ABS(L) THEN 2180
2250  FOR K=1 TO 3
2260  REM-CHECK IF LEGAL MOVE
2270  IF S(L(1),K)=L THEN 2350
2280  NEXT K
2290 IF L=L(1) THEN 2350
2300 PRINT "NOT POSSIBLE - ";
2310 INPUT Z9
2320 PRINT "";
2330 GOTO 2210
2340 REM-CHECK FOR HAZARDS
2350 L(1)=L
2360 REM-WUMPUS
2370 IF L#L(2) THEN 2430
2380 PRINT "... OOPS! BUMPED A WUMPUS!"
2390 REM-MOVE WUMPUS
2400 GOSUB 2080
2410 IF F=0 THEN 2430
2420 REM-PIT
2430 IF L <> L(3) AND L <> L(4) THEN 2480
2440 PRINT "YYYIIIIEEEE . . . FELL IN A PIT"
2450 F=-1
2460 RETURN
2470 REM-BATS
2480 IF L <> L(5) AND L <> L(6) THEN 2520
2490 PRINT "ZAP--SUPER BAT SNATCH! ELSEWHEREVILLE FOR YOU!"
2500 L=FNA(1)
2510 GOTO 2350
2520 RETURN
2530 REM-SELECT CAVE
2540 GOTO 2580
2550 PRINT "ERROR    ";
2560 INPUT Z9
2570 PRINT "";
2580 PRINT "CAVE #(0-6)? ";
2590 INPUT N
2600 PRINT
2620 IF N<0 OR N>6 OR INT(N) <> ABS(N) THEN 2550
2630 GOSUB N+1 OF 2650,2730,2810,2890,2970,3050,3130
2640 RETURN
2650 REM-DODECAHEDRON
2660 RESTORE 2590
2670 DATA 2,5,8,1,3,10,2,4,12,3,5,14,1,4,6
2680 DATA 5,7,15,6,8,17,1,7,9,8,10,18,2,9,11
2690 DATA 10,12,19,3,11,13,12,14,20,4,13,15,6,14,16
2700 DATA 15,17,20,7,16,18,9,17,19,11,18,20,13,16,19
2710 GOSUB 3240
2720 RETURN
2730 REM-MOBIUS STRIP
2740 RESTORE 2750
2750 DATA 20,2,3,19,1,4,1,4,5,2,3,6,3,6,7
2760 DATA 4,5,8,5,8,9,6,7,10,7,10,11,8,9,12
2770 DATA 9,12,13,10,11,14,11,14,15,12,13,16,12,16,17
2780 DATA 14,15,18,15,18,19,16,17,20,2,17,20,1,18,19
2790 GOSUB 3240
2800 RETURN
2810 REM-STRING OF BEADS
2820 RESTORE 2830
2830 DATA 2,3,20,1,3,4,1,2,4,2,3,5,4,6,7
2840 DATA 5,7,8,5,6,8,6,7,9,8,10,11,9,11,12
2850 DATA 9,10,12,10,11,13,12,14,15,13,15,16,13,14,16
2860 DATA 14,15,17,16,18,19,17,19,20,17,18,20,1,18,19
2870 GOSUB 3240
2880 RETURN
2890 REM-HEX NET ON TORUS
2900 RESTORE 2910
2910 DATA 6,10,16,6,7,17,7,8,18,8,9,19,9,10,20
2920 DATA 1,2,15,2,3,11,3,4,12,4,5,13,5,6,14
2930 DATA 7,16,20,8,16,17,9,17,18,10,18,19,6,19,20
2940 DATA 1,11,12,2,12,13,3,13,14,4,1415,5,11,15
2950 GOSUB 3240                          4,14,15,5,11,15 <BUG!!
2960 RETURN                                    ↖This is the culprit
2970 REM- DENDRITE W/ DEGENERACIES
2980 RESTORE 2990
2990 DATA 1,1,5,2,2,5,3,3,6,4,4,6,1,2,7
3000 DATA 3,4,7,5,6,10,8,9,9,8,8,10,7,9,11
3010 DATA 10,13,14,12,13,13,11,12,12,11,15,16,14,17,18
3020 DATA 14,19,20,15,17,17,15,18,18,16,19,19,16,20,20
3030 GOSUB 3240
3040 RETURN
3050 REM-ONE WAY LATTICE
3060 RESTORE 3070
3070 DATA 5,4,8,1,5,6,2,6,7,3,7,8,8,9,12
3080 DATA 5,9,10,6,10,11,7,11,12,12,13,16,9,13,14
3090 DATA 10,14,15,11,15,16,16,17,20,13,17,18,14,18,19
3100 DATA 15,19,20,1,4,20,1,2,17,2,3,18,3,4,19
3110 GOSUB 3240
3120 RETURN
3130 REM- INPUT OWN CAVE
3140  FOR J=1 TO 20
3150  PRINT "ROOM #";J;
3160  INPUT S(J,1),S(J,2),S(J,3)
3170   FOR K=1 TO 3
3180   IF S(J,K)>0 AND S(J,K)<21 AND INT(S(J,K))=ABS(S(J,K))
3190   PRINT "***** ERROR!!!!!"                  → THEN 3210
3200   GOTO 3150
3210   NEXT K
3220  NEXT J
3230 RETURN
3240 REM-INPUT CAVE
3250  FOR J=1 TO 20
3260   FOR K=1 TO 3
3270   READ S(J,K)
3280   NEXT K
3290  NEXT J
3300 RETURN
3310 END
```

Another new game from Creative Computing....

WAR 3

WAR3 is a version of the tried and true Gunner game, occasionally called the "Battleship" game. It can be played by two or three players, or one playing oneself (still very exciting). The original version was written by Mike Forman, revisions by M. E. Lyon and Brian West of Morse High School, San Diego.

The game is rather straight forward, and generates excitement, with a great amount of vicarious aggression.

A suggested sequence to use with WAR3 might be:

1. Study the trajectories of objects noting:
   a. Angles of Launch
   b. Velocity
   c. Effect of gravity upon projectiles
   d. Distance, and its relation to velocity and angle of launch.
2. Discuss the observations of trajectories of objects such as a ball, a pebble (small), etc.
3. Play WAR3 with the aid of the computer.
4. Challenge those who have played WAR3 to develop a similar game. Maybe, modify WAR3 to allow four or more players.

Listing and sample run of WAR3 on next page.

Another new game from Creative Computing....

DR. Z

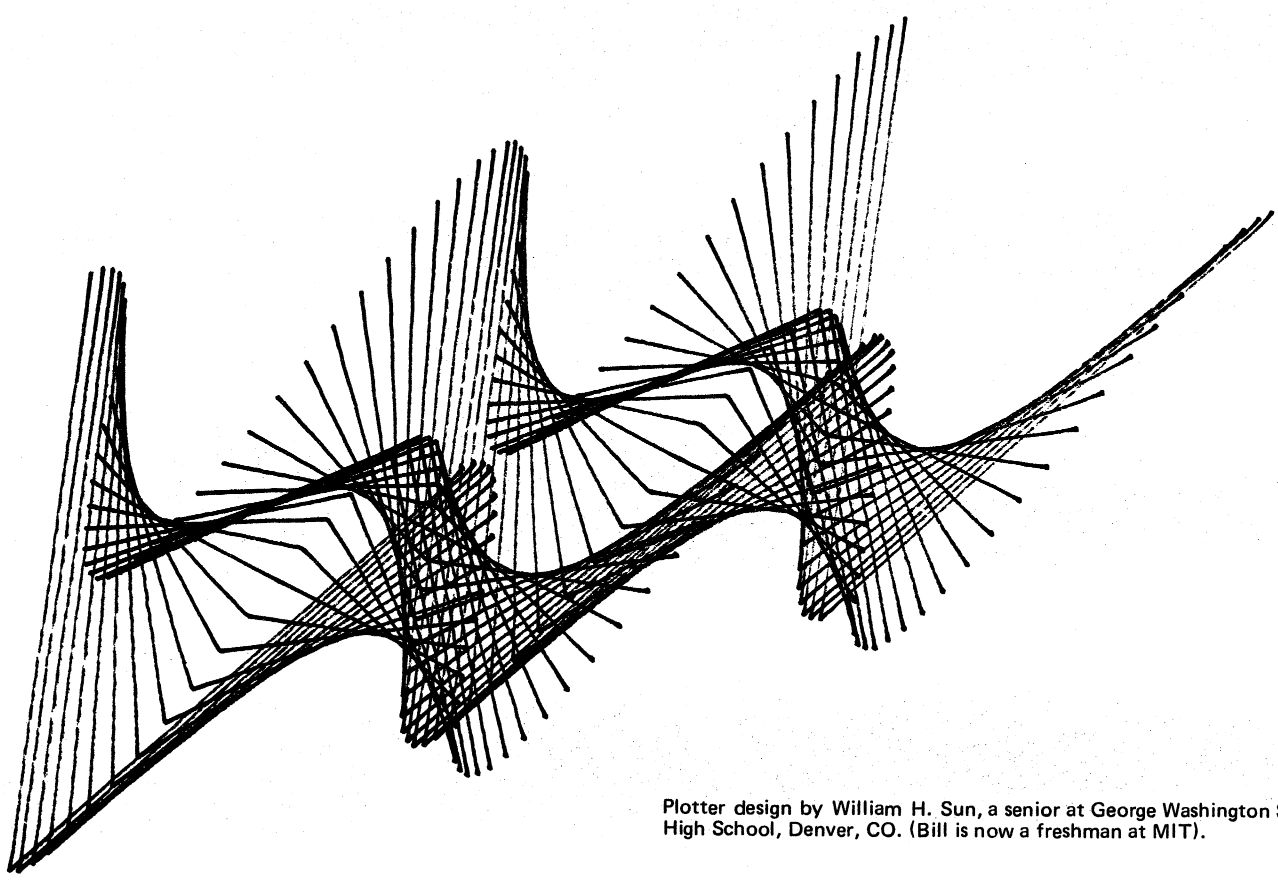
Using DR.Z your computer "interacts" with you in true Rogerian form, never making a value judgment of your response.

DR.Z is multi-lingual and "professional confidence" is guaranteed, especially with a video display terminal. However, if you have a printer, try employing a unique language known only to you and DR.Z.

If you would prefer to employ DR.Z in an educational mode, you might consider the following sequence of activities.

1. Discuss communication, exploring:
   a. The role of spoken language
   b. The role of written language
   c. Non-verbal language
      1. facial expressions
      2. posture of body
      3. hand gestures
2. Experience a session with DR.Z.
3. Develop computerizations of other "purely human" situations.

Listing and sample run of Dr. Z are 1 page over.



Plotter design by William H. Sun, a senior at George Washington Sr. High School, Denver, CO. (Bill is now a freshman at MIT).

WAR3

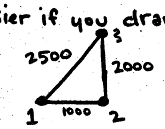
WAR 3
LISTING

```
10  DIM A$[3]
20  T=0
30  REM 'WAR 3'.  ORIGINAL 8K MODV FOCAL BY MIKE FORMAN
40  REM TSS/8 BASIC IV VERSION BY M E LYON JR 1972
50  REM H-P 2000 VERSION BY BRIAN WEST 1975
60  DIM V[3],X[3],P[3],R[3,3]
70  MAT V=ZER
80  MAT X=ZER
90  MAT P=ZER
100  MAT R=ZER
110  DATA 1,2,2,3,3,1,1,3,3,2,2,1,2,3,3,1,1,2,0
120  PRINT "THIS IS THE BASIC VERSION OF 'WAR3'.  TWO OR THREE MAY PLAY"
130  PRINT "DO YOU NEED INSTRUCTIONS";
140  INPUT A$
150  IF A$="YES" THEN 1310
160  PRINT ""
170  PRINT "NO. OF PLAYERS";
180  INPUT N
190  IF N=2 THEN 240
200  IF N=3 THEN 270
210  PRINT "ERROR--TWO OR THREE PLAYERS."
220  PRINT
230  GOTO 160
240  N1=1
250  PRINT ""
260  GOTO 290
270  N1=N
280  PRINT ""
290  FOR J=1 TO N1
300  READ A,B
310  PRINT "DISTANCE (FT.) ";A;" TO ";B;
320  INPUT R[A,B]
330  R[B,A]=R[A,B]
340  NEXT J
350  PRINT ""
360  RESTORE
370  IF N=2 THEN 460
380  FOR J=1 TO N
390  READ A,B,C,D,E,F
400  IF R[A,B]<R[C,D]+R[E,F] THEN 440
410  PRINT "ERROR--ILLEGAL TRIANGLE.  RE-ENTER RANGES."
420  RESTORE
430  GOTO 290
440  NEXT J
450  PRINT
460  FOR J=1 TO N
470  PRINT "MUZZLE VELOCITY (FT./SEC.) OF ";J;
480  INPUT V[J]
490  NEXT J
500  PRINT ""
510  FOR J=1 TO N
520  X[J]=V[J]↑2/32
530  NEXT J
540  FOR A=1 TO N
550  FOR B=1 TO N
560  IF X[A]>R[A,B] THEN 610
570  PRINT "ERROR--";A;" CANNOT REACH ";B
580  PRINT "WHAT IS THE MUZZLE VELOCITY OF ";A;
590  INPUT V[A]
600  GOTO 510
610  NEXT B
620  NEXT A
630  N1=N
640  PRINT ""
650  PRINT ""
660  PRINT "ROUND ";T+1
670  PRINT
680  FOR M=1 TO N
690  IF N=3 THEN 750
700  C=1
710  IF M <> 1 THEN 730
720  C=2
730  PRINT "PLAYER ";M;" SHOOTING AT";C
740  GOTO 890
750  IF P[M]=12 THEN 1280
760  PRINT "PLAYER ";M;" SHOOTING AT";
770  INPUT C
780  IF C=1 THEN 830
790  IF C=2 THEN 830
800  IF C=3 THEN 830
810  PRINT "ERROR--PLAYERS DESIGNATED 1, 2, 3."
820  GOTO 760
830  IF C <> M THEN 860
840  PRINT "ERROR--CANNOT SHOOT SELF."
850  GOTO 760
860  IF P[C] <> 12 THEN 890
870  PRINT "ERROR-- ";C;" IS DEFUNCT"
880  GOTO 760
890  PRINT "FIRING ANGLE";
900  INPUT A3
910  IF A3<0 THEN 940
920  IF A3>180 THEN 940
930  GOTO 970
940  PRINT "ERROR--FIRED INTO GROUND.  ";M;" NOW DEFUNCT."
950  P[M]=12
960  GOTO 760
970  IF A3<90 THEN 1000
980  PRINT "ERROR--FIRED WRONG WAY, LOSE SHOT."
990  GOTO 760
1000  Z=SIN(A3*3.49064E-02)*V[M]↑2/32
1010  X=(R[M,C]/1000*RND(0))-(R[M,C]/1000*RND(0))
1020  D=X+Z
1030  D1=R[M,C]*.05
1040  IF D<D1 THEN 1080
1050  IF ABS(D-R[M,C])<D1 THEN 1110
1060  IF D<R[M,C] THEN 1140
1070  IF D>R[M,C] THEN 1160
1080  PRINT "  TOO CLOSE - ";M;" IS DEFUNCT."
1090  P[M]=12
1100  GOTO 1180
1110  PRINT "  A HIT - ";C;" IS DEFUNCT."
1120  P[C]=12
1130  GOTO 1180
1140  PRINT "  YOU UNDERSHOT BY ";ABS(D-R[M,C]);" FEET."
1150  GOTO 1270
1160  PRINT "  YOU OVERSHOT BY ";ABS(D-R[M,C]);" FEET."
1170  GOTO 1270
1180  N1=N1-1
1190  IF N1>1 THEN 1270
1200  FOR M1=1 TO N
1210  IF P[M1]=12 THEN 1250
1220  PRINT
1230  PRINT "GAME OVER.  ";M1;"WINS."
1240  STOP
1250  NEXT M1
1260  STOP
1270  PRINT ""
1280  NEXT M
1290  T=T+1
1300  GOTO 650
1310  PRINT
1320  PRINT "THIS IS A WAR GAME.  TWO OR THREE PLAYERS ARE GIVEN"
1330  PRINT "(THEORETICAL) CANNONS WITH WHICH THEY ATTEMPT TO SHOOT EACH"
1340  PRINT "OTHER.  THE PARAMETERS FOR DISTANCES AND MUZZLE VELOCITIES ARE"
1350  PRINT "SET AT THE BEGINING OF THE GAME.  THE SHOTS ARE FIRED BY"
1360  PRINT "GIVING A FIRING ANGLE, EXPRESSED IN DEGREES FROM HORIZONTAL"
1370  PRINT
1380  PRINT "THE COMPUTER WILL KEEP TRACK OF THE GAME AND REPORT ALL"
1390  PRINT "MOVES.  A 'HIT' IS SCORED BY FIRING A SHOT WITHIN 5% OF THE TOTAL"
1400  PRINT "DISTANCE FIRED OVER.  GOOD LUCK"
1410  PRINT ""
1420  GOTO 160
1430  END
```

Instructions here — (not repeated in sample run)

SAMPLE RUN

It's easier if you draw a diagram




```
WAR3

THIS IS THE BASIC VERSION OF 'WAR3'.  TWO OR THREE MAY PLAY
DO YOU NEED INSTRUCTIONS?N


NO. OF PLAYERS?3


DISTANCE (FT.) 1   TO  2    ?1000
DISTANCE (FT.) 2   TO  3    ?2000
DISTANCE (FT.) 3   TO  1    ?2500


MUZZLE VELOCITY (FT./SEC.) OF  1    ?300
MUZZLE VELOCITY (FT./SEC.) OF  2    ?350
MUZZLE VELOCITY (FT./SEC.) OF  3    ?400


ROUND  1

PLAYER  1      SHOOTING AT?3
FIRING ANGLE?78
  YOU OVERSHOT BY  144.095      FEET.


PLAYER  2      SHOOTING AT?3
FIRING ANGLE?79
  YOU UNDERSHOT BY  566.111      FEET.


PLAYER  3      SHOOTING AT?1
FIRING ANGLE?76
  YOU UNDERSHOT BY  154.143      FEET.


ROUND  2

PLAYER  1      SHOOTING AT?2
                                  FIRING ANGLE?80
  YOU UNDERSHOT BY  1539.64      FEET.


PLAYER  2      SHOOTING AT?3
FIRING ANGLE?76
  YOU UNDERSHOT BY  202.396      FEET.


PLAYER  3      SHOOTING AT?1
FIRING ANGLE?72
  YOU OVERSHOT BY  149.81      FEET.


ROUND  3

PLAYER  1      SHOOTING AT?3
                                  FIRING ANGLE?80
  YOU UNDERSHOT BY  1536.77      FEET.


PLAYER  2      SHOOTING AT?3
FIRING ANGLE?77
                 A HIT -  3      IS DEFUNCT.
```

Player 2 wiped out Player 3, however, Player 1 is zeroing in on Player 2

```
ROUND  4

PLAYER  1      SHOOTING AT?2
FIRING ANGLE?78
  YOU OVERSHOT BY  144.814      FEET.


PLAYER  2      SHOOTING AT?1
                                  FIRING ANGLE?79
  YOU OVERSHOT BY  434.333      FEET.


ROUND  5

PLAYER  1      SHOOTING AT?2
FIRING ANGLE?79
  YOU OVERSHOT BY  53.9304      FEET.


PLAYER  2      SHOOTING AT?1
FIRING ANGLE?78
  YOU UNDERSHOT BY  74.1608      FEET.


ROUND  6

PLAYER  1      SHOOTING AT?2
FIRING ANGLE?8P
  A HIT -  2      IS DEFUNCT.

GAME OVER.   1      WINS.

DONE
```

DR. Z

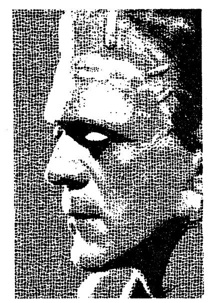
LISTING

```
LIST
DR.Z

10  DIM A$[72],B$[72]
15  REM  DEVELOPED BY  DR.Z  1972
20  PRINT "HELLO THERE, I'M YOUR COMPUTERTHERAPIST."
30  PRINT "WHAT IS YOUR NAME? AND TELL ME SOMETHING ABOUT YOURSELF."
40  PRINT "HOWEVER, DON'T TYPE MORE THAN ONE LINE. I TIRE EASILY."
50  INPUT A$
60  PRINT "WHAT DID YOU SAY YOUR NAME WAS AGAIN?"
70  INPUT B$
80  PRINT "HOW DO YOU FEEL TODAY?"
90  LET C=U=V=0
100  INPUT A$
110  PRINT
120  PRINT
130  IF C=10 THEN 720
140  LET Z=INT(10*RND(0))
150  IF U=Z THEN 140
160  IF V=Z THEN 140
170  LET U=Z
180  IF Z <> 0 THEN 200
190  GOTO 690
200  IF Z <> 1 THEN 220
210  GOTO 420
220  IF Z <> 2 THEN 240
230  GOTO 450
240  IF Z <> 3 THEN 260
250  GOTO 480
260  IF Z <> 4 THEN 280
270  GOTO 510
280  IF Z <> 5 THEN 300
290  GOTO 540
300  IF Z <> 6 THEN 320
310  GOTO 570
320  IF Z <> 7 THEN 340
330  GOTO 600
340  IF Z <> 8 THEN 360
350  GOTO 630
360  IF Z <> 9 THEN 380
370  GOTO 660
380  GOTO 690
390  PRINT "THAT'S VERY INTERESTING, TELL ME MORE."
400  PRINT
410  GOTO 690
420  PRINT "HAVE YOU FELT THIS WAY LONG?"
430  PRINT
440  GOTO 690
450  PRINT "DO YOU THINK THIS IS REASONABLE IN LIGHT OF YOUR INTERESTS?"
460  PRINT
470  GOTO 690
480  PRINT "DO YOUR FRIENDS FIND THIS ACCEPTABLE?"
490  PRINT
500  GOTO 690
510  PRINT "DO YOU FEEL COMFORTABLE WITH THIS FEELING?"
520  PRINT
530  GOTO 690
540  PRINT "DO YOU THINK THAT THIS IS A NORMAL FEELING?"
550  PRINT
560  GOTO 690
570  PRINT "WHY DO YOU THINK YOU FEEL THIS WAY?"
580  PRINT
590  GOTO 690
600  PRINT "HAVE YOU TALKED TO ANYONE ABOUT THIS?"
610  PRINT
620  GOTO 690
630  PRINT "WHY ARE YOU HERE?"
640  PRINT
650  GOTO 690
660  PRINT "ARE YOU SATISFIED WITH THE WAY YOUR IDEAS ARE DEVELOPING?"
670  PRINT
680  GOTO 690
690  LET C=C+1
700  LET U=Z
710  GOTO 100
720  PRINT "I THINK YOU ARE MAKING A GREAT ATTEMPT TO SOLVE YOUR"
730  PRINT "DIFFICULTIES, AND I SEE NO NEED TO CONTINUE THIS"
740  PRINT "SESSION ANY FURTHER."
750  PRINT B$" WOULD YOU MAKE ANOTHER APPOINTMENT WITH MY COMPUTER"
760  PRINT "FOR SOMETIME IN THE NEXT FEW WEEKS.WHAT DATE WOULD YOU"
770  PRINT "PREFER?"
780  INPUT A$
790  PRINT "THAT WILL BE FINE."
800  PRINT "I'VE ENJOYED COMMUNICATING WITH YOU."
810  PRINT "HAVE A NICE DAY."
820  FOR T=1 TO 6
830  PRINT
840  NEXT T
850  END
```

If your BASIC has ON-GOTO statements, you should be able to improve this coding dramatically.

SAMPLE RUN

```
RUN
DR.Z

HELLO THERE, I'M YOUR COMPUTERTHERAPIST.
WHAT IS YOUR NAME? AND TELL ME SOMETHING ABOUT YOURSELF.
HOWEVER, DON'T TYPE MORE THAN ONE LINE. I TIRE EASILY.
?HOMEEE, AND I HUHFGVB DGVXFCSDZX BHVBN JMNHB FEELINGS.
WHAT DID YOU SAY YOUR NAME WAS AGAIN?
?HOMEREE
HOW DO YOU FEEL TODAY?
?LIKE I SAID BEFORE, YOU KNOW.


WHY ARE YOU HERE?

?I FEEL YOU KNOW, UNDERSTAND?


ARE YOU SATISFIED WITH THE WAY YOUR IDEAS ARE DEVELOPING?

?YOU KNOW, HE, HIM, THEM, YOU KNOW.


HAVE YOU FELT THIS WAY LONG?

?I DON'T, YOU KNOW, THAT THIS THING IS, YOU KNOW.


WHY ARE YOU HERE?

?YOU KNOW, IT IS THE THING NOW, YOU KNOW.


HAVE YOU FELT THIS WAY LONG?

?YA, UGH, YOU KNOW.


DO YOU THINK THIS IS REASONABLE IN LIGHT OF YOUR INTERESTS?

?YA, YOU KNOW, THIS IS COMMUNICATION,YOU KNOW.


WHY ARE YOU HERE?

?YOU KNOW, YOU KNOW.


DO YOUR FRIENDS FIND THIS ACCEPTABLE?

?YA, WE ALL COMMUNICATE, YOU KNOW.


HAVE YOU TALKED TO ANYONE ABOUT THIS?

?ALL MY PEOPLE, YOU KNOW.


DO YOU THINK THAT THIS IS A NORMAL FEELING?

?THIS IS IT, YOU KNOW.


I THINK YOU ARE MAKING A GREAT ATTEMPT TO SOLVE YOUR
DIFFICULTIES, AND I SEE NO NEED TO CONTINUE THIS
SESSION ANY FURTHER.
HOMEREE WOULD YOU MAKE ANOTHER APPOINTMENT WITH MY COMPUTER
FOR SOMETIME IN THE NEXT FEW WEEKS.WHAT DATE WOULD YOU
PREFER?
?HOW ABOUT, YOU KNOW.
THAT WILL BE FINE.
I'VE ENJOYED COMMUNICATING WITH YOU.
HAVE A NICE DAY.




DONE
```



Another new game from Creative Computing . . . .

ROADRACE

Author: Unknown
Modified by: Bill Cotter, Pittsfield, Mass.
Description: You are the driver of a race car on the notorious NY Route 20. You'll have to drive 5 miles with ½ gallon of gas, while keeping alert for changes in the road conditions, other cars, etc.

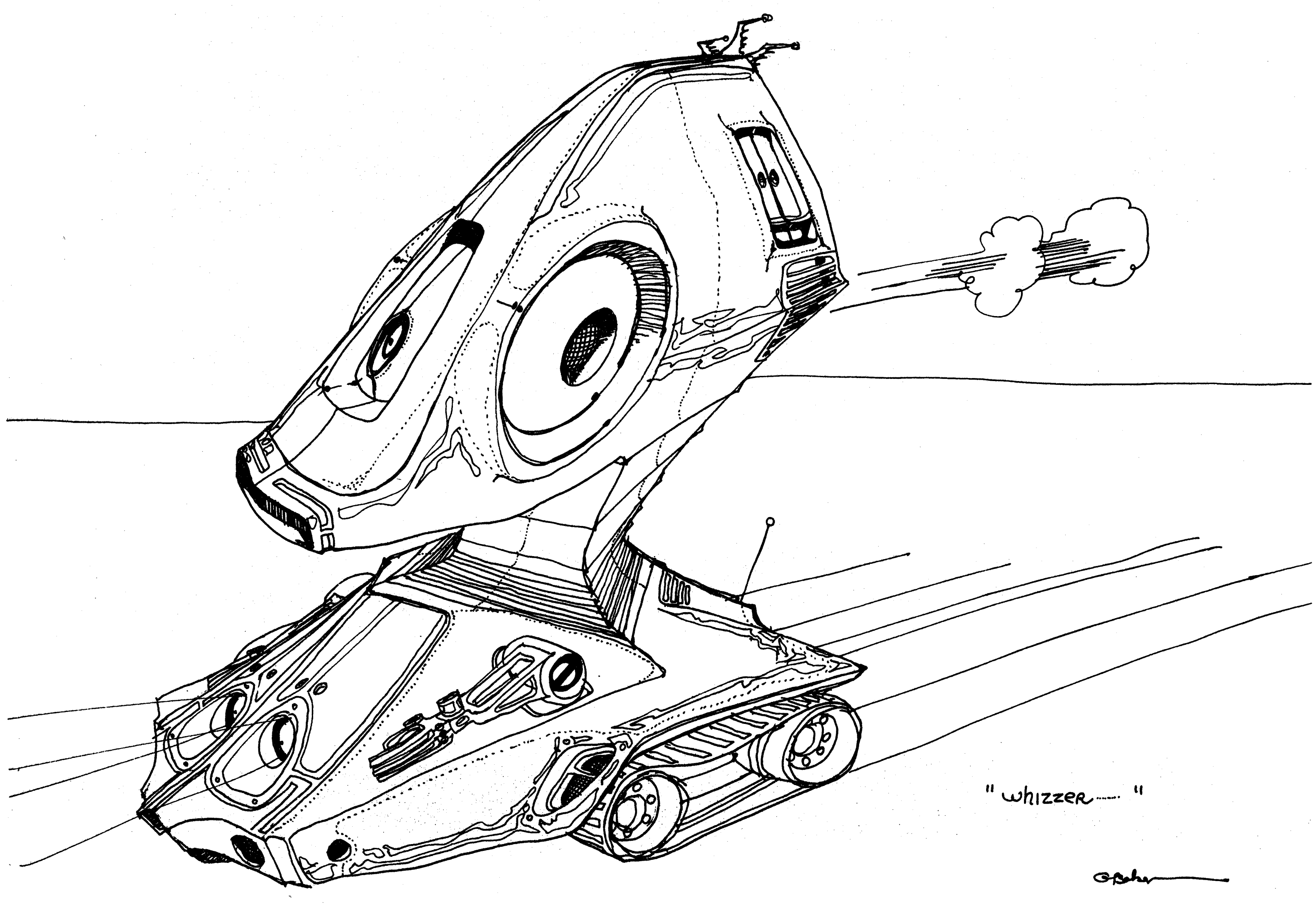
At the start you pick your car and course. During the race you control braking and acceleration.

Watch out for passing another car! If you try to go the same speed he's going, you're going to meet a Greyhound bus head-on!

Suggestions: The game is tough to win. I usually wipe out in a curve or run out of gas. You might want to increase your MPG rating . . . look at line 870.

Good luck!

Listing and run on next page.



ROADRACE

LISTING

```
100  PRINT "           THIS IS THE PITTSFIELD-ALBANY"
110  PRINT "                       ROAD RALLY"
120  PRINT
130 PRINT "WELCOME TO THE FIRST ANNUAL PITTSFIELD-TO-ALBANY ROAD RALLY."
140 PRINT "YOU'LL BE DRIVING DOWN RT. 20, TRYING TO WIN THE RACE AND"
150 PRINT "STAY ALIVE IN THE BARGAIN.  GOOD LUCK!!"
160 PRINT
170 PRINT "YOU HAVE YOUR CHOICE OF: (1) A VW; (2) 283 NOVA;"
180 PRINT "(3) Z-28; OR (4) FERRARI"
190 PRINT
200 PRINT "CHOOSE THE CAR YOU WANT BY THE NUMBER IN FRONT OF IT."
210 PRINT "REMEMBER, THE BETTER THE CAR THE MORE GAS IT USES."
220 PRINT "WHICH CAR";
230  INPUT C1
240  LET C1=INT(C1)
250 IF C1>4 THEN 280
260 IF C1 <1 THEN 280
270 GO TO 300
280 PRINT "INVALID CAR NUMBER. NEW CAR ";
290 GO TO 230
300 PRINT
310 IF N2=1 THEN 350
320 PRINT "YOU NOW CHOOSE WHICH COURSE YOU WANT TO RACE ON."
330 PRINT "THE EASIEST COURSE IS NUMBER 1, AND IS THE STRAIGHTEST"
340 PRINT "ROUTE.  NUMBER 5 CONSISTS MOSTLY OF TURNS AND TWISTS."
350 PRINT "WHICH COURSE DO YOU WANT (1 TO 5)";
360 INPUT C2
370 LET C2=INT(C2)
380 IF C2 < 1 THEN 410
390 IF C2 >5 THEN 410
400 GO TO 430
410 PRINT "INVALID COURSE NUMBER. NEW CHOICE ";
420  GO TO 360
430 IF N2=1 THEN 490
440 PRINT "YOU WILL NEED TO TRAVEL 5 MILES WITH .5 GALLONS OF GAS"
450 PRINT "YOUR STATUS WILL BE SHOWN EACH 10 SECONDS.  AFTER EACH STATUS"
460 PRINT "CHECK YOU WILL BE ASKED FOR A NEW RATE OF GAS.  A RATE OF"
470 PRINT "+10 IS HARD ACCELERATION, AND -10 IS HARD BRAKING.  ANY NUMBER"
480 PRINT "IN BETWEEN IS ALLOWABLE."
490 FOR I=1 TO C1
500 READ B,M,S
510 LET B=B/10
520 NEXT I
530 LET A1=.5
540 LET M1=0
550 LET C1=C1/2
560 LET V=0
570 PRINT
580 LET R1=0
590 LET T=0
600 LET D=0
610 LET Q1=0
620 PRINT "PRESENT VELOCITY = ";V;" NO. OF GALLONS = ";A1
630 PRINT "NO. OF MILES = ";M1;" TIME PASSED = ";T;" SECONDS"
640 IF M1>=5 THEN 1460
650 PRINT "WHAT IS YOUR NEW RATE OF GAS ";
660 INPUT G
670IF G<-10 THEN 700
680 IF G>10 THEN 700
690 GO TO 720
700 PRINT "NOT VALID. NEW RATE ";
710 GO TO 660
720 IF G<9 THEN 780
730 LET Z=Z+1
740 IF Z>4 THEN 760
750 GO TO 790
760 PRINT "YOUR ENGINE BLEW.  YOU GOT HIT BY A PISTON."
770 GO TO 1270
780 LET Z=0
790 LET V=INT(B*G-M*V+V)
800 LET T=T+10
810 PRINT
820 PRINT "ROAD CONDITIONS :";
830 IF V>0 THEN 850
840 LET V=0
850 LET M1=M1+V/460
860 IF G<0 THEN 890
870 LET A1=A1-(G*S)/5000
880 IF A1<0 THEN 1380
890 IF R1=1 THEN 1050
900 IF Q1=1 THEN 980
910  LET Q=INT((C2+1)*RND(X))
920 LET R=INT((3.75-C2)*RND(X))
930 IF R>0 THEN 1290
940 IF Q>0 THEN 1340
950 PRINT "CLEAR AND STRAIGHT"
960 PRINT
970 GO TO 620
980 LET H=INT(15+35.*RND(X))
990 LET H=H+5*C1
1000 IF V>H THEN 1500
1010 PRINT "THROUGH CURVE"
1020 PRINT
1030 LET Q1=0
1040 GO TO 620
1050 LET E=E-(V-D)*3.0
1060 IF E<0 THEN 1100
1070 PRINT "VEHICLE ";E;" FEET AHEAD"
1080 PRINV
1090 GO TO 620
1100 IF V-D<5 THEN 1180
1110 PRINT "VEHICLE PASSED BY ";
1120 LET D=V-D
1130 PRINT D;
1140 PRINT " MPH"
1150 PRINT
1160 LET R1=0
1170 GO TO 620
1180 PRINT "VEHICLE BEING PASSED "
1190 LET  D=INT(25+40*RND(X))
1200 PRINT "GRAYHOUND BUS IN OTHER LANE ";
1210 PRINT "DOING ";
1220 PRINT D;
1230 PRINT " MPH ";
1240 LET D=V+D
1250 PRINT "CRASH VELOCITY = ";
1260 PRINT D
1270 PRINV "WHERE IS YOUR FUNERAL BEING HELD ?"
1280 GO TO 1560
1290 PRINT "VEHICLE AHEAD 1000 FEET"
1300 PRINT
1310 LET D=INT(25+35*RND(X))
1320 LET R1=1
1330 GO TO 620
1340 PRINT " WARNING: CURVE AHEAD "
1350 LET Q1=1
1360 PRINT
1370 GO TO 620
1380 PRINT "EXCELLENT BUT WAIT!"
1390 PRINT
1400 PRINT "YOU RAN OUT OF GAS"
1410 GO TO 1550
1420 PRINT "BUT SOME HOW YOU MADE IT"
1430 PRINT
1440 LET R1=0
1450 GO TO 620
1460 PRINT
1470 PRINT
1480 PRINT "YOU MADE IT (LUCKY) !!!!!!!"
1490 GO TO 1560
1500  PRINT "ARE TERRIBLE"
1510 LET H=H-5*C1
1520 PRINT H;" WAS THE SPEED THROUGH THE CURVE"
1530 PRINT V;" WAS YOUR SPEED. BY THE WAY ";
1540 GO TO 1270
1550 PRINT "YOU LEAD FOOTED *#$%#$#&"
1560 PRINT "YOU WANT TO TRY AGAIN, RIGHT !!!!"
1570 PRINT "1-YES, 2-NO ";
1580 INPUT V
1590 IF V=2 THEN 1620
1600 N2=1
1610 GO TO 1640
1620 PRINT "CHICKEN"
1630 GO TO 1700
1640 RESTORE
1650 GO TO 220
1660 DATA 45,.53,10
1670 DATA 60,.5,13
1680 DATA 70,.41,15
1690 DATA 80,.39,18
1700 END
```

SAMPLE RUN

```
WELCOME TO THE FIRST ANNUAL PITTSFIELD-TO-ALBANY ROAD RALLY.
YOU'LL BE DRIVING DOWN RT. 20, TRYING TO WIN THE RACE AND
STAY ALIVE IN THE BARGAIN.  GOOD LUCK!!

YOU HAVE YOUR CHOICE OF: (1) A VW; (2) 283 NOVA;
(3) Z-28; OR (4) FERRARI

CHOOSE THE CAR YOU WANT BY THE NUMBER IN FRONT OF IT.
REMEMBER, THE BETTER THE CAR THE MORE GAS IT USES.
WHICH CAR ?1

YOU NOW CHOOSE WHICH COURSE YOU WANT TO RACE ON.
THE EASIEST COURSE IS NUMBER 1, AND IS THE STRAIGHTEST
ROUTE.  NUMBER 5 CONSISTS MOSTLY OF TURNS AND TWISTS.
WHICH COURSE DO YOU WANT (1 TO 5) ?2
YOU WILL NEED TO TRAVEL 5 MILES WITH .5 GALLONS OF GAS
YOUR STATUS WILL BE SHOWN EACH 10 SECONDS.  AFTER EACH STATUS
CHECK YOU WILL BE ASKED FOR A NEW RATE OF GAS.  A RATE OF
+10 IS HARD ACCELERATION, AND -10 IS HARD BRAKING.  ANY NUMBER
IN BETWEEN IS ALLOWABLE.

PRESENT VELOCITY =     0  NO. OF GALLONS =    .5
NO. OF MILES =     0  TIME PASSED =     0  SECONDS
WHAT IS YOUR NEW RATE OF GAS  ?10

ROAD CONDITIONS : WARNING: CURVE AHEAD

PRESENT VELOCITY =    45  NO. OF GALLONS =   .48
NO. OF MILES =    .0978261  TIME PASSED =    10  SECONDS
WHAT IS YOUR NEW RATE OF GAS  ?2        <--- SLOW DOWN FOR CURVE

ROAD CONDITIONS :THROUGH CURVE

PRESENT VELOCITY =    30  NO. OF GALLONS =  .476
NO. OF MILES =    .1630435  TIME PASSED =    20  SECONDS
WHAT IS YOUR NEW RATE OF GAS  ?8

ROAD CONDITIONS :CLEAR AND STRAIGHT

PRESENT VELOCITY =    50  NO. OF GALLONS =   .46
NO. OF MILES =    .2717391  TIME PASSED =    30  SECONDS
HARD ACCELERATION --> WHAT IS YOUR NEW RATE OF GAS  ?10

ROAD CONDITIONS :VEHICLE AHEAD 1000 FEET

PRESENT VELOCITY =    68  NO. OF GALLONS =   .44
NO. OF MILES =    .4195652  TIME PASSED =    40  SECONDS
WHAT IS YOUR NEW RATE OF GAS  ?4

ROAD CONDITIONS :VEHICLE PASSED BY      6  MPH

PRESENT VELOCITY =    49  NO. OF GALLONS =  .432
NO. OF MILES =    .5260869  TIME PASSED =    50  SECONDS
WHAT IS YOUR NEW RATE OF GAS  ?10

ROAD CONDITIONS :VEHICLE AHEAD 1000 FEET

PRESENT VELOCITY =    68  NO. OF GALLONS =  .412
NO. OF MILES =    .673913  TIME PASSED =    60  SECONDS
WHAT IS YOUR NEW RATE OF GAS  ?5

ROAD CONDITIONS :VEHICLE PASSED BY     22  MPH

PRESENT VELOCITY =    54  NO. OF GALLONS =  .402
NO. OF MILES =    .7913043  TIME PASSED =    70  SECONDS
WHAT IS YOUR NEW RATE OF GAS  ?10

ROAD CONDITIONS : WARNING: CURVE AHEAD

PRESENT VELOCITY =    70  NO. OF GALLONS =  .382
NO. OF MILES =    .9434782  TIME PASSED =    80  SECONDS
WHAT IS YOUR NEW RATE OF GAS  ?4        <--- DIDN'T SLOW ENOUGH FOR CURVE

ROAD CONDITIONS :ARE TERRIBLE
 39  WAS THE SPEED THROUGH THE CURVE
 50  WAS YOUR SPEED. BY THE WAY WHERE IS YOUR FUNERAL BEING HELD ?
```

Concen-tration

by Paul Calter
Vermont Technical College

How good are your personal memory banks? Find out by playing this computer version of the well-known card game, CONCENTRATION.

The entire deck of 52 cards is shuffled, and dealt face down on the table. The player turns over any two cards he pleases. If they are a *pair,* they are removed from play, the player scores one point, and picks again. If they are not a pair, the player tries to memorize the cards and their location for future picks, and returns them face down on the table in the same location.

In this computer version, submitted by James Vanderbeek of Vermont Technical College, the computer "deals" the cards in positions one to fifty-two. There is only one player (the computer does not pick cards in this version) who enters the numbers of the positions chosen. The computer then tells you which cards you have picked. If they are a pair, it removes them from play, and your score is incremented: if not, the identity of the cards is overprinted, and you have a chance to pick again.

After a total of 26 picks, the play ends and your score is printed, along with the computer's candid assessment of your ability.

Some possible expansions of this program are:

1. Play against the computer — where the computer picks its pairs completely at random, from the cards remaining on the "table."
2. Play against the computer — where the computer is blessed with a perfect memory of all cards previously overturned.
3. Play against the computer — where the computer usually remembers the cards previously shown, but sometimes has "lapses" of memory. (How would you program that?) This version would probably provide the most interesting play.
4. Play against another person, using the same terminal.

PROGRAM LISTING

```
190 RANDOMIZE
200 DIM A$(52),C$(52)
210 FOR X = 1 TO 52                          'MAKE THE DECK
220    READ E$
230    LET A$(X) = E$
240 NEXT X
250 ' SHUFFLE AND DEAL
260 FOR Z = 1 TO 1000
270    LET B$ = A$(INT(RND*52+1))
280    FOR V = 0 TO 51
290       IF B$ = C$(V) THEN 330
300    NEXT V
310    LET C$(D) = B$
320    LET D = D+1
330 NEXT Z
340 ' START TO PLAY
350 FOR N = 1 TO 26
360    PRINT "FIRST CARD";
370    INPUT U
380    LET G = 1
390    IF C$(U) = " " THEN 840
400    PRINT "SECOND CARD";
410    INPUT W
420    LET G = 2
430    IF C$(W) = " " THEN 840
440    IF U<>W THEN 470
450    PRINT "YOU CAN'T PICK THE SAME CARD TWICE!"
460    GOTO 400
470    IF SEG$(C$(U),1,1) = SEG$(C$(W),1,1) THEN 580
480    PRINT "#";U;"IS ";C$(U),"#";W;"IS ";C$(W),  'REVEAL THE CARDS
490    FOR I = 1 TO 50
500       PRINT "";
510    NEXT I
520    FOR P = 1 TO 4                          'HIDE THE CARDS
530       PRINT CHR$(13);"99999999999999999999999999999999";
540    NEXT P
550    PRINT
560    PRINT
570    GOTO 630
580    PRINT "THAT'S A MATCH --";C$(U),C$(W)
590    LET C$(U) =" "                         'TAKE THE PAIR
600    LET C$(W) =" "
610    LET S = S+1
620    PRINT "YOUR SCORE IS NOW";S;" YOU HAVE HAD";N;"PICKS."
630 NEXT N
640 ' THE RESULTS
650 LET S1 = S/(N/4)
660 PRINT "YOU SCORED";S;"OUT OF";N;"THAT IS ";
670 ON S1 + 1.5 GOTO 680,700,720,760
680 PRINT "POOR."
690 GOTO 770
700 PRINT "FAIR."
710 GOTO 770
720 PRINT "GOOD."
730 GOTO 770
740 PRINT "FANTASTIC! ! !"
750 GOTO 770
760 PRINT ". . . AAAH. . .UH....YOU MUST HAVE CHEATED!"
770 PRINT
780 PRINT "DO YOU WANT TO TRY AGAIN";
790 INPUT Z$
800 IF F$ = "YES" THEN 260
810 PRINT
820 PRINT "COME BACK AGAIN!!"
830 STOP
840 PRINT "YOU HAVE ALREADY MATCHED THAT CARD."
850 IF G = 1 THEN 360
860 GOTO 400
870 DATA"AS","2S","3S","4S","5S","6S","7S","8S","9S","10S","JS","KS","QS"
880 DATA"AH","2H","3H","4H","5H","6H","7H","8H","9H","10H","JH","QH","KH"
890 DATA"AD","2D","3D","4D","5D","6D","7D","8D","9D","10D","JD","QD","KD"
900 DATA"AC","2C","3C","4C","5C","6C","7C","8C","9C","10C","JC","QC","KC"
910 END
```

SAMPLE RUN

```
FIRST CARD? 12
SECOND CARD? 3
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

FIRST CARD? 14
SECOND CARD? 51
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

FIRST CARD? 12
SECOND CARD? 14
THAT'S A MATCH --JD           JH
YOUR SCORE IS NOW 1  YOU HAVE HAD 3 PICKS.
FIRST CARD? 22
SECOND CARD? 45
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

FIRST CARD? 31
SECOND CARD? 36
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

FIRST CARD? 45
SECOND CARD? 31
THAT'S A MATCH --QD           QC
YOUR SCORE IS NOW 2  YOU HAVE HAD 6 PICKS.
FIRST CARD? 44
SECOND CARD? 16
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

FIRST CARD? 29
SECOND CARD? 50
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

FIRST CARD? 29
SECOND CARD? 44
THAT'S A MATCH --10S          10C
YOUR SCORE IS NOW 3  YOU HAVE HAD 9 PICKS.
FIRST CARD? 50
SECOND CARD? 3
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

FIRST CARD? 30
SECOND CARD? 6
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
```

Another new game from Creative Computing....

Condot*

as reviewed by
Peter Olivieri
Boston College

You don't have to have connections to connect with this program. It's the old childhood favorite of "connect the dots." The objective, however, is not to draw a picture by connecting the dots, but rather to carve out squares of "real estate" with the computer as an able adversary. The player who connects the two dots which complete a square gets ownership of that square. In addition, the player gets the added bonus of moving once more. This can be quite advantageous in certain situations. A nine-square grid is provided as a playing board.

The instructions (see REM statements in program) are not as clear as they might be. When you wish to connect two dots you must type in the coordinates (row, column) of the empty space between the dots. However, in identifying what the coordinates are, you must count the "dots" of the grid also. Lest this be equally confusing, a sample of the grid follows complete with an identification of each coordinate where a line may be drawn.

Grid is on page with listing.

In games that I played, the same moves could be replicated in a succeeding game. Thus, once you discover a winning game, you cannot lose (this may be a function of the particular random number generator in use). You'll find it interesting to note that the computer mirrors the player's move in so far as possible. You may also find that the game moves rather slowly, especially for the first three or four moves. Be patient! Once squares begin to fall, the games moves swiftly to its conclusion.

There are some modifications that you may wish to consider if you are going to adopt this program for regular use. In addition to improving the REM statements in the program, I would suggest:

1. Modifying the program so that the grid is printed after *both* players have moved (rather than each time a move is made);
2. Modifying the program so that once a player had ownership of a majority of the squares, the game would end rather than proceed to its inevitable conclusion; and
3. Modify the program so that the player's initials appear in each square he captures.

An interesting problem developed in the running of this program that you should be aware of. One particular sequence of moves resulted in a "hangup" in the execution of the program. This may be the result of a programming error, a typing error somewhere along the line, a system problem, or whatever. A few of us spent a little time debugging, but to no avail. In any case, five runs using the following sequence of moves resulted in an average execution time (until interrupt) of 40 CPU seconds.

Your moves:		
	1.	1,2
	2.	2,1
	3.	2,3
	4.	5,6
	5.	2,5
	6.	4,5
	7.	6,1
	8.	7,2

A sample run of a game (in this instance, the player being triumphant) follows. While the game may quickly lose its interest to all but those with nothing else to do, the program itself is well worth examining. Creating the flowcharts and source statements for this game would be a worthwhile endeavor, particularly as a project in a programming course. Furthermore, even the process of tracing through the existing source program to discover the game's algorithm would be a worthwhile exercise for the knowledgeable programmer.

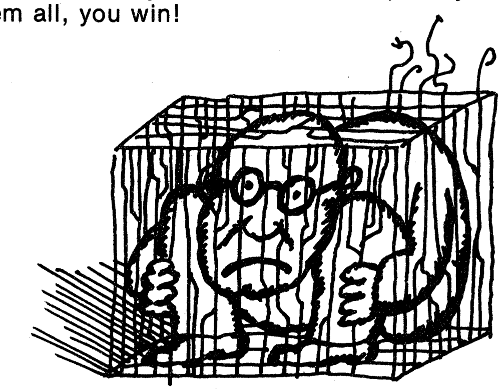
Listing and run of CONDOT 1 page over.

*CONDOT was written by Chuck Lund, St. Paul Public Schools, MN.

Another new game from Creative Computing....

CHASE

Author: Unknown
Modified by: Bill Cotter, Pittsfield, Mass.
Language: BASIC (Honeywell 600/6000)
Description: CHASE puts you in a maze made up of high-voltage fences and posts. This in itself isn't too unpleasant but there're also the five interceptor robots bent on just one thing—your destruction. If these robots touch you . . . that's the end of the game (and you!). There's one hope—make the robots hit the maze, or each other (they're like people—sometimes they'd rather be alone). If you destroy them all, you win!



Listing and run of CHASE on the next page.

CHASE

LISTING

```
100 REM CHASE
110 REM AUTHOR: UNKNOWN
120 REM MODIFIED TO RUN ON HONEYWELL 600/6000 BY BILL COTTER
130 PRINT"YOU ARE WITHIN THE WALLS OF A HIGH VOLTAGE MAZE"
140 PRINT"THERE ARE FIVE SECURITY MACHINES TRYING TO DESTROY YOU"
150 PRINT"YOU ARE THE '*'       THE INTERCEPTORS ARE THE '+'"
160 PRINT"THE AREAS MARKED 'X' ARE HIGH VOLTAGE"
170 PRINT"YOUR ONLY CHANCE FOR SURVIVAL IS TO MANUVER EACH "
180 PRINT"INTERCEPTOR INTO AN 'X'. ---- GOOD LUCK ----"
190 PRINT"MOVES ARE:  1.2.3"\PRINT"            8.*.4"
200 PRINT"            7.6.5"\PRINT
210 DIM A(10,20),N(2)
220 FOR B=1 TO 10
230 FOR C=1 TO 20
240 LET X=INT(10*RND(-1))
250 IF X=5 THEN 280
260 LET A(B,C)=ASC( )
270 GOTO 290
280 LET A(B,C)=ASC(X)
290 NEXT C
300 NEXT B
310 FOR D=1 TO 10
320 LET A(D,1)=ASC(X)
330 NEXT D
340 FOR E=1 TO 10
350 LET A(E,20)=ASC(X)
360 NEXT E
370 FOR F=1 TO 20
380 LET A(1,F)=ASC(X)
390 NEXT F
400 FOR G=1 TO 20
410 LET A(10,G)=ASC(X)
420 NEXT G
430 GOTO 500
440 LET H=INT(1+(10*RND(-1)))\IF H>10 THEN 440
450 LET I=INT(1+(20*RND(-1)))\IF I>20 THEN 450
460 IF A(H,I)=ASC(X) THEN 440
470 IF A(H,I)=ASC(*) THEN 440
480 IF A(H,I)=ASC(+) THEN 440
490 RETURN
500 GOSUB 440
510 LET A(H,I)=ASC(*)\LET J=H\LET K=I
520 GOSUB 440
530 LET A(H,I)=ASC(+)\LET L=H\LETM=I
540 GOSUB 440
550 LET A(H,I)=ASC(+)\LET N=H\LET O=I
560 GOSUB 440
570 LET A(H,I)=ASC(+)\LET P=H\LET Q=I
580 GOSUB 440
590 LET A(H,I)=ASC(+)\LET R=H\LET S=I
600 GOSUB 440
610 LET A(H,I)=ASC(+)\LET T=H\LETU=I
620 N(0)=1
630 FOR D2=1 TO 10
640 FOR B2=1 TO 20
650 N(1)=A(D2,B2)\CHANGE N TO N$\PRINT N$;
660 NEXT B2
670 PRINT
680 NEXT D2
690 INPUT Y\IF Y=0 THEN 800
700 LET V=J\LET W=K
710 ON Y GOTO 720,730,740,750,760,770,780,790
720 LET J=J-1\LET K=K-1\GOTO 800
730 LET J=J-1\GOTO 800
740 LET J=J-1\LET K=K+1\GOTO 800
750 LET K=K+1\GOTO 800
760 LET J=J+1\LET K=K+1\GOTO 800
770 LET J=J+1\GOTO 800
780 LET J=J+1\LET K=K-1\GOTO 800
790 LET K=K-1
800 IF A(J,K)=ASC(X) THEN 1160
810 LET A(V,W)=ASC( )\LET A(J,K)=ASC(*)
820 GOTO 940
830 IF A(X,Y)=ASC(X) THEN 910
840 LET V=X\LET W=Y
850 LET X=SGN(J-X)\LET Y=SGN(K-Y)
860 LET X=X+V\LET Y=Y+W
870 IF A(X,Y)=ASC(*) THEN 920\IF A(X,Y)=ASC( ) THEN 890
880 LET A(V,W)=ASC( )\RETURN
890 LET A(X,Y)=ASC(+)
900 LET A(V,W)=ASC( )
910 RETURN
920 PRINT "*** YOU HAVE BEEN DESTROYED BY A LUCKY COMPUTER ***"
930 GO TO 1180
940 LET X=L\LET Y=M\GOSUB 830
950 LET L=X\LET M=Y
960 LET X=N\LET Y=O\GOSUB 830
970 LET N=X\LET O=Y
980 LET X=P\LET Y=Q\GOSUB 830
990 LET P=X\LET Q=Y
1000 LET X=R\LET Y=S\GOSUB 830
1010 LET R=X\LET S=Y
1020 LET X=T\LET Y=U\GOSUB 830
1030 LET T=X\LET U=Y
1040 IF A(L,M)=ASC(X) THEN 1060
1050 GOTO 630
1060 IF A(N,O)=ASC(X) THEN 1080
1070 GOTO 630
1080 IF A(P,Q)=ASC(X) THEN 1100
1090 GOTO 630
1100 IF A(R,S)=ASC(X) THEN 1120
1110 GOTO 630
1120 IF A(T,U)=ASC(X) THEN 1140
1130 GOTO 630
1140 PRINT"YOU HAVE DESTROYED ALL YOUR OPPONENTS-THE GAME IS YOURS"
1150 GO TO 1180
1160 PRINT"YOU TOUCHED THE FENCE !!!!!!!!!!!"
1170 PRINT "***** ZAP ***** YOU'RE DEAD!!!"
1180 PRINT "ANOTHER GAME (YES OR NO)";
1190 INPUT N9$
1200 IF N9$<>"YES" THEN 1220
1210 GO TO 210
1220 END

ready

*
```

SAMPLE RUN

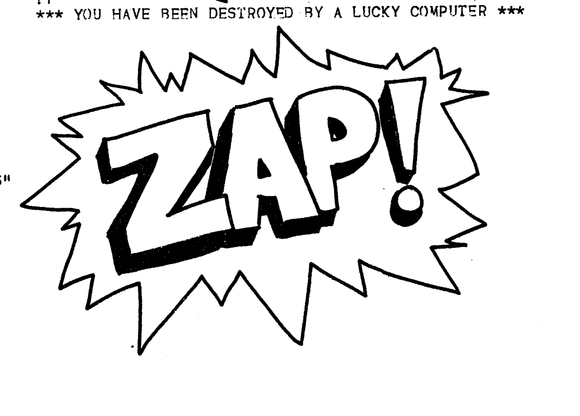
```
YOU ARE WITHIN THE WALLS OF A HIGH VOLTAGE MAZE
THERE ARE FIVE SECURITY MACHINES TRYING TO DESTROY YOU
YOU ARE THE '*'       THE INTERCEPTORS ARE THE '+'
THE AREAS MARKED 'X' ARE HIGH VOLTAGE
YOUR ONLY CHANCE FOR SURVIVAL IS TO MANUVER EACH
INTERCEPTOR INTO AN 'X'. ---- GOOD LUCK ----
MOVES ARE:  1.2.3
            8.*.4
            7.6.5

XXXXXXXXXXXXXXXXXXXX
X X   X      X     X
X   X+       X  X  X
X  +               X
X     X   + +      X
X         *       XX
XX    X           +X
XX   XX  X      X  X
X            X     X
XXXXXXXXXXXXXXXXXXXX
?7
XXXXXXXXXXXXXXXXXXXX
X X   X      X     X
X   X        X  X  X
X     +            X
X   +  X           X
X        + +      XX
XX    X  *       + X
XX   XX  X      X  X
X            X     X
XXXXXXXXXXXXXXXXXXXX
?7
XXXXXXXXXXXXXXXXXXXX
X X   X      X     X
X   X        X  X  X
X                  X
X      X           X
X     +           XX
XX    X + +        X
XX   XX *X      X  X
X            X     X
XXXXXXXXXXXXXXXXXXXX
?5
YOU HAVE DESTROYED ALL YOUR OPPONENTS-THE GAME IS YOURS
```

Why was "7" a good move here?

AND ANOTHER RUN

```
XXXXXXXXXXXXXXXXXXXX
X           XX X   X
X    X X     X     X
X      +           X
X X +X     +    +  X
X           X+     X
X       X    *     X
X          X       X
X                  X
XXXXXXXXXXXXXXXXXXXX
?8
XXXXXXXXXXXXXXXXXXXX
X           XX X   X
X    X X     X     X
X                  X
X X  X +           X
X    +     +X +    X
X       X  *+      X
X          X       X
X                  X
XXXXXXXXXXXXXXXXXXXX
?7
XXXXXXXXXXXXXXXXXXXX
X           XX X   X
X    X X     X     X
X                  X
X X  X             X
X       +   X      X
X     + X +   +    X
X         *X       X
X                  X
XXXXXXXXXXXXXXXXXXXX
?5
XXXXXXXXXXXXXXXXXXXX
X           XX X   X
X    X X     X     X
X                  X
X X  X             X
X           X      X
X       X+         X
X      +   X+      X
X          *       X
XXXXXXXXXXXXXXXXXXXX
?1
*** YOU HAVE BEEN DESTROYED BY A LUCKY COMPUTER ***
```

Would a move of "8" have been better?



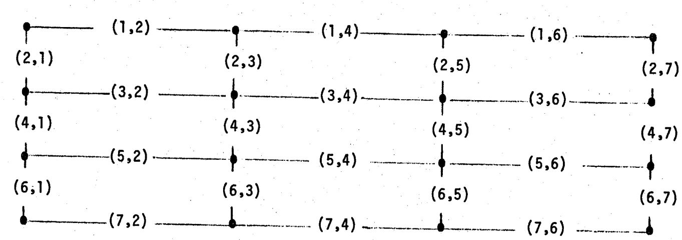
CONDOT LISTING

```
1 PRINT"THIS PROGRAM WIIL PLAY CONNFCT THE DOTS WITH YOU."
2 PRINT "THE GAME IS PLAYED ON A 4 X 4 ARRAY. WHEN"
3 PRINT"YOU WANT TO MAKE A MOVE YOU MUST TYPE IN"
4 PRINT"THE COORDINATES OF THE SPACF BETWEEN THE TWO DOTS YOU"
5 PRINT"WISH TO CONNECT. ENTER EACH OF YOUR MOVFS BY TYPING"
6 PRINT"THE ROW NUMBER, A COMMA AND THEN THE COIUMN NUMBER."
7 PRINT"THE UPPER LEFT HAND CORNER OF THE ARRAY IS 1,3."
8 PRINT"HERE WE GO."
20 DIM A(12,32)
30 V=0
40 FOR R=1 TO 12
50 FOR C=1 TO 12
60 IF R/2=INT(R/2) THEN 100
70 IF C/2=INT(C/2) THEN 100
80 A(R,C)=-50
90 GO TO 110
100 A(R,C)=0
110 NEXT C
120 NEXT R
130 IF V=1 THEN 200
200 GO SUB 1000
210 PRINT"YOUR MOVE";
220 INPUT X,Y
230 IF X=INT(X/1) THEN 260
240 PRINT"TYPING ERROR"
250 GO TO 210
260 IF (X-1)*(X-7)>0 THEN 240
265 X=X+2
270 IF (Y-1)*(Y-7)>0 THEN 240
272 Y=Y+2
280 IF (X+Y+1)/2<>INT((X+Y+1)/2) THEN 240
290 IF A(X,Y)<>0 THEN 240
300 A(X,Y)=50
310 IF X/2=INT(X/2) THEN 380
320 IF A(X-2,Y)+A(X-1,Y+1)+A(X-1,Y-1)<>150 THEN 350
330 P=1
340 A(X-1,Y)=1
350 IF A(X+2,Y)+A(X+1,Y+1)+A(X+1,Y-1)<>150 THEN 440
360 A(X+1,Y)=1
370 GO TO 200
380 IF A(X,Y-2)+A(X+1,Y-1)+A(X-1,Y-1)<>150 THEN 410
390 A(X,Y-1)=1
400 P=1
410 IF A(X,Y+2)+A(X+1,Y+1)+A(X-1,Y+1)<>150 THEN 440
420 A(X,Y+1)=1
430 GO TO 200
440 IF P=1 THEN 200
450 GO SUB 1000
460 PRINT"MY MOVE"
470 FOR R=4 TO 10 STEP 2
480 FOR C=4 TO 10 STEP 2
490 IF A(R-1,C)+A(R+1,C)+A(R,C-1)+A(R,C+1)<>150 THEN 680
500 A(R,C)=-1
510 IF A(R-1,C)<>0 THEN 550
520 A(R-1,C)=50
530 IF A(R-3,C)+A(R-2,C-1)+A(R-2,C+1)<>150 THEN 450
540 LET A(R-2,C)=-1
550 IF A(R+1,C)<>0 THEN 590
560 A(R+1,C)=50
570 IF A(R+3,C)+A(R+2,C-1)+A(R+2,C+1)<>150 THEN 450
580 A(R+2,C)=-1
590 IF A(R,C-1)<>0 THEN 630
600 A(R,C-1)=50
610 IF A(R,C-3)+A(R-1,C-2)+A(R+1,C+2)<>150 THEN 450
620 A(R,C-2)=-1
630 IF A(R,C+1)<>0 THEN 450
640 A(R,C+1)=50
650 IF A(R,C+3)+A(R-1,C+2)+A(R+1,C+2)<>150 THEN 450
660 A(R,C+2)=-1
670 GO TO 450
680 NEXT C
690 NEXT R
692 IF E>1 THEN 730
700 IF A(12-X,32-Y)<>0 THEN 730
710 A(12-X,32-Y)=50
712 IF E>1 THEN 870
720 GO TO 200
730 FOR R=3 TO 9
740 FOR C=3 TO 9
750 IF (R+C)/2=INT((R+C)/2) THEN 850
760 IF A(R,C)<>0 THEN 850
780 IF R/2=INT(R/2) THEN 830
790 IF A(R-2,C)+A(R-1,C-1)+A(R-1,C+1)=100 THEN 850
800 IF A(R+2,C)+A(R+1,C-1)+A(R+1,C+1)=100 THEN 850
810 A(R,C)=50
820 GO TO 200
830 IF A(R,C-2)+A(R-1,C-1)+A(R+1,C-1)=100 THEN 850
840 IF A(R,C+2)+A(R-1,C+2)+A(R+1,C+1)<>100 THEN 810
850 NEXT C
860 NEXT R
862 IF E>1 THEN 700
870 R=INT(RND(R)*7)+3
880 C=INT(RND(C)*7)+3
881 IF R/2=INT(R/2) THEN 885
882 IF C/2=INT(C/2) THEN 900
883 GO TO 870
885 IF C/2<>INT(C/2) THEN 900
886 GO TO 870
900 IF A(R,C)<>0 THEN 870
910 A(R,C)=50
920 GO TO 200
930 PRINT"DO YOU WANT TO PLAY AGAIN(TYPE 1 FOR YES OR 2 FOR NO)";
940 INPUT B
950 IF B=1 THEN 40
960 STOP
1000 P=0
1010 D=0
1020 E=0
1030 FOR R=3 TO 9
1040 FOR C=3 TO 9
1050 IF A(R,C)<>0 THEN 1080
1060 PRINT"   ";
1070 GO TO 1240
1080 IF A(R,C)<>-50 THEN 1110
1090 PRINT" . ";
1100 GO TO 1240
1110 IF A(R,C)<>-1 THEN 1140
1120 PRINT" C ";
1130 GO TO 1170
1140 IF A(R,C)<>1 THEN 1200
1150 PRINT" H ";
1170 D=D+A(R,C)
1180 E=E+1
1190 GO TO 1240
1200 IF R/2=INT(R/2) THEN 1230
1210 PRINT" - ";
1220 GO TO 1240
1230 PRINT " : ";
1240 NEXT C
1245 PRINT
1250 NEXT R
1260 IF E>=9 THEN 1280
1270 RETURN
1280 IF D>0 THEN 1310
1290 PRINT "I WON"
1300 GO TO 930
1310 PRINT 'YOU WON'
1320 GO TO 930
1400 END
```

Can you figure out how to make the grid larger or smaller ?

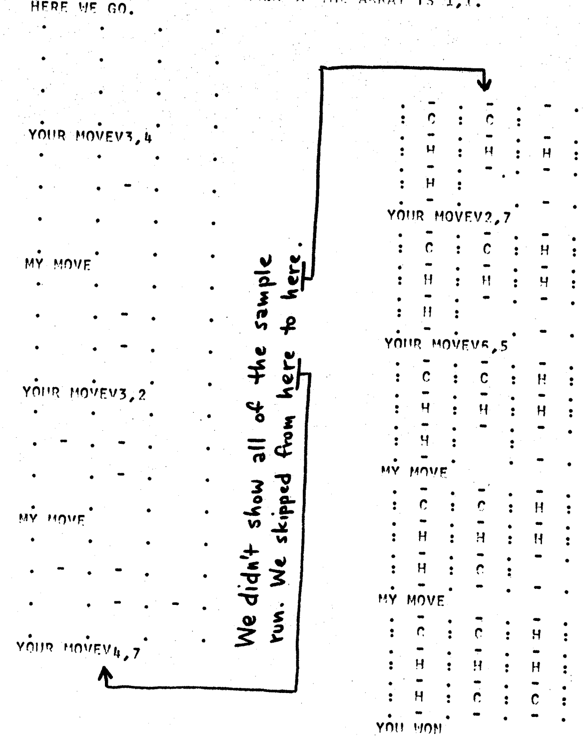
GRID

Here's a sample of the grid showing the Coordinates for each posible move



SAMPLE RUN

```
THIS PROGRAM WILL PLAY CONNECT THE DOTS WITH YOU.
THE GAME IS PLAYED ON A 4 X 4 ARRAY. WHEN
YOU WANT TO MAKE A MOVE YOU MUST TYPE IN
THE COORDINATES OF THE SPACE BETWEEN THE TWO DOTS YOU
WISH TO CONNECT. ENTER EACH OF YOUR MOVES BY TYPING
THE ROW NUMBER, A COMMA AND THEN THE COLUMN NUMBER.
THE UPPER LEFT HAND CORNER OF THE ARRAY IS 1,1.
HERE WE GO.
```

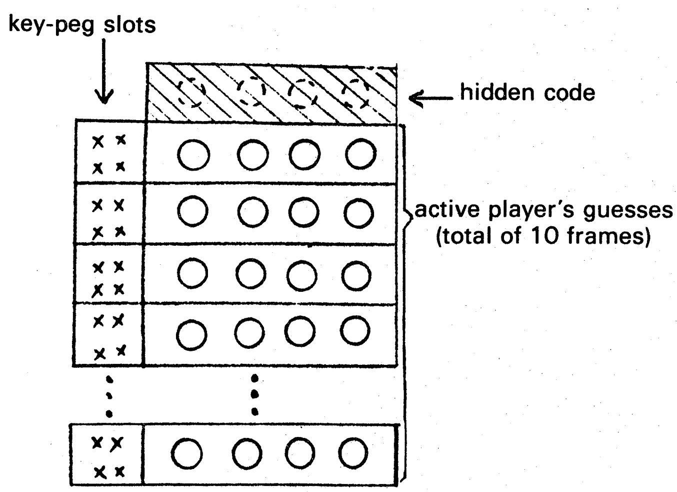


MASTERMIND

by David G. Struble
University of Dayton

The original invention of Mastermind is credited to an amateur mathematician, Mordechai Meirovich, who first displayed it at the 1971 Nurenburg Toy Fair.* Rights to the game were bought by Invicta who had moderate success with the game for 2½ years until the Christmas season of 1975 when it was the most popular packaged game. Sales surpassed even the old standby, Monopoly.

In its most basic form, Mastermind consists of a plastic game board, a dozen or so pegs which can be grouped into six basic colors, and two groups of black and white key pegs (sometimes called "inference pegs".) The game board resembles the figure below.



The game is played by two people, whom we shall designate as the "active" player and the "passive" player. The first step before play actually commences is to have the passive player (in our case, the computer) choose a total of four colored pegs at random from any of the six basic color groups (duplicate colors allowed, of course.) He then conceals these colors from the active player by placing the four pegs in the "hidden code" portion of the game board. It is now up to the active player to determine, in ten moves or less, the exact color and location of each of the four pegs comprising the hidden code.

To aid the active player in determining the hidden code, the passive player must award the active player a number of key pegs (inference pegs) after each guess, according to the following scheme: for *each* peg in the active player's current guess which corresponds exactly (in color *and* position) to a peg in the hidden code, the passive person places one *black* peg in the key-peg square adjacent to the passive player's current guess frame. Placing of the key pegs within the square is arbitrary since the relative position of the key peg carries no meaning. Clearly, when four black pegs are obtained, the hidden code is broken.

Secondly, the passive player must place one *white* key peg in the current key-peg square for *each* peg in the active player's current guess which matches (in color, but *not* position) a peg in the hidden code. Keep in mind that once a color peg in the player's current guess has been awarded a key peg, its function in determining the remaining number of key pegs to award for the current guess is finished. For example, suppose the hidden code were:

R B Y G

corresponding to red, blue, yellow, green, and the active player's current guess were:

G B B P

corresponding to green, blue, blue and purple.

The passive player should subsequently award one black and one white key peg for the following reasons: the blue color peg in position 2 of the current guess matches exactly in color and position with the hidden code. Secondly, the green color peg in position 1 of the current guess matches the color of the peg in position 4 of the hidden code. But since the *location* of the green peg is not exact, only a white peg is awarded. The blue and purple pegs in positions 3 and 4, respectively, of the current guess do not match either the color or position of the remaining pegs in the hidden code (positions 1 and 3) and hence, no other key pegs are awarded.

The game proceeds in this manner until the hidden code is broken or all ten frames have been filled. As noted earlier, the computer will play the passive player in our computer version, generating a hidden code and awarding the black and white key pegs after each guess.

The program offers the user two options, QUIT and BOARD, which may be entered at any time *after* the first move. QUIT instructs the program that you are fed up with playing Mastermind for the time being and wish to terminate the session. BOARD instructs the program to print out a summary of the moves prior to the time that the BOARD command was issued, including the guesses and key pegs awarded for each frame. Some players find that an arrangement of frames such as that provided by BOARD is easier to visualize and subsequently analyze. Beginners will find it most useful.

The program as listed will run on a Univac 70/7 under VMOS or equivalent Univac series 70 machine.

One last item. Any suggestions on how to get the machine to play the active person?

---

*Ed. Note—
To anyone familiar with children's games, it is obvious that Mastermind is simply a commercial adaptation (using colors rather than numbers) of the game Bulls and Cows. This game, much more popular in England than the U.S. is not, to my knowledge, commercially packaged although it is available in a computer (BASIC) version. This is the game BULCOW by Geoff Wyvill which appeared in *101 BASIC Computer Games* ($7.50 from *Creative Computing* Library, 42 Pleasant St., Newburyport, MA 01950). Geoff's computer version is especially intriguing since it plays two games simultaneously, the active player in one and the passive player in the other.
—DHA.

MASTERMIND

PROGRAM LISTING

```
100 PRINT
110 PRINT "THE GAME OF MASTERMIND"
120 PRINT
130 PRINT "COLOR CODES:"
140 PRINT "                    R = RED       O = ORANGE       Y = YELLOW"
150 PRINT "                    G = GREEN     B = BLUE         P = PURPLE"
160 PRINT
170 DIM B$(10),Y(10),Z(10)
180 C(0)=4
190 FOR N=1 TO 4
200 C(N)=INT(6*RND(-1)+1)
210 NEXT N
220 FOR N=1 TO 4
230 X=C(N)
240 GOSUB 730
250 C(N)=X
260 NEXT N
270 CHANGE C TO P$
280 FOR P=1 TO 10
290 PRINT
300 PRINT "MOVE NUMBER";P;
310 INPUT G$
320 IF G$="BOARD" THEN 910
330 IF G$="QUIT" THEN 440
340 B$(P)=G$
350 GOSUB 520
360 IF B=4 THEN 1010
370 GOSUB 600
380 PRINT B;" BLACK PEGS"
390 Y(P)=B
400 PRINT W;" WHITE PEGS"
410 Z(P)=W
420 NEXT P
430 PRINT "SORRY, YOU LOSE"
440 PRINT "CORRECT CODE WAS: ";P$
450 PRINT
460 PRINT "WANT TO PLAY AGAIN";
470 INPUT A$
480 IF A$="YES" THEN 190
490 PRINT
500 END
510 REM COMPUTE BLACK PEGS
520 CHANGE G$ TO G
530 B=0
540 FOR K=1 TO 4
550 IF G(K)<>C(K) THEN 570
560 B=B+1
570 NEXT K
580 RETURN
590 REM COMPUTE WHITE PEGS
600 CHANGE P$ TO R
610 W=0
620 FOR I=1 TO 4
630 FOR J=1 TO 4
640 IF G(I)<>R(J) THEN 680
650 W=W+1
660 R(J)=0
670 GOTO 690
680 NEXT J
690 NEXT I
700 W=W-B
710 RETURN
720 REM TRANSLATE COLOR CODES TO NUMERICS
730 IF X<>1 THEN 760
740 X=217
750 RETURN
760 IF X<>2 THEN 790
770 X=214
780 RETURN
790 IF X<>3 THEN 820
800 X=238
810 RETURN
820 IF X<>4 THEN 850
830 X=199
840 RETURN
850 IF X<>5 THEN 880
860 X=194
870 RETURN
880 X=215
890 RETURN
900 REM PRINT BOARD SUMMARY
910 V=P-1
920 PRINT USING 930
930 :GUESS   BLACKS   WHITES
940 PRINT USING 950
950 :-----   ------   ------
960 FOR I=1 TO V
970 PRINT USING 980,B$(I),Y(I),Z(I)
980 : ####      #        #
990 NEXT I
1000 GOTO 290
1010 PRINT "YOU WIN!"
1020 GOTO 450
```

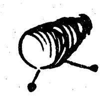
SAMPLE RUN

```
THE GAME ØF MASTERMIND

CØLØR CØDES:
                    R = RED       Ø = ØRANGE      Y = YELLØW
                    G = GREEN     B = BLUE        P = PURPLE

MØVE NUMBER 1   ?BRBR
 1  BLACK PEGS
 1  WHITE PEGS

MØVE NUMBER 2   ?PØPØ
 1  BLACK PEGS
 0  WHITE PEGS

MØVE NUMBER 3   ?YGYG
 1  BLACK PEGS
 0  WHITE PEGS

MØVE NUMBER 4   ?BRPY
 0  BLACK PEGS
 2  WHITE PEGS

MØVE NUMBER 5   ?RGBP
 2  BLACK PEGS
 1  WHITE PEGS

MØVE NUMBER 6   ?BØARD
GUESS    BLACKS    WHITES
-----    ------    ------
 BRBR       1         1
 PØPØ       1         0
 YGYG       1         0
 BRPY       0         2
 RGBP       2         1

MØVE NUMBER 6   ?RGØB
 1  BLACK PEGS
 3  WHITE PEGS

MØVE NUMBER 7    ?RØGB
 2  BLACK PEGS
 2  WHITE PEGS

MØVE NUMBER 8   ?RØBG
YØU WIN!

WANT TØ PLAY AGAIN?YES
```

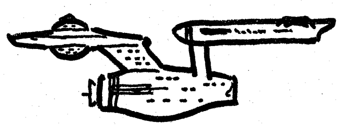
One playing strategy is to guess pairs of all 6 colors on the first 3 moves. Is this the best strategy? Why or why not?

Young woman or old — What do you see?

Another new game from Creative Computing . . . .



DEEPSPACE



Author: Unknown
Modified by: Bill Cotter, Pittsfield, Mass.
Language: BASIC (Honeywell 600/6000)
Description: DEEPSPCE is another version of a space battle. You become the commander of either a scout ship, cruiser, or battleship. You then pick the weapons, and planetary system to patrol, and it's time to do battle.

The closer you get to the enemy, the better your chance of destroying him. Unfortunately, his chance of destroying you also improves. If you get too close, you can damage yourself; when a vessel's damage rating reaches or exceeds 100, it's destroyed.

Suggestion: Change the time between reports—this will shorten the game by allowing you to get closer faster. Also, for a truly random game, Honeywell users should change RND(0) to RND(—1).

PROGRAM LISTING

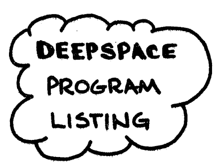
```
100 PRINT
110 PRINT
120 PRINT "DEEPSPACE    ";DAT$
130 PRINT
140 PRINT
150 PRINT "THIS IS DEEPSPACE, A TACTICAL SIMULATION OF SHIP-TO SHIP"
160 PRINT "COMBAT IN DEEP SPACE."
170 PRINT "DO YOU WISH INSTRUCTIONS (YES OR NO)";\INPUT I$
180 IF I$="YES" THEN 200
190 GO TO 610
200 PRINT "YOU ARE ONE OF A GROUP OF CAPTAINS ASSIGNED TO PATROL A"
210 PRINT "SECTION OF YOUR STAR EMPIRE'S BORDER AGAINST HOSTILE"
220 PRINT "ALIENS. ALL YOUR ENCOUNTERS HERE WILL BE AGAINST HOSTILE"
230 PRINT "VESSELS. YOU WILL FIRST BE REQUIRED TO SELECT A VESSEL"
240 PRINT "FROM ONE OF THREE TYPES, EACH WITH ITS OWN CHARACTERISTICS:"
250 PRINT
260 PRINT "TYPE","SPEED","CARGO SPACE","PROTECTION"
270 PRINT "1 SCOUT","10X","16","1"
280 PRINT "2 CRUISER","4X","24","2"
290 PRINT "3 BATTLESHIP","2X","30","5"
300 PRINT
310 PRINT "SPEED IS GIVEN RELATIVE TO THE OTHER SHIPS."
320 PRINT "CARGO SPACE IS IN UNITS OF SPACE ABOARD SHIP WHICH CAN BE"
330 PRINT "FILLED WITH WEAPONS."
340 PRINT" PROTECTION IS THE RELATIVE STRENGTH OF THE SHIP'S ARMOR"
350 PRINT "AND FORCE FIELDS"
360 PRINT
370 PRINT "ONCE A SHIP HAS BEEN SELECTED, YOU WILL BE INSTRUCTED TO ARM"
380 PRINT "IT WITH WEAPONRY FROM THE FOLLOWING LIST:"
390 PRINT
400 PRINT "TYPE                          CARGO SPACE    REL. STRENGTH"
410 PRINT "1 PHASER BANKS                    12              4"
420 PRINT "2 ANTI-MATTER MISSILE             4              20"
430 PRINT "3 HYPERSPACE LANCE                4              16"
440 PRINT "4 PHOTON TORPEDO                  2              10"
450 PRINT "5 HYPERON NEUTRALIZATION FIELD   20               6"
460 PRINT
470 PRINT"WEAPONS #1 & #5 CAN BE FIRED 100 TIMES EACH: ALL OTHERS CAN"
480 PRINT "BE FIRED ONCE FOR EACH ON BOARD."
490 PRINT "A TYPICAL LOAD FOR A CRUISER MIGHT CONSIST OF:"
500 PRINT "                    1-#1 PHASER BANK          =12"
510 PRINT "                    2-#3 HYPERSPACE LANCES    =8"
520 PRINT "                    2-#4 PHOTON TORPEDOES     =4"
530 PRINT "                                          ----------"
540 PRINT "                                       24 UNITS OF CARGO"
550 PRINT " A WORD OF CAUTION:FIRING HIGH YIELD WEAPONS AT CLOSE (< 100)"
560 PRINT "RANGE CAN BE DANGEROUS TO YOUR SHIP AND MINIMAL DAMAGE CAN"
570 PRINT "OCCUR AS FAR OUT AS 200 IN SOME CIRCUMSTANCES."
580 PRINT
590 PRINT "RANGE IS GIVEN IN THOUSANDS OF KILOMETERS."
```

LISTING continues on next page.

DEEPSPACE 05/29/75 **SAMPLE RUN**

```
THIS IS DEEPSPACE, A TACTICAL SIMULATION OF SHIP-TO SHIP
COMBAT IN DEEP SPACE.
DO YOU WISH INSTRUCTIONS (YES OR NO) ?NO
DO YOU WISH A MANEUVER CHART ?YES
                        ************
    MANEUVER CHART

1       FIRE PHASERS
2       FIRE ANTI-MATTER MISSILE
3       FIRE HYPERSPACE LANCE
4       FIRE PHOTON TORPEDO
5       ACTIVATE HYPERON NEUTRALIZATION FIELD
6       SELF-DESTRUCT
7       CHANGE VELOCITY
8       DISENGAGE
9       PROCEED

YOU HAVE A CHOICE OF THREE SYSTEMS TO PATROL.
1 ORION
2 DENEB
3 ARCTURUS
SELECT A SYSTEM(1-3) ?3
WHICH SPACECRAFT WOULD YOU LIKE.(1-3) ?2
YOU HAVE     24 UNITS OF CARGO SPACE TO FILL WITH WEAPONRY.
CHOOSE A WEAPON AND THE AMOUNT YOU WISH. ?1,1
YOU HAVE     12 UNITS OF CARGO SPACE TO FILL WITH WEAPONRY.
CHOOSE A WEAPON AND THE AMOUNT YOU WISH. ?2,1
YOU HAVE      8 UNITS OF CARGO SPACE TO FILL WITH WEAPONRY.
CHOOSE A WEAPON AND THE AMOUNT YOU WISH. ?3,1
YOU HAVE      4 UNITS OF CARGO SPACE TO FILL WITH WEAPONRY.
CHOOSE A WEAPON AND THE AMOUNT YOU WISH. ?4,2

RANGE TO TARGET:    658.3301
RELATIVE VELOCITY:    3.154701
ACTION ?9

RANGE TO TARGET:    599.0996
RELATIVE VELOCITY:    3.154701
ACTION ?9

RANGE TO TARGET:    539.8691
RELATIVE VELOCITY:    3.154701
ACTION ?7
CHANGE TO BE EFFECTED ?2
CHANGE BEYOND MAXIMUM POSSIBLE
INCREASING TO MAXIMUM

RANGE TO TARGET:    460.1757
RELATIVE VELOCITY:    4
ACTION ?9
DAMAGE CONTROL REPORTS YOUR VESSEL DAMAGE AT:    5.329875

RANGE TO TARGET:    380.4822
RELATIVE VELOCITY:    4
ACTION ?1
SCANNERS REPORT ENEMY DAMAGE NOW:    1.260274
DAMAGE CONTROL REPORTS YOUR VESSEL DAMAGE AT:    0.74178

RANGE TO TARGET:    300.7888
RELATIVE VELOCITY:    4
ACTION ?2
SCANNERS REPORT ENEMY DAMAGE NOW:    40.81256
DAMAGE CONTROL REPORTS YOUR VESSEL DAMAGE AT:    15.35915

RANGE TO TARGET:    221.0953
RELATIVE VELOCITY:    4
ACTION ?1
SCANNERS REPORT ENEMY DAMAGE NOW:    43.63048
DAMAGE CONTROL REPORTS YOUR VESSEL DAMAGE AT:   45.269

RANGE TO TARGET:    141.4019
RELATIVE VELOCITY:    4
ACTION ?3
SCANNERS REPORT ENEMY DAMAGE NOW:    144.1723
ENEMY VESSEL DESTROYED
DAMAGE CONTROL REPORTS YOUR VESSEL DAMAGE AT:    140.0901
YOUR VESSEL HAS BEEN DESTROYED
 ANOTHER BATTLE ?NO
TRY AGAIN LATER!
```

```
600 GO TO 640
610 PRINT "DO YOU WISH A MANEUVER CHART";\INPUT M$
620IF M$="YES" THEN 640
630 GO TO 770
640 PRINT "                        **************"
650 PRINT "    MANEUVER CHART  "
660 PRINT
670 PRINT " 1      FIRE PHASERS"
680 PRINT " 2      FIRE ANTI-MATTER MISSILE"
690 PRINT " 3      FIRE HYPERSPACE LANCE"
700 PRINT " 4      FIRE PHOTON TORPEDO"
710 PRINT " 5      ACTIVATE HYPERON NEUTRALIZATION FIELD"
720 PRINT " 6      SELF-DESTRUCT"
730 PRINT " 7      CHANGE VELOCITY"
740 PRINT " 8      DISENGAGE"
750 PRINT " 9      PROCEED"
760 PRINT
770 PRINT"YOU HAVE A CHOICE OF THREE SYSTEMS TO PATROL."
780 PRINT"1 ORION"
790 PRINT"2 DENEB"
800 PRINT"3 ARCTURUS"
810 PRINT "SELECT A SYSTEM(1-3)";\INPUT S9
820 IF S9=1 THEN 2380
830 IF S9=2 THEN 2430
840 GO TO 2480
850 D0=0
860 D1=0
870 N1=0
880 N2=0
890 N3=0
900 N4=0
910 D=0
920 PRINT"WHICH SPACECRAFT WOULD YOU LIKE.(1-3)";\INPUT S
930 IF S=1 THEN 1790
940 IF S=2 THEN 1830
950 IF S=3 THEN 1870
960 GO TO 920
970 C=C0
980 PRINT"YOU HAVE ";C;"UNITS OF CARGO SPACE TO FILL WITH WEAPONRY."
990 PRINT"CHOOSE A WEAPON AND THE AMOUNT YOU WISH.";\INPUT W,N
1000 IF W=1 THEN 1910
1010 IF W=2 THEN 2010
1020 IF W=3 THEN 2100
1030 IF W=4 THEN 2190
1040 IF W=5 THEN 2280
1050 GO TO 980
1060 IF N*C1>C THEN 2530
1070 C=C-N*C1
1080 IF W=1 THEN 1990
1090 IF W=2 THEN 2080
1100 IF W=3 THEN 2170
1110 IF W=4 THEN 2260
1120 GO TO 2360
1130 IF C>1 THEN 980
1140 REM
1150 S1=S0*RND(0)
1160 R=(3*RND(0) +5)*100
1170 PRINT
1180 PRINT "RANGE TO TARGET:";R
1190 PRINT "RELATIVE VELOCITY:";S1
1200 PRINT"ACTION";\INPUT M
1210 IF M=1 THEN 1940
1220 IF M=2 THEN 2030
1230 IF M=3 THEN 2120
1240 IF M=4 THEN 2210
1250 IF M=5 THEN 2310
1260 IF M=6 THEN 1660
1270 IF M=7 THEN 1390
1280 IF M=8 THEN 2760
1290 IF R<500 THEN 1500
1300 IF S1>0 THEN 1330
1310 R=R+(S1*8.3)^1.25
1320 GO TO 1340
1330 R=R-(S1*8.3)^1.25
1340 IF R>1500 THEN 2590
1350 IF R>0 THEN 1370
1360 R=-R
1370 PRINT
1380 GO TO 1180
1390PRINT"CHANGE TO BE EFFECTED";\INPUT S2
1400 IF (S1+S2)>S0 THEN 2550
1410 S1=S1+S2
1420 GO TO 1180
1430 F0=P1*(Z/R)^1.5
1440 REM
1450 D0=(2*F0+3*F0*RND(0))/5
1460D=D+D0
1470 PRINT"SCANNERS REPORT ENEMY DAMAGE NOW:";D
1480 IF D>99 THEN 2720
1490 GO TO 1510
1500 D0=0
1510 REM
1520 K=E1+E2*RND(0)
1530 REM
1540 E=E3+E4*RND(0)+5/P0*RND(0)
1550 REM
1560 F3=E*(K/R)^1.85
1570 D2=(3*F3+3*F3*RND(0))/5.5
1580 D1=D1+D2
1590 IF (Z*D0)/(R*500)>2.2 THEN 1620
1600 D3=D0*2/(R^2*P0)
1610 D1=D1+D3
1620 PRINT"DAMAGE CONTROL REPORTS YOUR VESSEL DAMAGE AT:";D1
1630 IF D1>99 THEN 2740
1640 IF D>99 THEN 2760
1650 GO TO 1300
1660 PRINT "SELF DESTRUCT FAILSAFE ACTIVATED!!"
1670 PRINT"INPUT 1 TO RELEASE FAILSAFE"\INPUT U
1680 IF U=1 THEN 1700
1690 GO TO 1290
```




```
1700 PRINT "SELF DESTRUCT ACCOMPLISHED"
1710 IF R>60 THEN 1740
1720 PRINT"ENEMY VESSEL ALSO DESTROYED"
1730 GO TO 2760
1740 D4=3200/R
1750 D=D+D4
1760 IF D>99 THEN 1720
1770 PRINT"ENEMY VESSEL SURVIVES WITH";D;"DAMAGE"
1780 GO TO 2760
1790 S0=10
1800 C0=16
1810 P0=1
1820 GO TO 970
1830 S0=4
1840 C0=24
1850 P0=2
1860 GO TO 970
1870 S0=2
1880C0=30
1890 P0=5
1900 GO TO 970
1910 C1=.12
1920N=100
1930 GO TO 1060
1940 P1=4
1950 IF N1=0 THEN 2610
1960 N1=N1-1
1970 Z=200
1980 GO TO 1430
1990 N1=N1+N
2000 GO TO 1130
2010 C1=4
2020 GO TO 1060
2030 P1=20
2040 IF N2=0THEN 2640
2050 N2=N2-1
2060 Z=500
2070 GO TO 1430
2080 N2=N2+N
2090 GO TO 1130
2100 C1=4
2110 GO TO 1060
2120 P1=16
2130 IF N3=0 THEN 2660
2140 N3=N3-1
2150 Z=550
2160 GO TO 1430
2170 N3=N3+N
2180 GO TO 1130
2190 C1=2
2200 GO TO 1060
2210 P1=10
2220 IF N4=0 THEN 2680
2230 N4=N4-1
2240 Z=400
2250 GO TO 1430
2260 N4=N4+N
2270 GO TO 1130
2280 C1=.20
2290 N=100
2300 GO TO 1060
2310 P1=6
2320 IF N5=0 THEN 2700
2330 N5=N5-1
2340 Z=250
2350 GO TO 1430
2360 N5=N5+N
2370 GO TO 1130
2380 E1=150
2390 E2=500
2400 E3=3
2410 E4=4
2420 GO TO 850
2430 E1=200
2440 E2=350
2450 E3=4
2460E4=3
2470 GO TO 850
2480 E1=150
2490 E2=400
2500 E3=5
2510 E4=2
2520 GO TO 850
2530 PRINT"NOT ENOUGH SPACE. RESELECT"
2540 GO TO 980
2550 PRINT"CHANGE BEYOND MAXIMUM POSSIBLE"
2560 PRINT"INCREASING TO MAXIMUM"
2570 S1=S0
2580 GO TO 1300
2590 PRINT"OUT OF SENSOR RANGE.AUTOMATIC DISENGAGE."
2600 GO TO 2760
2610 PRINT"PHASER BANKS DRAINED"
2620 PRINT"SELECT ANOTHER COURSE OF ACTION"
2630 GO TO 1200
2640 PRINT" ALL ANTI-MATTER MISSLES EXPENDED"
2650 GO TO 2620
2660 PRINT"ALL HYPERSPACE LANCES EXPENDED"
2670 GO TO 2620
2680 PRINT "ALL PHOTON TORPEDO TUBES EMPTY"
2690 GO TO 2620
2700 PRINT "HYPERON NEUTRALIZATION FIELD DRAINED"
2710 GO TO 2620
2720 PRINT "ENEMY VESSEL DESTROYED"
2730 GO TO 1510
2740 PRINT"YOUR VESSEL HAS BEEN DESTROYED"
2750 GO TO 2760
2760 PRINT" ANOTHER BATTLE";\INPUT R$
2770 IF R$="YES" THEN 810
2780 PRINT"TRY AGAIN LATER!"
2790 END
```

Another new game from Creative Computing . . . .

BOB-STONES

Dohn Addleman
Pennsylvania Dept. of Education

The idea for this new number game was derived from a contest called "Bobstones" as described in the novel, *Watership Down.* Playing rules and description are in the remarks portion of the program listing.

Suggestions:

A. Play the game against the computer.

B. Once you are familiar with the rules, play the game with a friend and use real dice.

C. Incorporate the following modifications into the program:

1. Update the program so that the players can vary the number of points needed to win.
2. Update the program so that if a tie results, the winner must win by two points.
3. Update the program so that it will verify that the human is recording the computer's responses correctly.
4. Update the program to record the most recent sums of the dice rolls. Then, the computer can use this information to determine the probability of the next roll being odd or even.

PROGRAM LISTING

```
1 REM     THIS IS A NUMBER GAME CALLED 'BOBSTONES'.  THE IDEA FOR THIS
2 REM  GAME DERIVED FROM A CONTEST CALLED 'BOBSTONES' DESCRIBED IN THE
3 REM  NOVEL "WATERSHIP DOWN".
4 REM     THIS IS THE LISTING OF THE BASIC LANGUAGE SOURCE CODE.  THE
5 REM  LISTING CAN BE DIVIDED INTO FOUR SECTIONS.  LINES 100 THRU 640
6 REM  PERTAIN TO GAME INSTRUCTIONS, DATA VARIABLE INITIALIZATION,
7 REM  PRINTING THE SCORE AND DETERMINATION OF THE PLAYERS' TURNS.
8 REM  LINES 650 THRU 1150 PERTAIN TO THE HUMAN PLAYER.  LINES
9 REM  1160 THRU 1890 PERTAIN TO THE COMPUTER AS A PLAYER.  LINES
10 REM 1900 THRU 2050 DEAL WITH RESOLUTION OF TIES AND DETERMINATION
11 REM OF THE WINNER.
12 REM
13 REM
14 REM FIRST SECTION - LINES 100 THRU 640
15 REM   1.  INSTRUCTIONS
16 REM   2.  INITIALIZATION
17 REM   3.  WHO GOES FIRST ?
18 REM   4.  COMPUTER SIMULATES ROLL OF DICE
19 REM   5.  PRINT GAME SCORE ?
20 REM   6.  IS GAME POSSIBLY OVER ?
21 REM
22 REM
23 REM SECOND SECTION - LINES 650 THRU 1150
24 REM   1.  HUMAN GUESSES IF SUM OF DICE IS ODD OR EVEN
25 REM   2.  IS RESPONSE APPROPRIATE ?
26 REM   3.  COMPUTER TELLS HUMAN IF RIGHT OR WRONG AND ADJUSTS SCORE
27 REM       IF CORRECT GUESS
28 REM   4.  HUMAN GUESSES SUM OF DICE
29 REM   5.  IS RESPONSE APPROPRIATE ?
30 REM   6.  COMPUTER TELLS HUMAN IF RIGHT OR WRONG AND ADJUSTS SCORE
31 REM       IF CORRECT GUESS
32 REM   7.  IS RESPONSE APPROPRIATE ?
33 REM   8.  COMPUTER TELLS HUMAN IF RIGHT OR WRONG AND ADJUSTS SCORE
34 REM       IF CORRECT GUESS
35 REM   9.  GO TO NEXT ROLL OF DICE
```

LISTING continues on next page

SAMPLE RUN

```
*RUN

   THIS IS A NUMBER GAME CALLED BOBSTONES.  THE OBJECT OF
BOBSTONES IS TO GUESS THREE THINGS ABOUT THE ROLL OF A PAIR
OF DICE.  ON EACH TURN, THE COMPUTER SIMULATES THE ROLL OF
THE DICE.  THEN, YOU OR THE COMPUTER (YOUR OPPONENT) GUESS

                                                       SCORE
 1. IF THE SUM OF THE DICE IS ODD OR EVEN              1 POINT
 2. THE SUM OF THE DICE                                2 POINTS
 3. THE NUMBER ON EACH OF THE TWO DICE                 3 POINTS

   THE WINNER IS THE FIRST PLAYER TO SCORE 11 POINTS.  IF A
TIE RESULTS, THE WINNER IS THE FIRST PLAYER TO BREAK THE TIE.
   GOOD LUCK !


YOU FIRST OR ME
?ME

YOUR TURN.
IS THE SUM ODD OR EVEN
?ODD
YOU ARE CORRECT.
NOW, GUESS THE SUM
?7
SORRY, THE SUM IS  5 .

MY TURN.
*** ON THIS ROLL OF THE DICE, THE TWO NUMBERS ARE  2   AND  2  .
*** THE SUM IS  4 .
MY GUESS IS THAT THE SUM IS EVEN.
AM I RIGHT OR WRONG
?RIGHT
MY GUESS OF THE SUM IS  2
AM I RIGHT OR WRONG
?WRONG

THE SCORE IS ME 1  - YOU 1 .

YOUR TURN.
IS THE SUM ODD OR EVEN
?ODD
YOU ARE CORRECT.
NOW, GUESS THE SUM
?7
SORRY, THE SUM IS  11

MY TURN.
*** ON THIS ROLL OF THE DICE, THE TWO NUMBERS ARE  6   AND  6  .
*** THE SUM IS  12  .
MY GUESS IS THAT THE SUM IS EVEN.
AM I RIGHT OR WRONG
?RIGHT
MY GUESS OF THE SUM IS  12
AM I RIGHT OR WRONG
?RIGHT
MY GUESS IS THAT THE NUMBERS ARE  6  AND  6  .
AM I RIGHT OR WRONG
?RIGHT

THE SCORE IS ME 7  - YOU 2 .

YOUR TURN.
IS THE SUM ODD OR EVEN
?ODD
YOU ARE CORRECT.
NOW, GUESS THE SUM
?9,
 INVALID DATUM
 ?9
SORRY, THE SUM IS  7 .

MY TURN.
*** ON THIS ROLL OF THE DICE, THE TWO NUMBERS ARE  1   AND  4  .
*** THE SUM IS  5 .
MY GUESS IS THAT THE SUM IS ODD.
AM I RIGHT OR WRONG
?RIGHT
MY GUESS OF THE SUM IS  7
AM I RIGHT OR WRONG
?WRONG

THE SCORE IS ME 8  - YOU 3 .

YOUR TURN.
IS THE SUM ODD OR EVEN
?EVEN
SORRY, THE SUM IS 3  .

MY TURN.
*** ON THIS ROLL OF THE DICE, THE TWO NUMBERS ARE  5   AND  6  .
*** THE SUM IS  11   .
MY GUESS IS THAT THE SUM IS ODD.
AM I RIGHT OR WRONG
?RIGHT
MY GUESS OF THE SUM IS  11
AM I RIGHT OR WRONG
?RIGHT
MY GUESS IS THAT THE NUMBERS ARE  5  AND  6  .
AM I RIGHT OR WRONG
?RIGHT

THE SCORE IS ME 14    - YOU 3 .

I WIN! ANOTHER GAME?
?NO
SEE YOU LATER.
 *
```

BOBSTONES
PROGRAM LISTING

```
36 REM
37 REM
38 REM   THIRD SECTION - LINES 1160 THRU 1890
39 REM    1.   COMPUTER TELLS HUMAN THE NUMBERS OF THE ROLL OF THE DICE
40 REM          AND THE SUM
41 REM    2.   COMPUTER (AS PLAYER) GUESSES IF SUM ODD OR EVEN
42 REM    3.   COMPUTER ASKS HUMAN IF ITS GUESS RIGHT OR WRONG
43 REM    4.   HUMAN RESPONDS AND SCORE IS ADJUSTED IF COMPUTER GUESSED
44 REM          CORRECTLY
45 REM    5.   COMPUTER GUESSES THE SUM OF THE DICE.
46 REM    6.   COMPUTER ASKS HUMAN IF ITS GUESS RIGHT OR WRONG
47 REM    7.   HUMAN RESPONDS AND SCORE IS ADJUSTED IF COMPUTER GUESSED
48 REM          CORRECTLY
49 REM    8.   COMPUTER GUESSES OR KNOWS THE NUMBERS FOR THIS ROLL OF THE
50 REM          DICE
51 REM    9.   COMPUTER ASKS HUMAN IF ITS GUESS RIGHT OR WRONG
52 REM   10.   HUMAN RESPONDS AND SCORE IS ADJUSTED IF COMPUTER GUESSED
53 REM          CORRECTLY
54 REM   11.   GO TO NEXT ROLL OF DICE
55 REM
56 REM
57 REM   FOURTH SECTION - LINES 1900 THRU 2050
58 REM    1.   IS GAME TIED ?
59 REM    2.   IF NO TIE, WINNER IS DETERMINED
60 REM    3.   HUMAN ASKED TO PLAY ANOTHER GAME
100 REM
110 REM FIRST SECTION.
120 REM
130 PRINT
140 PRINT "     THIS IS A NUMBER GAME CALLED BOBSTONES.  THE OBJECT OF"
150 PRINT "BOBSTONES IS TO GUESS THREE THINGS ABOUT THE ROLL OF A PAIR"
160 PRINT "OF DICE.  ON EACH TURN, THE COMPUTER SIMULATES THE ROLL OF"
170 PRINT "THE DICE.  THEN, YOU OR THE COMPUTER (YOUR OPPONENT) GUESS"
180 PRINT
190 PRINT "                                                    SCORE"
200 PRINT " 1. IF THE SUM OF THE DICE IS ODD OR EVEN          1 POINT"
210 PRINT " 2. THE SUM OF THE DICE                            2 POINTS"
220 PRINT " 3. THE NUMBER ON EACH OF THE TWO DICE             3 POINTS"
230 PRINT
240 PRINT "     THE WINNER IS THE FIRST PLAYER TO SCORE 11 POINTS.  IF A"
250 PRINT "TIE RESULTS, THE WINNER IS THE FIRST PLAYER TO BREAK THE TIE."
260 PRINT "     GOOD LUCK !"
270 DEF FND(X)=INT(6*RND(X)+1)
280 DIM A(2)
290 LET A(1)=0
300 LET A(2)=0
310 LET Z1=-1
320 LET Z2=-1
330 LET Z3=-1
340 LET Z4=-1
350 LET Z5=-1
360 LET J1=0
370 PRINT
380 PRINT
390 PRINT "YOU FIRST OR ME"
400 INPUT Z$
410 IF Z$="YOU" THEN 450
420 IF Z$="ME" THEN 450
430 PRINT "/// TYPE THE WORD 'YOU' OR THE WORD 'ME'."
440 GOTO 390
450 LET D1=FND(Z1)
460 IF Z1=0 THEN 480
470 LET Z1=0
480 LET D2=FND(0)
490 LET S=D1+D2
500 IF J1=0 THEN 650
510 IF Z$<>"ME" THEN 580
520 IF J2<>0 THEN 1160
530 PRINT
540 PRINT "THE SCORE IS ME";A(2);" - YOU";A(1);"."
550 IF A(1)>=11 THEN 1900
560 IF A(2)>=11 THEN 1900
570 GOTO 670
580 IF Z$<>"YOU" THEN 2020
590 IF J2<>1 THEN 670
600 PRINT
610 PRINT "THE SCORE IS YOU";A(1);" - ME";A(2);"."
620 IF A(1)>=11 THEN 1900
630 IF A(2)>=11 THEN 1900
640 GOTO 1160
650 LET J1=-1
660 IF Z$="YOU" THEN 1160
670 PRINT
680 PRINT "YOUR TURN."
690 LET J2=1
700 LET R=S-(INT(S/2)*2)
710 PRINT "IS THE SUM ODD OR EVEN"
720 INPUT A$
730 IF A$="ODD" THEN 770
740 IF A$="EVEN" THEN 800
750 PRINT "/// TYPE THE WORD 'ODD' OR THE WORD 'EVEN'."
760 GOTO 710
770 IF R=1 THEN 820
780 PRINT "SORRY, THE SUM IS";S;"."
790 GOTO 450
800 IF R=0 THEN 820
810 GOTO 780
820 PRINT "YOU ARE CORRECT."
830 LET A(1)=A(1)+1
840 PRINT "NOW, GUESS THE SUM"
850 INPUT G1
860 IF G1<2 THEN 890
870 IF G1>12 THEN 890
880 GOTO 910
890 PRINT "/// THE SUM MUST BE BETWEEN 2 AND 12."
900 GOTO 840
910 IF G1=S THEN 940
920 PRINT "SORRY, THE SUM IS ";S;"."
930 GOTO 450
940 PRINT "YOU ARE CORRECT."
950 LET A(1)=A(1)+2
960 PRINT "WHAT ARE THE TWO NUMBERS WHICH PRODUCED  ";S
970 INPUT N1,N2
980 IF N1<1 THEN 1030
990 IF N2<1 THEN 1030
1000 IF N1>6 THEN 1030
1010 IF N2>6 THEN 1030
1020 GOTO 1050
1030 PRINT "/// THE NUMBERS MUST BE BETWEEN 1 AND 6."
1040 GOTO 960
1050 IF N1=D1 THEN 1090
1060 IF N2=D1 THEN 1110
1070 PRINT "SORRY, THE NUMBERS ARE ";D1;" AND ";D2;"."
1080 GOTO 450
1090 IF N2=D2 THEN 1130
1100 GOTO 1070
1110 IF N1=D2 THEN 1130
1120 GOTO 1070
1130 PRINT "YOU ARE CORRECT."
1140 LET A(1)=A(1)+3
1150 GOTO 450
1160 LET J2=0
1170 PRINT
1180 PRINT "MY TURN."
1190 PRINT "*** ON THIS ROLL OF THE DICE, THE TWO NUMBERS
       ARE ";D1;" AND ";D2;"."
1200 PRINT "*** THE SUM IS ";S;"."
1210 LET A1=INT(2*RND(Z2)+1)
1220 IF Z2=0 THEN 1240
1230 LET Z2=0
1240 IF A1=1 THEN 1270
1250 PRINT "MY GUESS IS THAT THE SUM IS ODD."
1260 GOTO 1280
1270 PRINT "MY GUESS IS THAT THE SUM IS EVEN."
1280 PRINT "AM I RIGHT OR WRONG"
1290 INPUT D$
1300 IF D$="RIGHT" THEN 1340
1310 IF D$="WRONG" THEN 450
1320 PRINT "/// TYPE THE WORD 'RIGHT' OR THE WORD 'WRONG'."
1330 GOTO 1280
1340 LET A(2)=A(2)+1
1350 IF A1=1 THEN 1410
1360 LET B1=INT(5*RND(Z3)+1)
1370 IF Z3=0 THEN 1390
1380 LET Z3=0
1390 LET B2=B1+B1+1
1400 GOTO 1430
1410 LET B1=FND(0)
1420 LET B2=B1+B1
1430 PRINT "MY GUESS OF THE SUM IS ";B2
1440 PRINT "AM I RIGHT OR WRONG"
1450 INPUT D$
1460 IF D$="RIGHT" THEN 1500
1470 IF D$="WRONG" THEN 450
1480 PRINT "/// TYPE THE WORD 'RIGHT' OR THE WORD 'WRONG'."
1490 GOTO 1440
1500 LET A(2)=A(2)+2
1510 IF B2<>2 THEN 1550
1520 LET C1=1
1530 LET C2=1
1540 GOTO 1810
1550 IF B2<>3 THEN 1590
1560 LET C1=1
1570 LET C2=2
1580 GOTO 1810
1590 IF B2<>11 THEN 1630
1600 LET C1=5
1610 LET C2=6
1620 GOTO 1810
1630 IF B2<>12 THEN 1670
1640 LET C1=6
1650 LET C2=6
1660 GOTO 1810
1670 IF B2>7 THEN 1740
1680 LET K1=B2-1
1690 LET C1=INT(K1*RND(Z4)+1)
1700 IF Z4=0 THEN 1720
1710 LET Z4=0
1720 LET C2=B2-C1
1730 GOTO 1810
1740 LET K1=B2-6
1750 LET K3=K1-1
1760 LET K2=7-K1
1770 LET C1=(INT(K2*RND(Z5)+1)+K3)
1780 IF Z5=0 THEN 1800
1790 LET Z5=0
1800 LET C2=B2-C1
1810 PRINT "MY GUESS IS THAT THE NUMBERS ARE ";C1;" AND ";C2;"."
1820 PRINT "AM I RIGHT OR WRONG"
1830 INPUT D$
1840 IF D$="RIGHT" THEN 1880
1850 IF D$="WRONG" THEN 450
1860 PRINT "/// TYPE THE WORD 'RIGHT' OR THE WORD 'WRONG'."
1870 GOTO 1820
1880 LET A(2)=A(2)+3
1890 GOTO 450
1900 IF A(1)<>A(2) THEN 1930
1910 IF J2<>0 THEN 1160
1920 GOTO 670
1930 IF A(1)>A(2) THEN 2030
1940 PRINT
1950 PRINT "I WIN! ANOTHER GAME?"
1960 INPUT C$
1970 IF C$="YES" THEN 290
1980 IF C$="NO" THEN 2010
1990 PRINT "/// TYPE THE WORD 'YES' OR THE WORD 'NO'."
2000 GOTO 1960
2010 PRINT "SEE YOU LATER."
2020 END
2030 PRINT
2040 PRINT "YOU WIN! ANOTHER GAME?"
2050 GOTO 1960
```

Another new game from Creative Computing . . . .

WATCHMAN

Written by: Mac Oglesby, Putney, Vermont.
Language: Standard BASIC.
Description: This is a new version of the old "draw the figure without retracing or lifting your pencil" topological puzzle. The user acts as a watchman hired to patrol the streets of a small village as efficiently as possible. To do this he must find a path that will not retrace any earlier steps.

Suggestions: For CRT users—
(1) Clear screen before every map.
(2) Blink the character showing current position.

```
100 REM WATCHMAN  (BASIC PROGRAM BEGINS AT LINE 240)  WAS WRITTEN BY
110 REM MAC OGLESBY, PUTNEY, VT.  05346 IN FEBRUARY 1975.
120 REM LAST CHANGED IN MARCH 1975 BY BILL COTTER
130 REM
140 REM DESCRIPTION--YOU'VE BEEN HIRED TO PATROL THE VILLAGE STREETS
150 REM WITHOUT RETRACING STEPS.  AN OPTIONAL MAP AFTER EACH TURN SHOWS
160 REM YOUR LOCATION AND FOOTPRINTS.  A NEW VERSION OF AN OLD "DRAW
170 REM THE FIGURE WITHOUT RETRACING OR LIFTING YOUR PENCIL" TOPOLOGICAL
180 REM PUZZLE.
190 REM
200 REM INSTRUCTIONS--TYPE RUN
210 REM
220 REM * * * * * * * * * * * * * * * * * * * * *
230 REM *** INITIALIZATION
240 DIM D(23),P(15,23)
250 LET A3$ = ","
260 LET F=I=M=0
270 FOR J=1 TO 15
280    READ D$
290    CHANGE D$ TO D
300    FOR K=1 TO D(0)
310       LET P(J,K)=D(K)
320    NEXT K
330 NEXT J
340 DATA "      0*********0","      * 1ST ST. *","      *         *"
350 DATA ******0*********0******,*1   2* 2ND ST. *3   4*
360 DATA *S   N*        *R   T*,*T   D* 3RD ST. *D   H*
370 DATA *      0*********0      *,*A   A*        *A   A*
380 DATA *V   V*        *V   V*,*E   E* 4TH ST. *E   E*
390 DATA *N     0*********0******,*U             *
400 DATA *E       5TH ST. *,*****************
410 MATREAD E(5,4)
420 DATA 1,15,21,1,15,210,210,21,6,30,42,6,6,10,210,14,10,35,14,1
430 FOR J=1 TO 5
440    READ R(J)
450 NEXT J
460 DATA 1,4,8,12,15
470 FOR J=1 TO 4
480    READ C(J)
490 NEXT J
500 DATA 1,7,17,23
510 FOR J=1 TO 5
520    READ G$(J)
530 NEXT J
540 DATA ST,ND,RD,TH,TH
550 FOR J=1 TO 8
560    READ B(J)
570 NEXT J
580 DATA 12,13,22,23,32,33,42,43
590 PRINT
600 PRINT "WATCHMAN   ";DAT$
610 PRINT "WANT INSTRUCTIONS";
620 INPUT I$
630 IF I$<>"NO" THEN 660
640 LET I=1
650 GOTO 690
660 PRINT
670 PRINT "YOU'VE BEEN HIRED AS WATCHMAN FOR THE VILLAGE."
680 PRINT "YOUR JOB IS TO PATROL ALL ITS ROADS WITHOUT RETRACING STEPS.'
690 PRINT
700 PRINT "VILLAGE MAP:"
710 GOSUB 2550
720 IF I=1 THEN 770
730 PRINT "YOU NAME A CORNER BY TYPING 2 NUMBERS: FIRST THE STREET,"
740 PRINT "THEN THE AVENUE.  FOR EXAMPLE, 32 MEANS THE CORNER OF"
750 PRINT "3RD ST. AND 2ND AVE."
760 REM *** GET MOVES
770 PRINT
780 IF M>0 THEN 850
790 PRINT "YOU BEGIN AT ANY CORNER MARKED O"
800 PRINT "BEGIN WHERE (ST.,AVE.)";
810 INPUT A1$,A2$
820 LET A$=A1$&A3$
830 LET A$=A$&A2$
840 GOTO 1110
850 PRINT "YOU'RE AT";A1;G$(A1);" STREET AND";A2;G$(A2);" AVENUE."
860 IF M=2 THEN 970
870 IF I=1 THEN 940
880 PRINT
890 PRINT "AT EACH TURN YOU WALK TO ANY ADJACENT CORNER WHICH IS"
900 PRINT "CONNECTED BY A ROAD NOT TROD EARLIER."
910 PRINT "ALTHOUGH YOU CAN'T WALK ON ANY ROAD TWICE, YOU MAY"
920 PRINT "REVISIT CORNERS."
930 PRINT
940 LET M=2
950 GOTO 1040
960 REM *** GAME OVER?
970 IF E(A1,A2)>1 THEN 1030
980 PRINT "WANT FINAL MAP";
990 INPUT M$
1000 IF M$="NO" THEN 2700
1010 GOSUB 2530
1020 GOTO 2700
1030 PRINT "FOR A MAP TYPE '0,0'"
1040 PRINT "WALK TO WHAT CORNER (ST.,AVE.)";
1050 INPUT A1$,A2$
1060 LET A$=A1$&A3$
1070 LET A$=A$&A2$
1080 IF A$<>"0,0" THEN 1110
1090 GOSUB 2530
1100 GOTO 1040
1110 IF LEN(A$)>9 THEN 1150
1120 GOSUB 2410
1130 REM *** LEGAL COMMAND?
1140 IF LEN(A$)=2 THEN 1180
1150 PRINT "ILLEGAL MOVE!"
1160 PRINT "INPUT IGNORED."
1170 GOTO 770
1180 LET Z9$ = SST(A$,1,1)
1190 LET A3=VAL(Z9$)
1200 LET Z9$ = SST(A$,2,1)
1210 LET A4=VAL(Z9$)
1220 IF A4>4 THEN 1150
1230 IF M=>1 THEN 1470
1240 REM *** CAN HE START THERE?
1250 FOR J=1 TO 8
1260    IF VAL(A$)=B(J) THEN 1320
1270 NEXT J
1280 PRINT "SORRY, YOU CAN'T START THERE!"
1290 PRINT "START AT ONE OF THESE CORNERS:"
1300 PRINT "12, 13, 22, 23, 32, 33, 42, OR 43."
1310 GOTO 1160
1320 LET M=1
1330 LET A1=A3
1340 LET A2=A4
1350 LET O$=A$
1360 REM *** CHANGE O TO * ON THE MAP
1370 FOR J=1 TO 5
1380    FOR K=1 TO 4
1390       IF P(R(J),C(K))<>ASC(O) THEN 1410
1400       LET P(R(J),C(K))=ASC(*)
1410    NEXT K
1420 NEXT J
1430 REM *** MARK BEGINNING LOCATION
1440 LET P(R(A1),C(A2))=ASC(&)
1450 GOTO 770
1460 REM *** ADJACENT CORNER?
1470 IF VAL(O$)<>VAL(A$) THEN 1510
1480 PRINT "YOU'RE AT THAT CORNER NOW!"
1490 PRINT "INPUT IGNORED."
1500 GOTO 1030
1510 LET Z9$=SST(O$,1,1)
1520 LET O1=VAL(Z9$)
1530 LET Z9$=SST(O$,2,1)
1540 LET O2=VAL(Z9$)
1550 IF A3-O1=0 THEN 1580
1560 IF ABS(A3-O1)=1 THEN 1590
1570 GOTO 1150
1580 IF ABS(A4-O2)=1 THEN 1610
1590 IF A4-O2<>0 THEN 1150
1600 REM *** WHERE'S HE HEADED?
1610 IF A3-O1<>0 THEN 1720
1620 IF A4-O2<0 THEN 1680
1630 REM *** EAST?
1640 LET H=5
1650 LET H1=7
1660 GOTO 1800
1670 REM *** WEST?
1680 LET H=7
1690 LET H1=5
1700 GOTO 1800
```

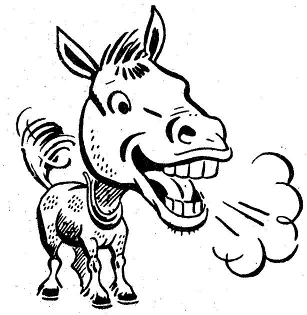
```
1710 REM *** NORTH?
1720 IF A3-01>0 THEN 1770
1730 LET H=2
1740 LET H1=3
1750 GOTO 1800
1760 REM *** SOUTH?
1770 LET H=3
1780 LET H1=2
1790 REM *** DOES PATH EXIST AND IS IT UNTROD?
1800 IF E(01,02)/H<>INT(E(01,02)/H) THEN 1150
1810 REM *** MOVE SEEMS LEGAL, SO RECORD FOOTSTEPS
1820 LET A1=A3
1830 LET A2=A4
1840 LET O$=A$
1850 LET E(01,02)=E(01,02)/H
1860 LET E(A1,A2)=E(A1,A2)/H1
1870 REM *** MARK HIS LOCATION
1880 LET P(R(A1),C(A2))=ASC(&)
1890 LET P(R(01),C(02))=ASC(*)
1900 REM *** KEEP VACATED CORNER?
1910 IF E(01,02)>1 THEN 1930
1920 LET P(R(01),C(02))=ASC(:)
1930 IF H<>2 THEN 1980
1940 FOR J=1 TO ABS(R(A1)-R(01))-1
1950    LET P(R(01)-J,C(02))=ASC(:)
1960 NEXT J
1970 GOTO 2120
1980 IF H<>3 THEN 2030
1990 FOR J=1 TO ABS(R(A1)-R(01))-1
2000    LET P(R(01)+J,C(02))=ASC(:)
2010 NEXT J
2020 GOTO 2120
2030 IF H<>5 THEN 2080
2040 FOR J=1 TO ABS(c(A2)-C(02))-1
2050    LET P(R(01),C(02)+J)=ASC(:)
2060 NEXT J
2070 GOTO 2120
2080 FOR J=1 TO ABS(C(A2)-C(02))-1
2090    LET P(R(01),C(02)-J)=ASC(:)
2100 NEXT J
2110 REM *** CONTINUE WALK IF NO CHOICE IS AVAILABLE
2120 IF E(A1,A2)>7 THEN 770
2130 IF E(A1,A2)=6 THEN 770
2140 IF F=1 THEN 2200
2150 IF I=1 THEN 2190
2160 PRINT
2170 PRINT "(THE COMPUTER AUTOMATICALLY MOVES YOU ALONG TO THE NEXT CORNER"
2180 PRINT "AT WHICH A DECISION IS NECESSARY.)"
2190 LET F=1
2200 IF E(A1,A2)<>7 THEN 2250
2210 LET A4$=STR$(10*A1-1+A2)
2220 LET A9=LEN(A4$)
2230 LET A$=SST(A4$,A9-2,2)
2240 GOTO 1140
2250 IF E(A1,A2)<>5 THEN 2300
2260 LET A4$=STR$(10*A1+1+A2)
2270 LET A9=LEN(A4$)
2280 LET A$=SST(A4$,A9-2,2)
2290 GOTO 1140
2300 IF E(A1,A2)<>3 THEN 2350
2310 LET A4$=STR$(10*A1+10+A2)
2320 LET A9=LEN(A4$)
2330 LET A$=SST(A4$,A9-2,2)
2340 GOTO 1140
2350 IF E(A1,A2)<>2 THEN 770
2360 LET A4$=STR$(10*A1-10+A2)
2370 LET A9=LEN(A4$)
2380 LET A$=SST(A4$,A9-2,2)
2390 GOTO 1140
2400 REM *** ROUTINE TO DROP COMMAS, DASHES, ETC.
2410 LET C1=0
2420 CHANGE A$ TO A
2430 FOR J=1 TO A(0)
2440    IF (ASC(5)-A(J))*(A(J)-ASC(1))<0 THEN 2470
2450    LET C1=C1+1
2460    LET A(C1)=A(J)
2470 NEXT J
2480 LET A(0)=C1
2490 CHANGE A TO A$
2510 RETURN
2520 REM *** PRINT MAP ROUTINE
2530 PRINT
2540 PRINT "& = YOU    :::: = FOOTPRINTS"
2550 PRINT
2560 FOR J=1 TO 15
2570    LET D(0)=0
2580    FOR K=23 TO 1 STEP -1
2590    IF P(J,K)>32 THEN 2610
2600    IF D(0)=0 THEN 2630
2610    LET D(K)=P(J,K)
2620    LET D(0)=D(0)+1
2630    NEXT K
2640    CHANGE D TO D$
2650    PRINT D$
2660 NEXT J
2670 PRINT
2680 RETURN
2690 REM *** END OF GAME
2700 PRINT
2710 REM *** LOSER?
2720 FOR J=1 TO 5
2730    FOR K=1 TO 4
2740       IF E(J,K)=1 THEN 2770
2750       PRINT "YOU WERE SUPPOSED TO PATROL THE WHOLE VILLAGE!"
2760       GOTO 2820
2770    NEXT K
2780 NEXT J
2790 REM *** WINNER!!
2800 PRINT "*** CONGRATULATIONS ***"
2810 PRINT "YOU'VE WALKED THROUGHOUT THE VILLAGE WITHOUT RETRACING STEPS!"
2820 PRINT "TYPE RUN TO TRY AGAIN."
2830 END

ready
```

SAMPLE RUN

```
WATCHMAN    05/06/75
WANT INSTRUCTIONS ?YES

YOU'VE BEEN HIRED AS WATCHMAN FOR THE VILLAGE.
YOUR JOB IS TO PATROL ALL ITS ROADS WITHOUT RETRACING STEPS.

VILLAGE MAP:

       O**********O
       * 1ST ST.  *
       *          *
******O**********O******
*1    2* 2ND ST. *3    4*
*S    N*         *R    T*
*T    D* 3RD ST. *D    H*
*      O**********O     *
*A    A*         *A    A*
*V    V*         *V    V*
*E    E* 4TH ST. *E    E*
*N     O**********O******
*U                *
*E        5TH ST. *
******************

YOU NAME A CORNER BY TYPING 2 NUMBERS: FIRST THE STREET,
THEN THE AVENUE.  FOR EXAMPLE, 32 MEANS THE CORNER OF
3RD ST., AND 2ND AVE.

YOU BEGIN AT ANY CORNER MARKED O
BEGIN WHERE (ST.,AVE.) ?2,2

YOU'RE AT     2 ND STREET AND     2 ND AVENUE.

AT EACH TURN YOU WALK TO ANY ADJACENT CORNER WHICH IS
CONNECTED BY A ROAD NOT TROD EARLIER.
ALTHOUGH YOU CAN'T WALK ON ANY ROAD TWICE, YOU MAY
REVISIT CORNERS.

WALK TO WHAT CORNER (ST.,AVE.) ?2,1

(THE COMPUTER AUTOMATICALLY MOVES YOU ALONG TO THE NEXT CORNER
AT WHICH A DECISION IS NECESSARY.)

YOU'RE AT     4 TH STREET AND     3 RD AVENUE.
FOR A MAP TYPE '0,0'
WALK TO WHAT CORNER (ST.,AVE.) ?4,2

YOU'RE AT     3 RD STREET AND     2 ND AVENUE.
FOR A MAP TYPE '0,0'
WALK TO WHAT CORNER (ST.,AVE.) ?3,2
YOU'RE AT THAT CORNER NOW!
INPUT IGNORED.
FOR A MAP TYPE '0,0'
WALK TO WHAT CORNER (ST.,AVE.) ?2,2

YOU'RE AT     2 ND STREET AND     2 ND AVENUE.
FOR A MAP TYPE '0,0'
WALK TO WHAT CORNER (ST.,AVE.) ?0,0

& = YOU    :::: = FOOTPRINTS

       ************
       * 1ST ST.  *
       *          *
::::::&*****************
:1    2: 2ND ST. *3    4*
:S    N:         *R    T*
:T    D: 3RD ST. *D    H*
:      ************     *
:A    A:         *A    A*
:V    V:         *V    V*
:E    E: 4TH ST. *E    E*
:N     :::::::::::*******
:U                :
:E        5TH ST. :
::::::::::::::::::::

WALK TO WHAT CORNER (ST.,AVE.) ?2,3

YOU'RE AT     2 ND STREET AND     3 RD AVENUE.
FOR A MAP TYPE '0,0'
WALK TO WHAT CORNER (ST.,AVE.) ?3,3

YOU'RE AT     3 RD STREET AND     3 RD AVENUE.
FOR A MAP TYPE '0,0'
WALK TO WHAT CORNER (ST.,AVE.) ?3,2

YOU'RE AT     3 RD STREET AND     2 ND AVENUE.
WANT FINAL MAP ?YES

& = YOU    :::: = FOOTPRINTS

       ************
       * 1ST ST.  *
       *          *
::::::*:::::::::::******
:1    2: 2ND ST. :3    4*
:S    N:         :R    T*
:T    D: 3RD ST. :D    H*
:      &::::::::::*     *
:A    A:         *A    A*
:V    V:         *V    V*
:E    E: 4TH ST. *E    E*
:N     :::::::::::*******
:U                :
:E        5TH ST. :
::::::::::::::::::::

YOU WERE SUPPOSED TO PATROL THE WHOLE VILLAGE!
TYPE RUN TO TRY AGAIN.

ready
```

Another new game from Creative Computing . . . .

DELMAR



Author: Ron Morgan and Kirk Roderick, Northridge, Calif.

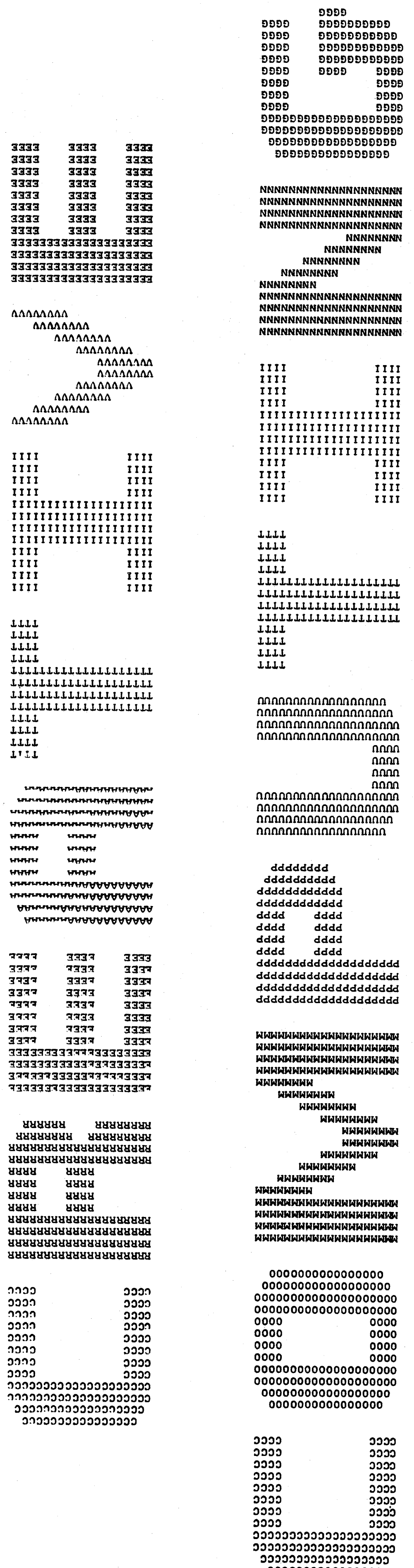
Language: Fortran IV.
Description: See Listing.

```
                         DELMAR
C
C      THIS PRØGRAM SIMULATES A HØRSE RACE STRICTLY USING THE
C      RANDØM FUNCTIØN.
C
C
C      BY RØN MØRGAN, KIRK RØDERICK   12/02/74
C      CSUN, 18111 NØRDHØFF STREET, NØRTHRIDGE  CA  91324
C      THE DØUBLE PRECISIØN REALS, HØRSE AND JØCK, CAN HØLD A
C      MAXIMUM ØF 16 ALPHA-NUMERIC CHARACTERS.  THE CØMMØN BLØCKS
C      REFER TØ EACH ØF THE SUBRØUTINES WHICH ACCESS THEM.
C
C      HØRSE AND JØCK ARE DØUBLE PRECISIØN REAL ARRAYS THAT WILL
C      HØLD THE NAMES ØF THE HØRSES AND JØCKEYS.  IHAND CØNTAINS
C      THE HANDICAP THAT THE HØRSES WILL START AWAY FRØM THE
C      FINISH LINE (FRØM 1 TØ 20).  IØDD IS THE ØDDS (-1) THAT
C      WILL BE PRINTED ØUT (DIVIDES 40).  IHØRSE AND IMT CØNTAIN
C      THE HØRSE NUMBER AND AMØUNT FØR EACH PLAYER'S BET.  IPLAY
C      IS THE NUMBER ØF PLAYERS (1-5).  IBANK CØNTAINS THE AMØUNT
C      ØF MØNEY EACH PLAYER HAS LEFT (INITIALIZED TØ 100).
C      IWIN IS THE NUMBER ØF THE WINNING HØRSE.
C
C
       PRØGRAM DELMAR
       DØUBLE PRECISIØN HØRSE(5),JØCK(5)
       CØMMØN /HØRSEY/ HØRSE,JØCK
       CØMMØN /NUMØDD/ IHAND(5),IØDD(5)
       CØMMØN /BET/ IHØRSE(5),IMT(5),IPLAY,IBANK(5),IWIN
       CALL ESCAPE(8S)
       CALL TIME(3,IHØURS,MINS)
       IMINS=IHØURS*60+MINS
       WRITE(1,3)
 3     FØRMAT(30X,6HDELMAR,2/5X"PRØGRAM SIMULATES A HØRSE RACE
      +BY USE ØF THE RANDØM FUNCTIØN",/)
       WRITE(1,0)\,"DØ YØU WANT INSTRUCTIØNS? "
       IANS=3H
 4     READ(0,5) IANS
 5     FØRMAT(A3)
       IF (IANS.EQ.3HNØ ) GØ TØ 8
       IF (IANS.EQ.3HYES) GØ TØ 6
       WRITE(1,0)\,"PLEASE ANSWER YES ØR NØ: "
       GØ TØ 4
 6     WRITE(1,0)"THIS IS A SIMULATED HØRSE RACE.  THERE ARE 20"
       WRITE(1,0)"HØRSES AND JØCKEYS IN THE STABLE.  YØU START"
       WRITE(1,0)"ØUT WITH $100.  ØNLY WHØLE NUMBER AMØUNTS ARE"
       WRITE(1,0)"ALLØWED IN BETS (I.E. 100 NØT 100.00)."
       WRITE(1,0)"THERE IS A MAXIMUM ØF 5"
       WRITE(1,0)"PLAYERS. THE HØRSES AND JØCKEYS FØR EACH RACE ARE"
       WRITE(1,0)"PICKED RANDØMLY AS ARE THE ØDDS AND THE ACTUAL RAC
      +E."
       WRITE(1,0)"HØWEVER, THE ØDDS DØ HAVE A WEIGHT (ØR HANDICAP)"
       WRITE(1,0)"IN THE ØUTCØME ØF THE RACE."
 8     WRITE(1,0)\,"HØW MANY PLAYERS? "
       READ(0,10) IPLAY
 10    FØRMAT(I1)
       IF (IPLAY.LE.5.AND.IPLAY.GT.0) GØ TØ 15
       WRITE(1,0)"TØØ MANY PLAYERS.  MAXIMUM ØF 5."
       GØ TØ 8
 15    CALL ESCAPE(55S)
       DØ 20 I=1,IPLAY
       IBANK(I)=100
 20    CØNTINUE
 40    DØ 50 I=1,5
       IHØRSE(I)=IMT(I)=IØDD(I)=IHAND(I)=0
       HØRSE(I)=JØCK(I)=16H
 50    CØNTINUE
       CALL PICKS
       CALL ØDDS
       CALL BETS
       CALL ESCAPE (0)
       CALL RACE
       CALL BANKER
 55    IF SENSE LIGHT 1,67,56
 56    WRITE(1,0)\,"DØ YØU WANT ANØTHER RACE? "
       IANS=3H
 60    READ(0,5) IANS
       IF (IANS.EQ.3HYES) GØ TØ 65
       IF(IANS.EQ.3HNØ ) STØP
       WRITE(1,0)\,"PLEASE ANSWER YES ØR NØ: "
       GØ TØ 60
 65    CALL TIME(3,IHØURS,MINS)
       MINS=IHØURS*60+MINS-28
       IF (IMINS.GT.MINS) GØ TØ 40
       WRITE(1,0)"THIS IS YØUR LAST RACE: "
       SENSE LIGHT 1
       GØ TØ 40
 67    STØP
       END
C
C
C      SUBRØUTINE FØR PICKING HØRSES AND JØCKEYS
C
C      --RANDØM FUNCTIØN
C      THE RANDØM FUNCTIØN RETURNS A PSEUDØ-RANDØM NUMBER, R, WHERE
C      0<=R<1.  THERE ARE TWØ MØDES AVAILABLE:  REPRØDUCIBLE AND
C      NØN-REPRØDUCIBLE.  THE ARGUMENT SELECTS THE MØDE.
C
C                                              MØDE ØF      MØDE ØF
C      NAME        FUNCTIØN                    ARGUMENT     RESULT
C
C      RANDØM(I)   NEXT PSEUDØ-RANDØM VALUE    INTEGER      REAL
C
C
C      IF I IS NEGATIVE, THE GENERATØR IS SET BACK TØ THE BEGINNING
C      ØF ITS REPRØDUCIBLE SEQUENCE.  THE RESULT, R, WILL BE THE SAME
C      AS THE FIRST TIME RANDØM IS CALLED AFTER LØADING.  IF I IS
C      ZERØ, THE NEXT NUMBER IN THE REPRØDUCIBLE SEQUENCE IS RETURNED
C      AS THE RESULT.  IF I IS GREATER THAN ZERØ, AN INDEFINITE
C      NUMBER (FRØM 0 TØ 63) ØF THE NEXT SEQUENTIAL INTERMEDIATE
C      12-BIT VALUES ARE SKIPPED BEFØRE THE RANDØM NUMBER IS
C      CØNSTRUCTED.  THE NUMBER ØF VALUES SKIPPED IS DETERMINED BY THE
C      LEAST SIGNIFICANT PØRTIØN ØF THE CURRENT VALUE ØF THE REAL-
C      TIME CLØCK IN THE CØMPUTER.
C
C
C      THE RANDØM NUMBERS ARE CØNSTRUCTED FRØM THREE 12-BIT GRØUPS
C      CALCULATED FRØM 5**(2*N+1)  (MØD 2**39), USING THE LEFT-MØST
C      12-BITS.
C
C         J AND L ARE THE NUMBERS THAT ARE CHØSEN RANDØMLY THAT
C         CØNTRØL THE HØRSE(J) AND JØCKEY(L) THAT ARE CHØSEN.
C         H AND JØ ARE DEFINED IN DATA AS THE HØRSES(H) AND JØCKEYS(JØ)
C         THE CHECK TØ SEE IF THE HØRSE(ØR JØCKEY) WAS ALREADY CHØSEN
C         STARTS AT 90.
C
          SUBRØUTINE PICKS
          DØUBLE PRECISIØN HØRSE(5),JØCK(5),H(20),JØ(20)
          CØMMØN /HØRSEY/ HØRSE,JØCK
          DIMENSIØN J(5),L(5)
          DATA H /16HACK ACK         ,16HØVERGLØRY       ,
         +16HTIMETØLIGHT     ,16HGØLDEN WALLET   ,16HRAVISHING RUBY  ,
         +16HRED RUNNING GØØD,16HACRØCYANØSIS    ,16HNADINE MALCØLM  ,
         +16HWHØØPEE         ,16HSHAMALEEN       ,16HMISS ALERT      ,
         +16HCØMMAND PRINCE  ,16HIRØNSIDE        ,16HWILD SURF       ,
         +16HCANNØNERØ II    ,16HDAMAGE CØNTRØL  ,16HSHIEK KAMIAKIN  ,
         +16HAUDACITY        ,16HCØURT CLØWN     ,16HSWEET ANASTACIA /
          DATA JØ /16HKSINNER         ,16HGBAZE           ,
         +16HJLEØNARD        ,16HSHØEMAKER       ,16HJTGØNZALEZ      ,
         +16HSGØLDSMITH      ,16HMLEWIS          ,16HRYAKA           ,
         +16HWDELIA          ,16HAØCHØA          ,16HGLAWLESS        ,
         +16HPINEDA          ,16HMRUJANØ         ,16HKFURLØNG        ,
         +16HSARCHULETA      ,16HMUALENAUELA     ,16HJWILBURN        ,
         +16HFMENA           ,16HSVALDEZ         ,16HPEREZ           /
          CALL ESCAPE (0)
 70       DØ 120 I=1,5
 80       J(I)=IFIX(RANDØM(5)*20)
          L(I)=IFIX(RANDØM(5)*20)
          IF(J(I)=0.ØR.L(I)=0) GØ TØ 80
          IF (I.EQ.1) GØ TØ 105
 90       DØ 100 L2=1,I-1
          IF(J(I)=J(L2).ØR.L(I)=L(L2)) GØ TØ 80
 100      CØNTINUE
 105      HØRSE(I)=H(J(I))
          JØCK(I)=JØ(L(I))
 120      CØNTINUE
          RETURN
          END
C
C
C         SUBRØUTINE FØR DETERMINING ØDDS
C
C           THE ØDDS ARE DETERMINED BY HAVING 40 DIVIDED BY THE
C           HANDICAP.  THIS PRØDUCES THE ØDDS FRØM 2-1 TØ 40-1.
            SUBRØUTINE ØDDS
            CØMMØN /NUMØDD/ IHAND(5),IØDD(5)
            CALL ESCAPE (0)
            DØ 150 I=1,5
 130        IHAND(I)=IFIX(RANDØM(5)*20)
            IF (IHAND(I).LE.0) GØ TØ 130
 140        IØDD(I)=IFIX(40/IHAND(I))
 150        CØNTINUE
            RETURN
            END
C
C
C         SUBRØUTINE FØR TAKING BETS
C
C           IF THE PLAYER DØES NØT HAVE ANY MØRE MØNEY (IBANK=0)
C           THEN THAT PLAYER DØES NØT GET A CHANCE TØ BET AND THE
C           AMØUNT ØF THE BET IS PUT TØ 0 (IMT=0).
C
            SUBRØUTINE BETS
            DØUBLE PRECISIØN HØRSE(5),JØCK(5)
            CØMMØN /HØRSEY/ HØRSE,JØCK
            CØMMØN /BET/ IHØRSE(5),IMT(5),IPLAY,IBANK(5),IWIN
            CØMMØN /NUMØDD/ IHAND(5),IØDD(5)
            CALL ESCAPE (0)
            WRITE(1,160)
 160        FØRMAT(1X,3/,1X"#",6X,"HØRSE"12X"JØCKEY"22X,"ØDDS",/)
```

```
      DØ 180 I=1,5
      WRITE(1,170) I,HØRSE(I),JØCK(I),IØDD(I)
170   FØRMAT("(",I1,")",5X,A16,1X,A16,12X,I2,"-1")
180   CØNTINUE
      WRITE(1,190)
190   FØRMAT(\,1X,3/"TYPE THE NUMBER ØF THE HØRSE, THE AMØUNT ØF
     + THE BET.")
      DØ 220 I=1,IPLAY
      CALL ESCAPE (195S)
      IF (IBANK(I).LE.0) GØ TØ 217
195   WRITE(1,200) I
200   FØRMAT(\,1X,/,"PLAYER # "I1,1X)
      READ(0,0) IHØRSE(I),IMT(I)
      IF (IHØRSE(I).LE.5.AND.IHØRSE(I).GT.0) GØ TØ 215
      WRITE(1,0)"NØ SUCH HØRSE #."
      GØ TØ 195
215   IF (IMT(I).LE.IBANK(I).ØR.IMT(I).LT.0) GØ TØ 220
      WRITE(1,0)"YØU DØN'T HAVE THAT MUCH MØNEY.  ENTER HØRSE #,
     +AND BET."
      GØ TØ 195
217   IMT(I)=0
220   CØNTINUE
      RETURN
      END
C
C
C     THIS SUBRØUTINE RUNS THE RACE INTERNALLY AND PRINTS THE RESULTS
C
C       IBELL CØNTAINS ALL G'S SØ THAT THE BELLS CAN BE ØUTPUT
C       USING THE C FØRMAT.  THE RACE IS RUN BY ADDING A RANDØM
C       NUMBER (FRØM 0-5) TØ THE ACCUMULATED TØTAL AND THE
C       HANDICAP (IHAND).  WHEN ØNE HØRSE CRØSSES THE FINISH LINE
C       (IHAND>=60) THE RACE ENDS AND THE RESULTS ARE PRINTED TØ
C       SHØW WHERE EACH HØRSE WAS WHEN THE WINNER CRØSSED THE
C       FINISH LINE.
C
      SUBRØUTINE RACE
      CØMMØN /BET/ IHØRSE(5),IMT(5),IPLAY,IBANK(5),IWIN
      CØMMØN /NUMØDD/ IHAND(5),IØDD(5)
      DØUBLE PRECISIØN IBELL
      CALL ESCAPE (0)
      WRITE(1,0)"AT THE SØUND ØF THE BELL THEY'RE ØFF."
      DØ 230 I=1,10000
230   CØNTINUE
      IBELL=10HGGGGGGGGGG
      WRITE(1,233) IBELL
233   FØRMAT(\,C10)
235   DØ 240 I=1,5
      IHAND(I)=IHAND(I)+IFIX(RANDØM(5)*5)
240   CØNTINUE
      DØ 245 I1=1,5
      IF (IHAND(I1).GE.60) GØ TØ 250
245   CØNTINUE
      GØ TØ 235
250   IWIN1=IHAND(1)
      DØ 260 I=1,5
      IWIN1=MAX0(IWIN1,IHAND(I))
260   CØNTINUE
      DØ 265 I=1,5
      IF(IWIN1.EQ.IHAND(I)) IWIN1=I
265   CØNTINUE
      WRITE(1,267)
267   FØRMAT(49X,"FINISH LINE^")
      IWIN=IWIN1
      WRITE(1,0) "HERE ARE THE RESULTS:"
      DØ 290 I=1,5
      WRITE(1,270) I
270   FØRMAT(\,"(",I1,")")
      DØ 280 N=1,IHAND(I)-1
      WRITE(1,0)\,"-"
280   CØNTINUE
      WRITE(1,0) "*"
290   CØNTINUE
      RETURN
      END
C
C
C     THIS SUBRØUTINE KEEPS A RECØRD ØF WINS AND LØSSES.
C
C       THE BANK TØTALS ARE DETERMINED BY WHETHER THE PLAYER
C       CHØSE THE WINNER.  IF A BANK BALANCE BECØMES GREATER
C       THAN 9999 THE PRØGRAM ENDS AND PRINTS ØUT A MESSAGE,
C       ØR IF ALL THE PLAYERS RUN ØUT ØF MØNEY ANØTHER MESSAGE IS
C       PRINTED ØUT.
C
      SUBRØUTINE BANKER
      CØMMØN /BET/ IHØRSE(5),IMT(5),IPLAY,IBANK(5),IWIN
      CØMMØN /NUMØDD/ IHAND(5),IØDD(5)
      CALL ESCAPE (0)
      WRITE(1,300) IWIN
300   FØRMAT(1X/"HØRSE #",I1," WØN."/"HERE ARE YØUR BANK BALANCES:")
      I1=I2=0
      DØ 340 I=1,IPLAY
      IF(IHØRSE(I).EQ.IWIN) GØ TØ 310
      IBANK(I)=IBANK(I)-IMT(I)
      GØ TØ 320
310   IBANK(I)=IBANK(I)+IMT(I)*(IØDD(IWIN))
320   WRITE(1,330) I,IBANK(I)
330   FØRMAT(1X,/"PLAYER #",I1," $ ",F7.2)
      IF (IBANK(I).GT.9999) SENSE LIGHT 2
      IF(IBANK(I).LE.0) I2=I2+1
340   CØNTINUE
      IF SENSE LIGHT 2,350,345
345   IF(I2.GE.IPLAY) GØ TØ 360
      RETURN
350   WRITE(1,0)"YØU HAVE BRØKEN THE BANK!!!!!"
      STØP
360   WRITE(1,0)"YØU ALL RAN ØUT ØF MØNEY.  PLEASE PAY THE"
      WRITE(1,0)"CASHIER WHEN YØU EXIT.  THANK YØU."
      STØP
      END
```

SAMPLE RUN

```
                              DELMAR

    PROGRAM SIMULATES A HORSE RACE BY USE OF THE RANDOM FUNCTION

DO YOU WANT INSTRUCTIONS? NO
HOW MANY PLAYERS? 2


 #      HORSE              JOCKEY                      ODDS

(1)     DAMAGE CONTROL     GLAWLESS                     4-1
(2)     GOLDEN WALLET      JWILBURN                     8-1
(3)     WHOOPEE            SARCHULETA                   2-1
(4)     COMMAND PRINCE     JTGONZALEZ                   5-1
(5)     WILD SURF          JLEONARD                    13-1


TYPE THE NUMBER OF THE HORSE, THE AMOUNT OF THE BET.
PLAYER # 1 2,10.00

PLAYER # 2 3,50.00
AT THE SOUND OF THE BELL THEY'RE OFF.
                                                 FINISH LINE↑
HERE ARE THE RESULTS:
(1)----------------------------------------------*
(2)-------------------------------------*
(3)--------------------------------------------------------------*
(4)-------------------------------------*
(5)----------------------------------------------*

HORSE #3 WON.
HERE ARE YOUR BANK BALANCES:

PLAYER #1 $   90.00

PLAYER #2 $  200.00
DO YOU WANT ANOTHER RACE? YES


 #      HORSE              JOCKEY                      ODDS

(1)     WHOOPEE            MLEWIS                      20-1
(2)     OVERGLORY          KSINNER                      6-1
(3)     ACROCYANOSIS       SVALDEZ                      2-1
(4)     ACK ACK            MRUJANO                      3-1
(5)     AUDACITY           MUALENAUELA                 40-1


TYPE THE NUMBER OF THE HORSE, THE AMOUNT OF THE BET.
PLAYER # 1 3,33.28

PLAYER # 2 4,25.00
AT THE SOUND OF THE BELL THEY'RE OFF.
                                                 FINISH LINE↑
HERE ARE THE RESULTS:
(1)-----------------------------------*
(2)----------------------------------------------*
(3)-------------------------------------------------------------*
(4)----------------------------------------------------*
(5)------------------------------------------------------*

HORSE #3 WON.
HERE ARE YOUR BANK BALANCES:

PLAYER #1 $  156.00

PLAYER #2 $  175.00
DO YOU WANT ANOTHER RACE? YES


 #      HORSE              JOCKEY                      ODDS

(1)     COURT CLOWN        JWILBURN                     4-1
(2)     WHOOPEE            SVALDEZ                      2-1
(3)     WILD SURF          GBAZE                        4-1
(4)     SHIEK KAMIAKIN     AOCHOA                       4-1
(5)     TIMETOLIGHT        SARCHULETA                   3-1


TYPE THE NUMBER OF THE HORSE, THE AMOUNT OF THE BET.
PLAYER # 1 5,30.00

PLAYER # 2 2,50.00
AT THE SOUND OF THE BELL THEY'RE OFF.
                                                 FINISH LINE↑
HERE ARE THE RESULTS:
(1)----------------------------------------*
(2)-----------------------------------------------*
(3)--------------------------------------------*
(4)--------------------------------------------*
(5)-----------------------------------------------------------*

HORSE #5 WON.
HERE ARE YOUR BANK BALANCES:

PLAYER #1 $  246.00

PLAYER #2 $  125.00
DO YOU WANT ANOTHER RACE? YES
```



POSTER

Bradford Huntress
Hingham, Mass.

The program POSTER is an adaptation of the program *Banner.* It was designed to run under Educomp EDU250 BASIC, but it can easily be adapted to other BASICs as well.

What started out to be a modest attempt merely to adapt the original program to our BASIC ended up as a complete revision now bearing little resemblance to its predecessor. The first problem was finding a simulation for the statement 'change'. I found an EDU250 user-defined function that simulated the ASC( ) function, so the first problem was solved.

Meanwhile, the letters the program would print out were sometimes garbled and altogether plain-looking. They all had square corners with very simple subroutines that controlled their printout. For our system which has less than 4K core per user, the fact that each subroutine performed one and only one function per subroutine, was extremely wasteful. So in my program, I made multi-purpose subroutines that performed two or more functions per subroutine. In doing this I completely changed the shape of all the letters and added a few more characters of my own. All the letters now have rounded corners where appropriate. In my program no two letters except the 'O' and the zero print out the same way. In the former version the '2' and the 'Z' are the same, the 'I' and the '1' are the same, etc. Since it now bears so little resemblance to its forerunner, I decided to call it POSTER instead of Banner.

The software:

Educomp EDU250 BASIC under OS/8 with the following "non-standard" features:

- IF THEN (statement)
- Multiple statements per line; delimiter=':'
- CAT(X$,Y$) concatenates two strings and puts it in the assigned string.
- MID(Z$,X,Y) takes a substring of Z$ starting at X and counting Y characters, and puts it in the assigned string.

Adaptations:

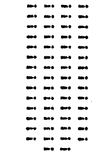
The statement DEF FNA(X)=CAT(" +0",CAT(MID(X,1,1)," "))—992 simulates the ASC( ) function which takes the string argument and finds the ASCII code for it.

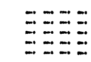
To run the program:

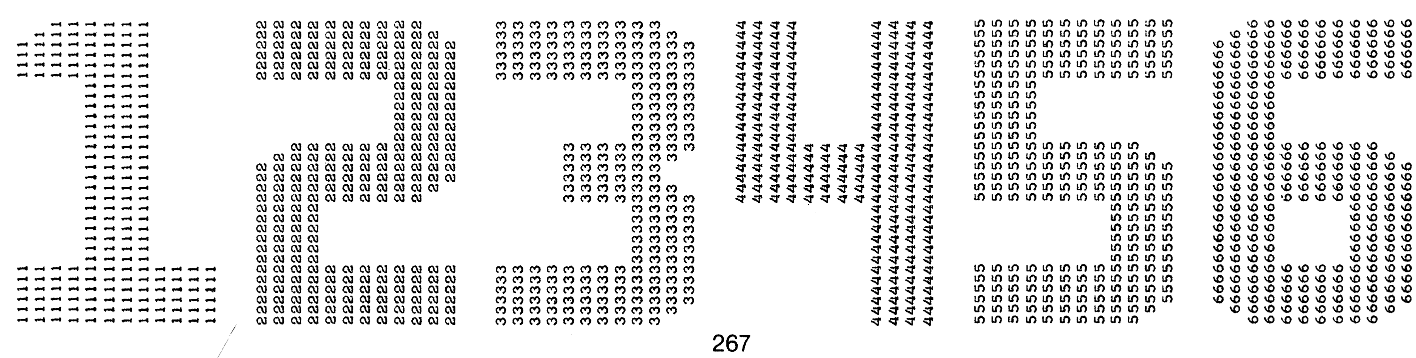
Type 'RUN'

The computer will query you for the height, width, and lefthand margin in inches. It will then ask you to type in your message. The computer does the rest.

```
2 !BRADFORD HUNTRESS H.H.S. HINGHAM MASS
200 PRINT"INPUT HEIGHT, WIDTH, & LEFTHAND MARGIN.(ALL IN INCHES)":INPUTL
,R,S
260 PRINT"INPUT MESSAGE":LINPUTB$
261 DEFFNA(X)=CAT(" +0",CAT(MID(X,1,1),"   "))-992
262 S=S*12:A=2*R:GOSUB1500:H$=CAT(H$(L)," "):GOSUB940
270 A=2*R:F=F+1:A2=INT((F/6-INT(F/6))*6+.001):C=A:IFF>B$(0)THEN990
271 IFF/6=INT(F/6)THENA2=6
272 J$=MID(B$(INT((F-1)/6)+1),A2,1):A1=FNA(J$)
277 G$="":FORX=1TOL:G$=CAT(G$,J$):NEXTX
284 FORX=1TOL:G$(X)="":FORY=1TOX:G$(X)=CAT(G$(X),J$):NEXTY
286 NEXTX
290 FORX=1TOA/2:PRINT:NEXTX
294 IFA1=47THENGOSUB880:GOTO270
295 IFA1=39THENG6=1:GOSUB1600:G6=0:GOSUB1600:GOTO270
296 IFA1=33THENG6=2:GOSUB1800:G6=1:GOSUB1800:GOTO270
300 IFA1=46THEN345
302 IFA1=32THEN630
303 IFA1<48THEN270
305 IFA1>57THEN310
306 ONA1-47GOTO500,445,621,625,635,552,615,605,595,585
310 IFA1<65THEN270
320 ONA1-64GOTO350,360,370,380,390,400,410,430,440,450,460,470,480,490,5
00,510,520
331 ONA1-81GOTO530,550,560,570,580,590,600,610,620:GOTO270
345 FORX=1TOA:PRINTTAB(S);G$;G$:NEXTX
349 GOTO 270
350 G6=0:GOSUB1300:A=C/2:GOSUB640:A=C:GOSUB690:A=C/2:GOSUB640:GOSUB1400:
GOTO270
360 GOSUB640:GOSUB650:A=C/2:GOSUB640:G6=0:GOSUB805:GOTO270
370 G6=1:GOSUB840:A=C/2:GOSUB640:A=C:GOSUB700:GOSUB700:GOTO270
380 GOSUB640:GOSUB700:A=C/2:GOSUB640:G6=0:GOSUB840:GOTO270
390 GOSUB640:GOSUB650:GOSUB650:GOTO270
400 GOSUB640:GOSUB690:GOSUB690:GOTO270
410 G6=1:GOSUB840:A=C/2:GOSUB640:A=C*.75:GOSUB700:A=A/3:GOSUB650:A=2*A
411 GOSUB760:G6=3:GOSUB1800:A=A/2:GOSUB710:GOTO270
430 GOSUB640:GOSUB710:GOSUB640:GOTO270
440 GOSUB700:GOSUB640:GOSUB700:GOTO270
445 GOSUB912:A=C/2:GOSUB700:A=C:GOSUB640:GOSUB740:GOTO270
450 GOSUB645:A=C/2:GOSUB750:A=C:GOSUB740:A=C/2:GOSUB640:G6=1:GOSUB1400:G
OTO270
460 GOSUB640:GOSUB860:GOTO270
470 GOSUB640:GOSUB740:GOSUB740:GOTO 270
480 GOSUB 640:GOSUB890:GOSUB880:GOSUB640:GOTO270
490 GOSUB 640:GOSUB890:GOSUB640:GOTO270
500 A=C/2:G6=1:GOSUB840:GOSUB640:GOSUB700:GOSUB700:GOSUB640:G6=0:GOSUB84
0:GOTO270
510 GOSUB640:GOSUB690:A=C/2:GOSUB790:G6=0:GOSUB1600:GOTO270
520 G6=1:GOSUB840:A=C/2:GOSUB640:A=C*.75:GOSUB700:A=C/4:GOSUB650
521 A=C/2:GOSUB640:A=C/4:G6=0:GOSUB840:GOSUB710:A=C:GOTO270
530 GOSUB640:GOSUB690:A=C/2:GOSUB640:G6=0:GOSUB805:GOTO270
540 GOSUB640:GOSUB690:A=C/2:GOSUB640:G6=0:GOSUB805:GOTO270
550 G6=2:GOSUB1800:A=C/2:GOSUB 770:A=C:GOSUB650:A=C/2:GOSUB 760:G6=3
551 GOSUB1800:GOTO270
552 GOSUB770:GOSUB650:A=C/2:GOSUB760:G6=3:GOSUB1800:GOTO270
560 GOSUB780:GOSUB640:GOSUB780:GOTO270
570 G6=1:GOSUB1300:A=C/2:GOSUB640:A=C:GOSUB740:A=C/2:GOSUB640:GOSUB1400:
GOTO270
580 GOSUB890:GOSUB880:GOTO270
585 G6=1:GOSUB1600:A=C/2:GOSUB790:A=C:GOSUB690:A=C/2:GOSUB640:G6=0:GOSUB
1400:GOTO270
590 G6=1:GOSUB1300:A=C/2:GOSUB640:A=C:GOSUB880:GOSUB890:A=C/2:GOSUB640
591 GOSUB1400:GOTO270
595 G6=1:GOSUB805:A=C/2:GOSUB640:A=C: GOSUB650:A=C/2:GOSUB640:G6=0
596 GOSUB805:GOTO270
600 GOSUB900:GOTO270
605 GOSUB780:GOSUB780:GOSUB640:GOTO270
610 G6=0:GOSUB960:GOSUB800:G6=1:GOSUB960:GOTO270
615 A=C/2:G6=1:GOSUB840:GOSUB640:GOSUB650:GOSUB650:GOSUB760:G6=3:GOSUB18
00:GOTO270
620 GOSUB920:GOTO270
621 A=C/2:G6=4:GOSUB1800:GOSUB760:A=C:GOSUB650:A=C/2:GOSUB770:G6=1:GOSUB
1800:GOTO270
625 GOSUB700:GOSUB650:A=C/2:GOSUB640:G6=0:GOSUB805:GOTO270
630 FORX=1TOC*3:PRINT:NEXTX
631 GOTO270
635 GOSUB790:GOSUB710:GOSUB640:GOTO270
640 FORY=1TOA:PRINTTAB(S);:FORX=1TO10:PRINTG$;:NEXTX
641 PRINT:NEXTY
642 RETURN
645 FORX=C/2TO1STEP-1:PRINTTAB(S+X);:FORY=1TO4*L-X:PRINTG$(1);:NEXTY
646 PRINT:NEXTX
647 RETURN
650 IF A<1THENA=1
670 FORX=1TOA:PRINTTAB(S);G$;G$;H$;H$;G$;G$;H$;H$;G$;G$:NEXTX
671 RETURN
690 FORX=1TOA:PRINTTAB(4*L+S);G$;G$;H$;H$;G$;G$:NEXTX
691 RETURN
700 FORX=1TOA:PRINTTAB(S);G$;G$;:PRINTTAB(S+8*L);G$;G$:NEXTX
701 RETURN
710 IF X<1THENA=1
711 FORX=1TOA:PRINTTAB(4*L+S);G$;G$:NEXTX
712 RETURN
740 FORX=1TOA:PRINTTAB(S);G$;G$:NEXTX
746 RETURN
750 FORX=1TOA:PRINTTAB(S);G$;G$;G$;G$:NEXTX
756 RETURN
760 FORY=1TOA:PRINTTAB(S);G$;G$;G$;G$;G$;G$;H$;H$;G$;G$:NEXTY
766 RETURN
770 FORX=1TOA:PRINTTAB(S);G$;G$;H$;H$;G$;G$;G$;G$;G$;G$:NEXTX
776 RETURN
780 FORX=1TOA:PRINTTAB(8*L+S);G$;G$:NEXTX
786 RETURN
790 FORX=1TOA:PRINTTAB(4*L+S);G$;G$;G$;G$;G$;G$:NEXTX
796 RETURN
800 FORX=1TOA:PRINTTAB(S);G$;G$;G$;G$;G$;G$:NEXTX
801 RETURN
805 IFG6=1THENG7=C/2:G9=-1:G8=1:GOTO810
806 G7=1:G9=1:G8=C/2
810 FORX=G7TOG8STEPG9:IFA1=82THENPRINTTAB(S);:GOTO812
811 PRINTTAB(S+X);
812 FORY=1TO2:IFA1=82THENIFY=1THENK9=X:GOTO814
813 K9=2*X
814 FORZ=1TO5*L-K9:PRINTG$(1);:NEXTZ
820 IFX>3THEN950
821 V=X
830 PRINTH$(2*V);" ";:NEXTY
832 PRINT:NEXTX
834 RETURN
840 IFG6=1THENG7=C/2:G8=1:G9=-1:GOTO842
841 G7=1:G8=C/2:G9=1
842 FORX=G7TOG8STEPG9:PRINTTAB(S+X);:FORY=1TO10*L-2*X:PRINTG$(1);:NEXTY
843 PRINT:NEXTX
844 RETURN
860 IFR>LTHENG6=L*4:G7=1:GOTO865
861 G7=INT(L/R+.5):G6=C*2*L/R
865 FORX=1TOG6STEPG7:PRINTTAB(S+4*L-X);G$;G$;TAB(S+4*L+X);G$;G$:NEXTX
871 RETURN
880 FORX=0TO6*LSTEP 6*L/C:PRINTTAB(X+S);G$;G$;G$;G$:NEXTX
881 RETURN
890 FORX=6*LTO0STEP-6*L/C:PRINTTAB(X+S);G$;G$;G$;G$:NEXTX
891 RETURN
900 FORX=0TO7*LSTEP(14*L)/(C*6):PRINTTAB(X+S);G$;G$;G$;
910 PRINTTAB(7*L-X+S);G$;G$;G$:NEXTX
911 RETURN
912 FORX=1TOC/2:PRINTTAB(S);G$;G$;:PRINTTAB(S+8*L);G$;:FORY=1TOX:PRINTG$
(1);:NEXTY
913 PRINT:NEXTX
914 RETURN
920 FORX=0TO7*LSTEP(7*L)/(C*3):PRINTTAB(S);G$;G$;TAB(X+S);G$;G$;G$;
930 PRINTTAB(8*L+S);G$;G$:NEXTX
931 RETURN
940 FORX=1TOA*3:PRINT:NEXTX
941 RETURN
950 FORW=1TOX-3:PRINT"  ";:NEXTW
951 V=3:GOTO 830
960 IFG6=1THENG7=4*L:G8=7*L:G9=4*L/C:GOTO962
961 G7=7*L:G8=4*L:G9=FIX(-4*L/C)
962 FORX=G7TOG8STEPG9:PRINTTAB(X+S);G$;G$;G$:NEXTX
963 RETURN
980 DATA""," ","  ","   ","    ","     "
990 FORX=1TOC*3:PRINT:NEXTX
1000 STOP
1300 FORX=C/2TO1STEP-1:IFG6=1THENPRINTTAB(S+X);:GOTO1304
1302 PRINTTAB(S);
1304 FORY=1TO10*L-X:PRINTG$(1);:NEXTY
1305 PRINT:NEXTX
1306 RETURN
1400 FORX=1TOC/2:IFG6=1THENPRINTTAB(S+X);:GOTO1403
1402 PRINTTAB(S);
1403 FORY=1TO10*L-X:PRINTG$(1);:NEXTY
1404 PRINT:NEXTX
1405 RETURN
1500 RESTORE:FORJ8=1TO6:READH$(J8):NEXTJ8
1501 RETURN
1600 IFG6=0THENG7=1:G8=C/2:G9=1:GOTO1602
1601 G7=C/2:G8=1:G9=-1
1602 FORX=G7TOG8STEPG9:PRINTTAB(4*L+X+S);:FORY=1TO6*L-2*X:PRINTG$(1);:NE
XTY
1603 PRINT:NEXTX
1604 RETURN
1800 IFG6/2=INT(G6/2)THENG7=C/2:G8=1:G9=-1:GOTO1812
1801 G7=1:G8=C/2:G9=1
1812 FORX=G7TOG8STEPG9:IFG6<=2THENG4=0:G5=4*L:GOSUB1820:GOSUB1830:GOTO18
60
1814 G4=8*L:G5=0:GOSUB1830:GOSUB1820:GOTO1860
1820 PRINTTAB(S+G4);G$;G$;:RETURN
1830 IFG6=4THENIFA1=50THENPRINTTAB(S);G$(X);
1840 PRINTTAB(S+G5+X);:FORY=1TO6*L-2*X:PRINTG$(1);:NEXT Y
1841 IFA1=71THENFORY=1TOX:PRINTG$(1);:NEXTY
1842 RETURN
1860 PRINT:NEXTX
1861 RETURN
```







Another new game from Creative Computing . . . .

LEM

Author: Unknown
Modified by: Bill Cotter, Pittsfield, Mass.
Language: BASIC (Honeywell 600/6000)
Description: The user is put at the controls of yet another Lunar Module. The first task is to pick the initial conditions—speed, etc. This lets the user progress in a learning fashion; there is no random factoring involved.

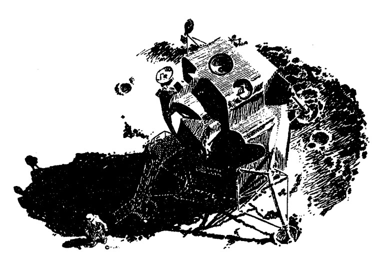
Factors to be considered:

(1) Landing speed—land harder than 3 meters/sec and that's it.
(2) Moving too fast over the terrain causes you to flip when you land.
(3) Your engines will blow up if used to the limit.

Suggestions: Improve lines 2080-2250 (the landing plot).

PROGRAM LISTING

```
100 PRINT "THIS IS A LUNAR LANDING PROGRAM. "
110 PRINT "DO YOU WISH INSTRUCTIONS";
120 INPUT V$
130 IF V$="NO" GOTO 330
140 PRINT "THIS IS THE LANDING ZONE."
150 PRINT TAB(10);"^";TAB(12); "Y AXIS (+)"
160 FOR I=1 TO 4
170 PRINT TAB(10);"!"
180 NEXT I
190 PRINT "----------+---------> X AXIS (+)"
200 FOR I=1 TO 5
210 PRINT TAB(10);"!"
220 NEXT I
230 PRINT "THE POSITIVE Z AXIS IS OUT OF THE PAPER"
240 PRINT "YOU ARE TRYING TO LAND ON THE CROSS."
250 PRINT "YOU HAVE CONTROL OF YOUR VERTICAL(Z), HORIZONTAL(Y),"
260 PRINT "AND TRANSVERSE(X) VELOCITIES. YOU ALSO HAVE THE "
270 PRINT "ADDITIONAL ABILITY TO CONTROL THE LENGTH OF TIME OF "
280 PRINT "BURN. YOU WILL SUPPLY ALL INITIAL DATA."
290 PRINT "ALL UNITS ARE METRIC."
300 PRINT
310 PRINT "REMEMBER IF YOU RUN OUT OF FUEL THATS IT."
320 PRINT
330 PRINT "WHAT IS THE INITIAL ALTITUDE";
340 INPUT A3
350 PRINT "WHAT IS THE INITIAL VERTICAL VELOCITY (DOWN IS +)";
360 INPUT V6
370 PRINT "WHAT IS THE DISTANCE Y";
380 INPUT D4
390 PRINT "WHAT IS Y VELOCITY";
400 INPUT V4
410 PRINT "WHAT IS THE DISTANCE X";
420 INPUT D5
430 PRINT "WHAT IS THE X VELOCITY";
440 INPUT V5
450 PRINT "WHAT IS THE MAXIMUM BURN RATE";
460 INPUT M
470 PRINT "WHAT IS YOUR FUEL CAPACITY";
480 INPUT F3
490 PRINT "WHAT IS THE GRAVITATIONAL CONSTANT";
500 INPUT G
510 PRINT "WHAT IS THE NAME OF YOUR SHIP";
520 INPUT N1$
530 PRINT
540 PRINT  "CONTROL TO "N1$" COMMENCE LANDING."
550 GO SUB 1900
560 LET V=V6
570 LET V1=V4
580 LET V2=V5
590 LET T=0
600 LET F=F3
610 LET A=A3
620 LET D1=D4
630 LET D2=D5
640 LET T3=T3+T
650 GOSUB 2080
660 PRINT "TIME ="T3"SEC."
670 PRINT "ALT="A" METERS V="V" METERS/SEC"
680 PRINT "DIST. X="D2"METERS. V="V2"METERS/SEC"
690 PRINT "DIST. Y="D1"METERS. V="V1"METERS/SEC"
700 PRINT "FUEL="F"UNITS"
710 PRINT "TIME INTERVAL";
720 INPUT T
730 PRINT "VERTICAL(Z) BURN";
740 INPUT B
750 PRINT "TRANSVERSE(X) BURN";
760 INPUT B2
770 LET B2=-B2
780 PRINT "HORIZONTAL (Y) BURN";
790 INPUT B1
800 LET B1=-B1
810 PRINT
820 GOTO 1340
830 LET F1=F
840 LET A1=A
850 IF F<=0 GOTO 1190
860 LET A=A-V*T-((G-B)*T)/2
870 IF A<=0 GOTO 910
880 LET V=V+(G-B)*T
890 GO SUB 2010
900 GO TO 640
910 LET G1=G-B
920 LET T1=((-2*V)+SQR(ABS(4*V^2+8*G*A1)))/(2*G)
930 LET V=V+(G1*T1)
940 GOSUB 2010
950 LET T4=T3+T1
960 IF V<3 GOTO 990
970 PRINT "AT T="T4" THE "N1$" CRASHED WITH A DESCENT ";
980 GOTO 1000
990 PRINT "AT T="T4"THE "N1$" LANDED WITH A DESCENT ";
1000 PRINT "VELOCITY OF"V"M/SEC"
1010 LET R=SQR(V1^2+V2^2)
1020 IF R>5 GOTO 1050
1030 PRINT "WITH A HORIZONTAL VELOCITY OF "R"M/SEC"
1040 GOTO 1080
1050 LET P9=1
1060 PRINT "AND FLIPPED OVER WITH A HORIZONTAL VELOCITY OF"R"M/SEC"
1070 LET P9=1
1080 PRINT "THE LANDING POINT WAS AT ("D2","D1")"
1090 LET D=SQR(D1^2+D2^2)
1100 IF R<=5 GOTO 1140
1110 PRINT D"METERS FROM THE LANDING SITE."
1120 GOTO 1790
1130 IF P9=1 GOTO 1790
1140 IF D>100 GOTO 1170
1150 PRINV "BEAUTIFUL "N1$" YOU WERE "D"METERS FROM THE LANDING SITE"
1160 GOTO 1790
1170 PRINT "GOOD LANDING "N1$". BUT YOU WERE "D"METERS OFF"
1180 GOTO 1790
1190 LET T=F1/(ABS(B)+ABS(B1)+ABS(B2))
1200 LET A=A-(V*((G-B)/2))
1210 LET V=V+(G-B)
1220 LET T1=((-2*V)+SQR(4*V^A+8*A1*G))/(2*G)
1230 LET V=V+G*T1
1240 GOSUB 2010
1250 LET D1=D1+V1*(T1-T)
1260 LET D2=D2+V2*(T1-T)
1270 IF V<3 GOTO 950
1280 LET T4=T3+T1
1290 PRINT "THE "N1$;N2" CRASHED AT T="T4"SEC AT THE POINT ("D2","D1")"
1300 LET R=SQR(V1^2+V2^2)
1310 PRINT "WITH A DOWNWARD VELOCITY "V"AND A FORWARD VELOCITY"R
1320 PRINT "CRASH DUE TO PILOT ERROR (THE IDIOT RAN OUT OF FUEL)"
1330 GOTO 1790
1340 IF ABS(B)<=M GOTO1400
1350 IF B<0 GOTO 1380
1360 LET B=M
1370 GOTO 1390
1380 LET B=-M
1390 LET Z=Z+1
1400 IF ABS(B1)<=M GOTO 1460
1410 IF B1<0 GOTO 1440
1420 LET B1=M
1430 GOTO 1450
1440 LET B1=-M
1450 LET Z1=Z1+1
1460 IF ABS(B2)<=M GOTO 1520
1470 IF B2<0 GOTO 1500
1480 LET B2=M
1490 GOTO 1510
1500 LET B2=-M
1510 LET Z2=Z2+1
1520 LET F=F-((ABS(B)+ABS(B1)+ABS(B2))*T)
1530 IF Z=1 GOTO 1600
1540 IF Z=2 GOTO 1780
1550 IF Z1=1 GOTO 1660
1560 IF Z1=2 GOTO 1780
1570 IF Z2=1 GOTO 1720
1580 IF Z2=2 GOTO 1780
1590 GOTO 1770
1600 IF B=0 GOTO 1550
1610 LET E=E-1
1620 IF E=0 GOTO 1780
1630 GOSUB 1880
1640 PRINT "VERTICAL ENGINE WILL BLOW IN"E" BURNS"
1650 GOTO 1550
1660 IF B1=0 GOTO 1570
1670 LET E1=E1-1
1680 IF E1=0 GOTO 1780
1690 GOSUB 1880
1700 PRINT "HORIZONTAL ENGINE WILL BLOW IN"E1" BURNS"
1710 GOTO 1570
1720  IF B2=0 GOTO 1770
1730 LET E2=E2-1
1740 IF E2=0 GOTO 1780
1750 GOSUB 1880
1760 PRINT "TRANSVERSE ENGINE WILL BLOW IN"E2" BURNS"
1770 GOTO 840
```

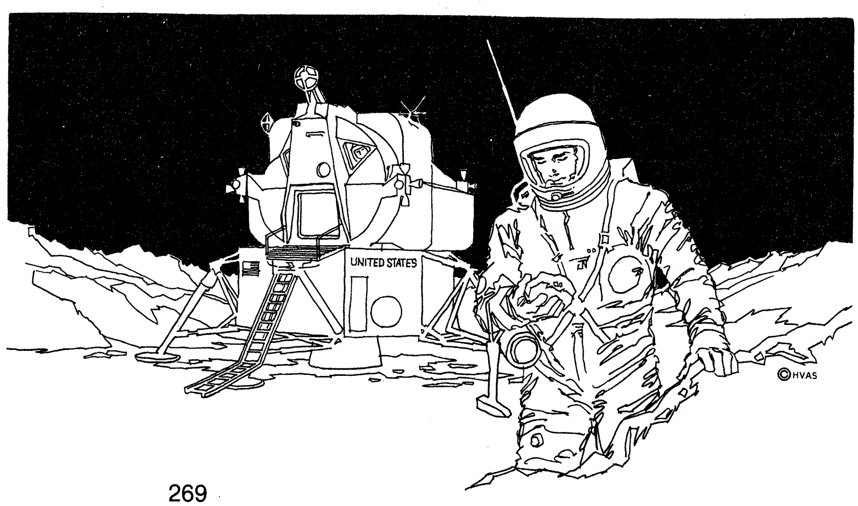


(MORE)

```
1780 GOSUB 1990
1790 PRINT
1800 PRINT "WOULD YOU LIKE TO TRY TO CRASH IT AGAIN STUPID";
1810 INPUT V$
1820 IF V$="NO"GOTO 1870
1830 PRINT "SAME INITIAL VALUES";
1840 INPUT V$
1850 IF V$="YES" GOTO 530
1860 GOTO 320
1870 STOP
1880 PRINT "  ";
1890 RETURN
1900 LET T3=0
1910 LET Z=0
1920 LET Z1=0
1930 LET Z2=0
1940 LET E=11
1950 LET E1=11
1960 LET E2=11
1970 LET P9=0
1980 RETURN
1990 PRINT "BANG!!!!!!!!!"
2000 RETURN
2010 LET V7=V1-B1*T
2020 LET D1=D1+(V1*T)+(T*(V1-V7))/2
2030 LET V8=V2-B2*T
2040 LET D2=D2+(V2*T)+(T*(V2-V8))/2
2050 LETV1=V7
2060 LET V2=V8
2070 RETURN
2080 IF A>100 GOTO 2250
2090 IF D2>100 GOTO 2250
2100 IF D1>100 GOTO 2250
2110 LET Q2=INT(D2/10)+10
2120 LET Q1=-INT(D1/20)+5
2130 PRINT "T="T3
2140 FOR I=1 TO 11
2150 IF I<>Q1 GOTO 2210
2160 IF I=6 GOTO 2190
2170 PRINT TAB(10);"!";TAB(Q2);"X ALTITUDE="A
2180 GOTO 2240
2190 PRINT "----------+----------";TAB(Q2);"X ALTITUDE=";A
2200 GOTO 2240
2210 IF I=6 GOTO 2230
2220 GOTO 2240
2230 PRINT "----------+----------"
2240 NEXT I
2250 RETURN
2260 END
```

SAMPLE RUN

```
THIS IS A LUNAR LANDING PROGRAM.
DO YOU WISH INSTRUCTIONS ?NO
WHAT IS THE INITIAL ALTITUDE ?600
WHAT IS THE INITIAL VERTICAL VELOCITY (DOWN IS +) ?25
WHAT IS THE DISTANCE Y ?200
WHAT IS Y VELOCITY ?-10
WHAT IS THE DISTANCE X ?25
WHAT IS THE X VELOCITY ?1
WHAT IS THE MAXIMUM BURN RATE ?50
WHAT IS YOUR FUEL CAPACITY ?1200
WHAT IS THE GRAVITATIONAL CONSTANT ?.6
WHAT IS THE NAME OF YOUR SHIP ?AQUARIUS

CONTROL TO AQUARIUS COMMENCE LANDING.
TIME =    0 SEC.
ALT=  600  METERS V=   25   METERS/SEC
DIST. X=   25 METERS. V=    1 METERS/SEC
DIST. Y=  200 METERS. V=  -10 METERS/SEC
FUEL= 1200 UNITS
TIME INTERVAL ?5
VERTICAL(Z) BURN ?0
TRANSVERSE(X) BURN ?-.2
HORIZONTAL (Y) BURN ?0

TIME =    5 SEC.
ALT=   473.5  METERS V=   28  METERS/SEC
DIST. X= 32.5 METERS. V=    0 METERS/SEC
DIST. Y=  150 METERS. V=  -10 METERS/SEC
FUEL= 1199 UNITS
TIME INTERVAL ?10
VERTICAL(Z) BURN ?0
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?0

TIME =   15 SEC.
ALT=   190.5  METERS V=   34  METERS/SEC
DIST. X= 32.5 METERS. V=    0 METERS/SEC
DIST. Y=   50 METERS. V=  -10 METERS/SEC
FUEL= 1199 UNITS
TIME INTERVAL ?2
VERTICAL(Z) BURN ?10
TRANSVERSE(X) BURN ?-.2
HORIZONTAL (Y) BURN ?2

TIME =   17 SEC.
ALT=   131.9  METERS V= 15.2  METERS/SEC
DIST. X= 32.9 METERS. V=  -.4 METERS/SEC
DIST. Y=   26 METERS. V=   -6 METERS/SEC
FUEL= 1174.6 UNITS
TIME INTERVAL ?4
VERTICAL(Z) BURN ?33?
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?1.5

T=   21
----------+----------X ALTITUDE= 75.9
TIME =   21 SEC.
ALT= 75.9  METERS V=   5.599999  METERS/SEC
DIST. X= 31.3 METERS. V=  -.4 METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1156.6 UNITS
TIME INTERVAL ?10
VERTICAL(Z) BURN ?.5
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?0

T=   31
----------+----------X ALTITUDE=   19.40001
TIME =   31 SEC.
ALT=   19.40001  METERS V=   6.599999  METERS/SEC
DIST. X= 27.3 METERS. V=  -.4 METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1151.6 UNITS
TIME INTERVAL ?1
VERTICAL(Z) BURN ?5
TRANSVERSE(X) BURN ?-1
HORIZONTAL (Y) BURN ?0

T=   32
----------+----------X ALTITUDE=   15.00001
TIME =   32 SEC.
ALT=   15.00001  METERS V=   2.199999  METERS/SEC
DIST. X= 27.4 METERS. V= -1.4 METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1145.6 UNITS
TIME INTERVAL ?2
VERTICAL(Z) BURN ?.5
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?0

T=   34
----------+----------X ALTITUDE=   10.50001
TIME =   34 SEC.
ALT=   10.50001  METERS V=   2.399999  METERS/SEC
DIST. X= 24.6 METERS. V= -1.4 METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1144.6 UNITS
TIME INTERVAL ?2
VERTICAL(Z) BURN ?.7
TRANSVERSE(X) BURN ?.7
HORIZONTAL (Y) BURN ?0

T=   36
----------+----------X ALTITUDE=   5.800013
TIME =   36 SEC.
ALT=   5.800013  METERS V=   2.199999  METERS/SEC
DIST. X= 20.4 METERS. V=-1.49012e-08    METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1141.8 UNITS
TIME INTERVAL ?2
VERTICAL(Z) BURN ?.8
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?0

T=   38
----------+----------X ALTITUDE=   1.600014
TIME =   38 SEC.
ALT=   1.600014  METERS V=   1.799999  METERS/SEC
DIST. X= 20.4 METERS. V=-1.49012e-08    METERS/SEC
DIST. Y=  -10 METERS. V=-5.96046e-08    METERS/SEC
FUEL=  1140.2 UNITS
TIME INTERVAL ?2
VERTICAL(Z) BURN ?.7
TRANSVERSE(X) BURN ?0
HORIZONTAL (Y) BURN ?0

AT T=   38.78595 THE AQUARIUS LANDED WITH A DESCENT VELOCITY OF
   1.721405 M/SEC
WITH A HORIZONTAL VELOCITY OF  6.14391e-08    M/SEC
THE LANDING POINT WAS AT ( 20.4 ,  -10 )
BEAUTIFUL AQUARIUS YOU WERE    22.71915 METERS FROM THE LANDING SITE

WOULD YOU LIKE TO TRY TO CRASH IT AGAIN STUPID ?NO
```



TWO~TO~TEN

Two-to-Ten is a game of chance played with a special deck of cards with only the cards 2 - 10. The game is similar to blackjack in that you are drawing cards and trying to come as close as possible to a goal number (chosen at random before each round) without going over it. You must come within a certain number of points of the goal number determined by a "lucky-limit" card. The catch to the game is that you are not given the exact value of the goal number but rather a clue that is only within 15% of the goal.

Can you think of a way to make Two-to-Ten more interesting? Perhaps playing it against the computer as an opponent? Let's hear your ideas!

I'm embarrassed to say that I don't remember who originally gave me Two-to-Ten, but if the author will drop a line, I'll credit him or her in the next issue. — DHA

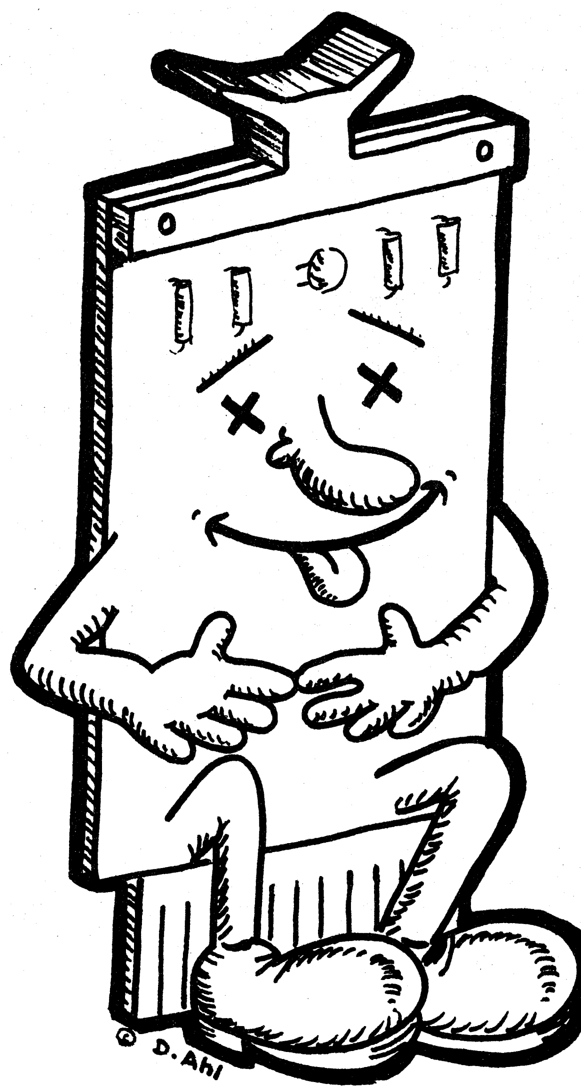
PROGRAM LISTING

```
50 PRINT \PRINT \PRINT
60 PRINT "WELCOME TO THE GAME TWO-TO-TEN.  THE NAME COMES FROM THE"
70 PRINT 'SPECIAL "DECK OF CARDS" USED.  THERE ARE NO FACE CARDS - ONLY'
80 PRINT "THE CARDS 2-10.  THIS GAME IS EASY AND FUN TO PLAY IF YOU"
90 PRINT "UNDERSTAND WHAT YOU ARE DOING SO READ THE INSTRUCTIONS"
100 PRINT "CAREFULLY."
110 PRINT "AT THE START OF THE GAME YOU BET ON WINNING.  TYPE IN ANY"
120 PRINT "NUMBER BETWEEN 0 AND 200.  RSTS THEN PICKS A RANDOM NUMBER
130 PRINT "YOU ARE TO REACH BY THE SUM TOTAL OF MORE CARDS CHOSEN."
140 PRINT "BECAUSE OF THE RARE CHANCE OF YOU GETTING TO THAT NUMBER"
150 PRINT "EXACTLY, YOU ARE GIVEN AN ALLOWANCE CARD.  THE OBJECT OF"
160 PRINT "THE GAME IS TO GET THE TOTAL OF CARDS WITHIN THE MYSTRY
170 PRINT "NUMBER WITHOUT GOING OVER."
180 PRINT "YOU ARE GIVEN A HINT AS TO WHAT THE NUMBER IS.  THIS IS NOT"
185 PRINT "THE EXACT NUMBER ONLY ONE CLOSE.  ALL YOU DO IN THIS GAME IS"
190 PRINT "DECIDE WHEN TO STOP.  AT THIS POINT YOUR TOTAL IS COMPARED"
195 PRINT "WITH THE NUMBER AND YOUR WINNINGS ARE DETERMINED."
197 PRINT\PRINT"GOOD LUCK!"\PRINT\PRINT\PRINT
199 M=200
200 RANDOMIZE
210 D=0\T=0
215 O=INT(10*RND(0))+25
220 N=INT(O*RND(0))+O
230 R=(INT(15*RND(0))+1)/100
250 S=INT(2*RND(0)+1)
260 IF S=1 THEN E=INT(N-(N*R))
265 GOTO 280
270 E=INT(N+(N*R))
280 A=INT(9*RND(0)+2)
285 PRINT "PLACE YOUR BET . . . YOU HAVE $"M"TO SPEND.";\INPUT B\PRINT
290 IF B>M THEN 295 ELSE 300
295 PRINT "YOU CAN'T BET MORE THAN YOU'VE GOT!"\GOTO 250
300 PRINT 'YOUR "LUCKY-LIMIT" CARD IS A'A
310 PRINT "YOU MUST COME WITHIN"A"WITHOUT GOING OVER TO WIN."
320 PRINT \PRINT "HERE WE GO!"
330 PRINT \PRINT
340 D=D+1
350 C=INT(9*RND(0)+2)
360 PRINT "CARD #"D"IS A"C".  YOU ARE TRYING TO COME NEAR"E
365 T=T+C
370 IF T>N THEN 375 ELSE 380
375 PRINT "YOUR TOTAL IS OVER THE NUMBER"N".  AN AUTOMATIC LOSS!"
377 GOTO 570
380 PRINT "TOUR TOTAL IS"T".  DO YOU WANT TO CONTINUE"; \INPUT Q$\PRINT
390 IF LEFT(Q$,1)="Y" THEN 330
410 IF T>=N-A AND T<=N THEN 500 ELSE 550
500 PRINT "YOU WIN !  THE NUMBER WAS"N".  YOUR GUESS TOTAL WAS"T"WITHIN"
510 PRINT "YOUR LIMIT CARD."
520 M=M+B
540 GOTO 600
550 PRINT "YOU BLEW IT!  THE NUMBER WAS"N", OUTSIDE YOUR ALLOWANCE BY";
560 PRINT (N-A)-T \PRINT
570 M=M-B
600 PRINT "YOU NOW HAVE $"M"IN CASH TO BET IN THE NEXT GAME!"
610 INPUT "WOULD YOU LIKE TO PLAY THAT NEXT GAME";Q$\PRINT
620 IF LEFT(Q$,1)="Y"THEN 200
630 PRINT "HOPE YOU HAD FUN"
999 END
```

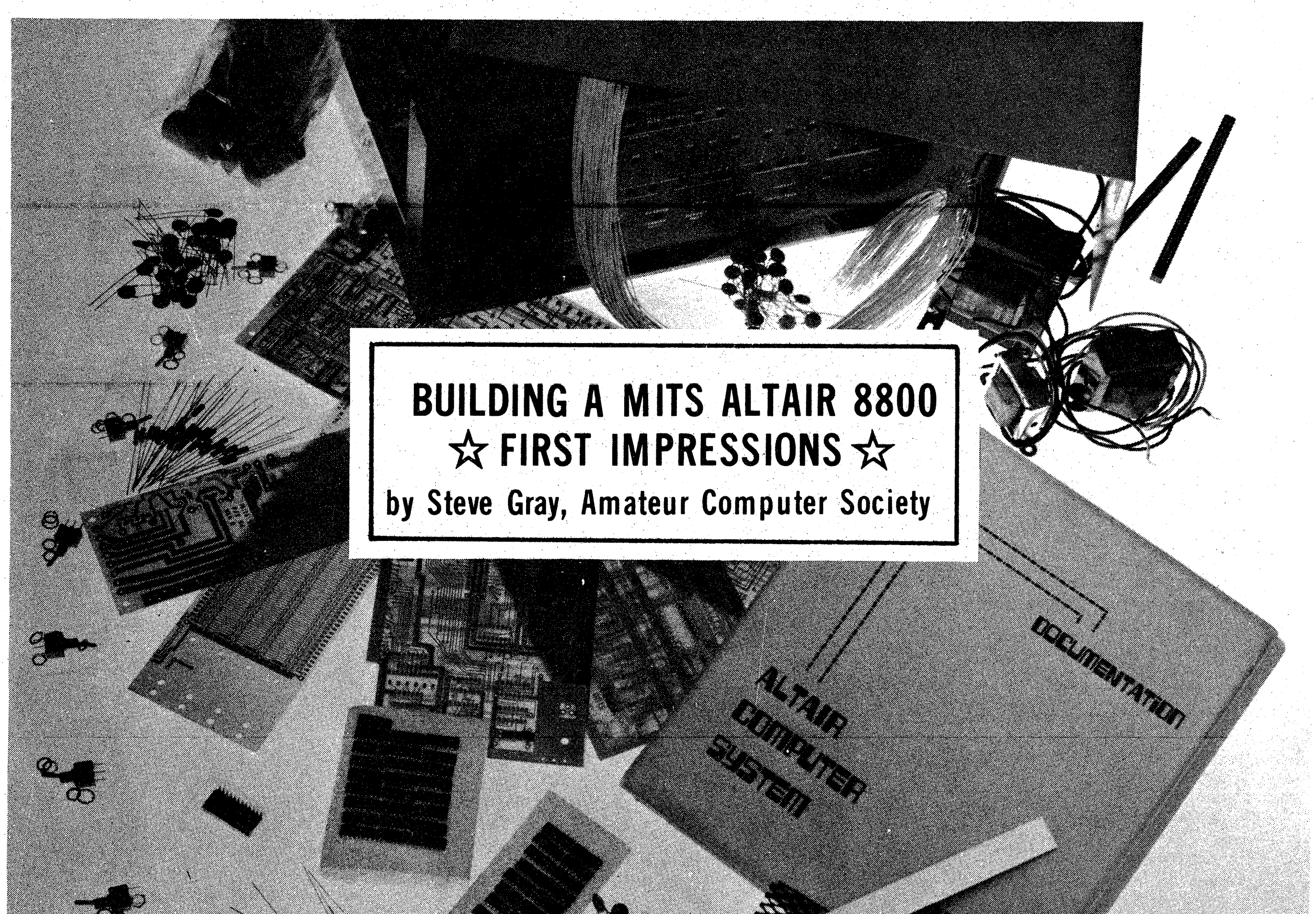
SAMPLE RUN

```
RUNNH

WELCOME TO THE GAME TWO-TO-TEN.  THE NAME COMES FROM THE
SPECIAL "DECK OF CARDS" USED.  THERE ARE NO FACE CARDS - ONLY
THE CARDS 2-10.  THIS GAME IS EASY AND FUN TO PLAY IF YOU
UNDERSTAND WHAT YOU ARE DOING SO READ THE INSTRUCTIONS
CAREFULLY.
AT THE START OF THE GAME YOU BET ON WINNING.  TYPE IN ANY
NUMBER BETWEEN 0 AND 200.  RSTS THEN PICKS A RANDOM NUMBER
YOU ARE TO REACH BY THE SUM TOTAL OF MORE CARDS CHOSEN.
BECAUSE OF THE RARE CHANCE OF YOU GETTING TO THAT NUMBER
EXACTLY, YOU ARE GIVEN AN ALLOWANCE CARD.  THE OBJECT OF
THE GAME IS TO GET THE TOTAL OF CARDS WITHIN THE MYSTRY
NUMBER WITHOUT GOING OVER.
YOU ARE GIVEN A HINT AS TO WHAT THE NUMBER IS.  THIS IS NOT
THE EXACT NUMBER ONLY ONE CLOSE.  ALL YOU DO IN THIS GAME IS
DECIDE WHEN TO STOP.  AT THIS POINT YOUR TOTAL IS COMPARED
WITH THE NUMBER AND YOUR WINNINGS ARE DETERMINED.

GOOD LUCK!

PLACE YOUR BET . . . YOU HAVE $ 200 TO SPEND.? 50

YOUR "LUCKY-LIMIT" CARD IS A 5
YOU MUST COME WITHIN 5 WITHOUT GOING OVER TO WIN.

HERE WE GO!

CARD # 1 IS A 10 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 10 .  DO YOU WANT TO CONTINUE? YES

CARD # 2 IS A 4 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 14 .  DO YOU WANT TO CONTINUE? YES

CARD # 3 IS A 7 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 21 .  DO YOU WANT TO CONTINUE? YES

CARD # 4 IS A 6 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 27 .  DO YOU WANT TO CONTINUE? YES

CARD # 5 IS A 9 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 36 .  DO YOU WANT TO CONTINUE? YES

CARD # 6 IS A 5 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 41 .  DO YOU WANT TO CONTINUE? YES

CARD # 7 IS A 5 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 46 .  DO YOU WANT TO CONTINUE? YES

CARD # 8 IS A 4 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 50 .  DO YOU WANT TO CONTINUE? YES

CARD # 9 IS A 9 .  YOU ARE TRYING TO COME NEAR 54
TOUR TOTAL IS 59 .  DO YOU WANT TO CONTINUE? NO

YOU WIN !  THE NUMBER WAS 61 .  YOUR GUESS TOTAL
WAS 59 WITHIN  YOUR LIMIT CARD.
YOU NOW HAVE $ 250 IN CASH TO BET IN THE NEXT GAME!
WOULD YOU LIKE TO PLAY THAT NEXT GAME? NO

HOPE YOU HAD FUN

READY
```

Hardware



Freaked-out Freddy Flip Chip



BUILDING A MITS ALTAIR 8800 ☆ FIRST IMPRESSIONS ☆

by Steve Gray, Amateur Computer Society

Knocking the biggest is an old tradition. Today, IBM not only makes as much profit as all the other computer manufacturers combined, it also receives as much criticism as all the others put together. Some of the judgments are deserved. Still, IBM must be doing something right. Even though IBM computers aren't the fastest or the best, the company has built up a reputation for service and support that competitors find hard to beat. Detractors say that computer buyers are preconditioned by IBM's past record to accept IBM as the greatest.

MITS has sold thousands of the Altair 8800 microcomputer kits, many more than all the other kit-makers combined, and has been widely criticized in the hobbyist newsletters for late delivery, for delivery "in bits and pieces," for poor design, and for an inadequate manual. All true, yet the Altair continues to outsell all other kits. How long this will last, however, is a big question. Although MITS may be said to have gotten there "firstest," new companies are appearing with kits that may turn out to be the "mostest," and which may become bestsellers. But no matter how good a kit a competitor may have, he'll have to go a long way to make it as well-known as the Altair, which is heavily advertised, especially in technical trade magzines, several of which are bound—in the complete four-color, 24-page 8800 brochure.

Building the Altair 8800

Starting to build the 8800 is pretty much like starting a Heathkit radio receiver. There are instructions telling how, and a drawing that shows where, to install the ICs (integrated circuits), followed by similar sections on installing the resistors, capacitors, etc. The MITS manual is not as detailed as a Heathkit manual, nor would one expect it to be. However, there are some problems in building the 8800, sometimes due to information lacking in the manual, sometimes due to what seems to be poor mechanical design, and sometimes because of a lack of thoroughness by the manufacturer in checking out the parts lists, the parts supplied, and having the two match up. For instance, some of the solder-pads for capacitors are so close to other pads, or to printed wiring, that a solder bridge is all too easy to make. The installation of the heat sinks calls for silicone grease, yet the parts list doesn't mention any. So you go buy some, and after finishing the kit, and checking out the parts left over, you find that what looks like a bit of white paint, in a small capsule, is the heat-sink grease you needed. There are other leftover parts that will make the builder wonder if he left anything out. Each front-panel switch comes with two nuts, lock washer, and positioning ring. Yet the drawings show only a nut used with each switch. What about those other parts?

Anyone building an 8800 should first learn to tell the difference between 8-32 and 6-32 screws, and between #4 and #6 washers. Even after he does, he'll have trouble with the screws, nuts and washers, because what is supplied doesn't match up with what's called for.

Questions will arise that most builders won't be able to answer for themselves, and which they will probably have to write (or call) MITS about. For instance, "Is there any difference, on the 4K memory boards, between the SN74LO4 ICs called for, and the SN7404 ICs provided?"

The 66 Wires

One of the biggest problems, familiar to every 8800 builder, is that of the 66 three-foot wires, which must be soldered, one by one, into holes around the edge of the Display/Control board. No matter how careful the builder is, several are bound to pop off; if not at this point, then later when the other ends of the wires are soldered to the expander board. MITS eliminated this problem in their new kit for the Altair 680 (based on the Motorola 6800), in which the computer board plugs directly into a connector on the front-panel board.

The various resistors and capacitors that have to be outboarded on several 8800 boards seem to indicate that this Altair wasn't fully field-tested before the design was frozen.

Sooner or later, in the middle of constructing the 8800, the builder realizes he should have read the entire manual before soldering a single component, so that some of the required modifications can be made at early stages. Thus he might prefer to solder the ¼-inch braid to the bottom of the power-supply board before screwing this board to the chassis. Not that it's impossible to do later on, but it's not easy at the point it comes up in the manual, when there are so many wires hanging on it, and it must be put back into place with all those spacers and screws.

Programming the 8800

After finishing the 8800, the builder is given a simple addition problem to enter into the memory and run. But after he enters the program, and proofreads it, he may find—if his 8800 uses 4K memory boards with Signetic 2604 RAMs, many of which "do not meet the required specifications for access time and refresh period," as the MITS newsletter puts it (in recalling all such memory boards)—that various bits will drop out, and the program will not run correctly. This is somewhat disheartening, even if he can make the program work by relocating it in an unaffected part of memory. Only one other program is provided in the manual, a multiply program.

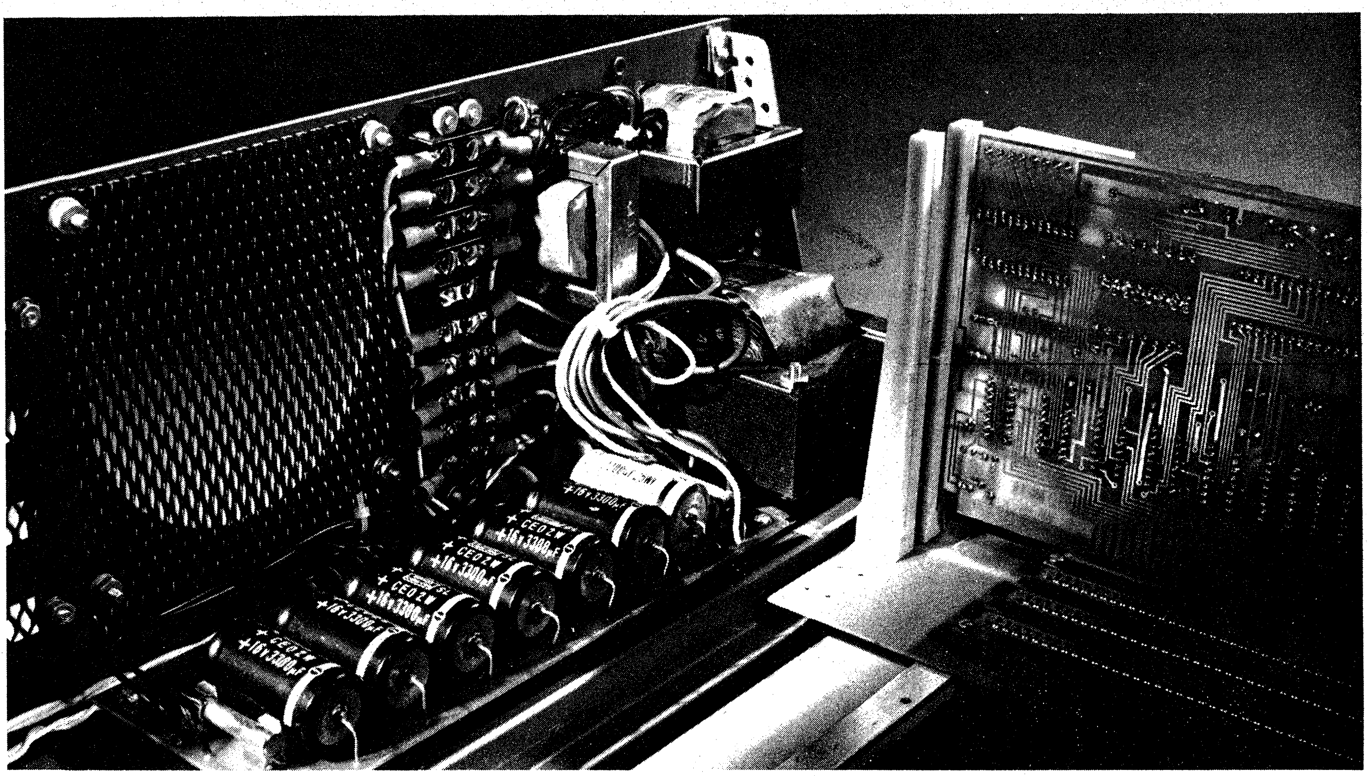
Checking out the finished 8800 requires a great deal of flipping the front-panel switches up and down. The switch-handles are so short, protruding less than 3/8 inch beyond the panel, that the front-panel lettering can soon be worn off, and one's nerves soon worn down as well. Longer handles, preferably with flattened ends, should have been used.

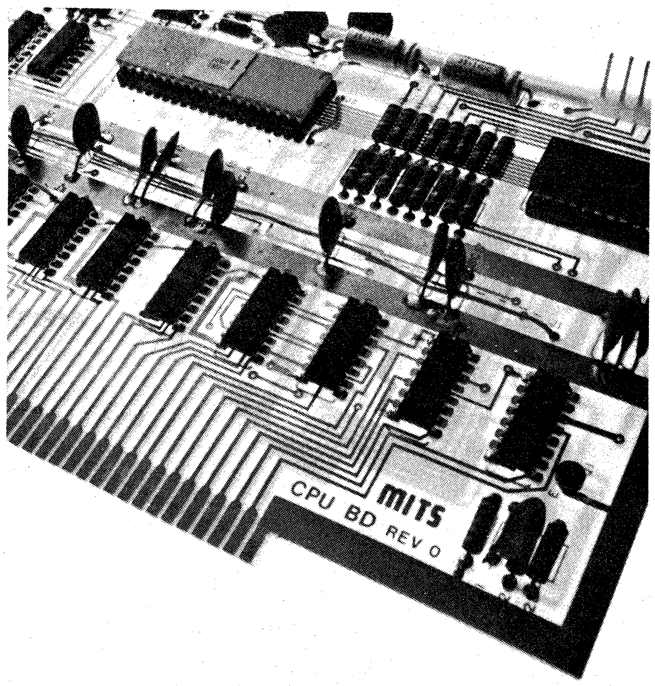
The 8800 Manuals

Let's take a closer look at the manual, or as MITS puts it, the set of manuals. The 82-page Assembly Manual is fairly well done, and shouldn't give much trouble to anyone with even a minimum of kit-building experience. The problems are more with assembling the parts, rather than with the instructions themselves, such as the 66 three-foot wires that break off so easily, the LEDs that somehow must be made to protrude just the right distance from the Display/Control board, the jungle of wires in the power supply, etc.

The "Theory of Operation Manual & Schematics" more closely resembles the set of notes a design engineer might make, and then hand to a technical writer with the request to "make a manual out of this." Only a computer-design engineer, or a hot-shot technician, could get a good understanding of how the 8080 works, from these 16 pages of text and a half-dozen schematics. The most important ICs are shown as blank boxes, without any explanation of what goes on inside: the 8080 MPU, 8212 (system status latch), and the memory ICs.

The discussion takes much for granted: "Note that the PRDY and PHOLD signals are synchronized to the leading edge of the Ø2 clock," although it isn't at all that obvious to most of the readers. The entire discussion of the CPU is on two pages, and that's all the Display/Control board operation gets. The Power Supply operation is covered in ten sentences. These pages tell a little of the *what* and *where,* but little or nothing about the *why.* There is no discussion of the CPU or Display/Control on-board regulator schematics.





The Operator's Manual takes up 92 pages. The Introduction consists of three pages of Boolean logic and truth tables, two on electronic logic (gates), a paragraph on number systems, three short pages on the binary system, a page of generalities on computer programming, two pages on a simple program for adding two numbers, and a page on computer languages.

The part on Organization of the Altair 8800 includes a page with a block diagram but without any explanation whatever, five pages on the CPU, 2/3 of a page on memory, three sentences on the clock, and a third of a page on Input/Output. One sentence notes that "subtraction and division are implemented by inverse addition," yet inverse addition is left unexplained, by word or example; not even binary addition is shown. Figure 2-3, The Working Registers, is different enough from Figure 2-2, the CPU diagram, to cause confusion. Only a few of the CPU blocks are explained, leaving many others to the reader's imagination.

One Part in Detail

One part of the manual is explained in detail, telling how to load a simple program that adds two numbers, using LDA, MOV, ADD, STA, and JMP. The meaning of each mnemonic is explained, then the mnemonics are repeated along with the bit patterns they represent. This is then repeated, with octal equivalents of the bit patterns. The reader is then shown how to enter the program with the front-panel switches, all very clearly. A short section, 2/3 of a page, tells how to store the numbers to be added, in detail.

This whole section is written in the detail that the whole manual should be, with all the explanations the reader should be given. But perhaps this detail is given here because this is the most important part of the entire manual; without such detail, the builder can't figure out how to run a simple program, and thus won't be able to use the computer he's spent weeks or months abuilding.

The only other program in the entire manual is for binary multiply, unaccompanied by a single word of explanation.

The single page on Memory Addressing describes, in several sentences each, the various addressing modes for the 8800. The page of Operating Hints covers three items: proofread your programs; scatter NOPs through a complicated program so that new steps can be added if necessary; for debugging, use the SINGLE STEP switch and observe the status LEDs.

The rest of the manual, 45 pages, covers the instruction set. For each, the manual gives the mnemonic, bit pattern, what operation is caused by the instruction, what status bits are affected, and sometimes a brief example. The example assumes some knowledge of programming, as does the paragraph on operation. For DI, the operation paragraph reads, "Implementation of the DI instruction resets any interrupt flip-flop. This causes the computer to ignore any subsequent interrupt signals."

Again, this explains what and where, but not why. There are no programs, other than the two brief ones for add and multiply, to demonstrate the use of all these instructions, as there should be to help the reader write his own programs with them. Many readers won't be able to progress beyond using only a few of the basic instructions, unless they get some help, either in the form of sample programs, or a text on the use of this (or a similar) instruction set.

These 45 pages are fine for a programmer already familiar with assembly language, but the beginner will be almost totally lost.

An appendix of 5½ pages condenses the instruction set into tables.

Memory for the 8080

Since some memory is needed to make the computer work, let's take a look at the documentation for the 4K RAM board. The 2¼ pages on Theory of Operation explain the refresh operation, with schematic references, telling *what,* but not *why,* other than to say, after several paragraphs about which pulse goes where, that "the output of this circuit provides a 500 nanosecond, +12 volt pulse to the RAM IC's to accomplish the required access."

The errata sheet says diode D2 will be a 1N746A 3.3 volt zener, unless "your kit is supplied with the Intel C2107A's," in which case "this diode will be a 1N4733, 5v zener instead." Mine was a 1N4742A.

The last page contains a modification to the power supply, to add a paralleled pair of resistors, but nothing at all on why.

The "first impressions" must end here, because I'm waiting for MITS to return the 4K memory boards, and to send some software. Next time, we'll look into the programming, both in assembly language and in BASIC, and into more of the hardware, including interfaces.

Despite the Knocks

Despite all that can be said against the Altair, the 8800 is well on its way to becoming the best-known and most-used microcomputer kit of the decade. It may not be the best (the Sphere seems to have many advantages) nor the best supported (the Scelbi provides much more software help, in print), but it is today's bestseller. And unless some other company can fight this already well-established computer kit with a bigger and better advertising campaign (and not necessarily with a better computer kit), then the 8800, the 680 and future Altair computers will give MITS the micro-equivalent of IBM's continuing and overwhelming success.

BUILDING A MITS ALTAIR 8800 GETTING A SYSTEM TOGETHER

Richard Kuzmack
Chesapeake Microcomputer Club, Inc.
1435 Layman Street
McLean, Virginia 22101

In this year's first issue of Creative, *Vol. 2, No. 1, we published Steve Gray's "Building a MITS Altair 8800: First Impressions." It is only fitting that in this last issue of the year we follow-up on that article with an account of interfacing-to and using the Altair 8800 once built. We asked Rich Kuzmack, President of the Chesapeake Microcomputer Club, if he'd write the follow-up for us, and this is his account.*

Once you have completed and checked out the assembly of your Altair 8800 main frame kit you are ready to develop a usable system by adding memory, input/output interfaces, and the input/output devices themselves. This is a good point to review your system plan, or start one if you have neglected this step thus far. Even if you are just beginning to think of maybe getting a personal computer of your own, stay tuned. In this article we will explore together the kinds of decisions you'll probably have to make in planning your system and the considerations you'll want to keep in mind as you make them.

The process of making a system plan usually consists of deciding how to get the most out of what you have the least. The money that you are able to devote to your system is most often going to be the limiting factor, although some will find the availability of time to work on it even more limiting. Neither is necessarily an absolute limit once you realize that you can substitute some patience for some of either one. It may take you longer to save up the money or assemble a board, but over the long haul you will have fewer regrets and headaches if you take the extra time required to avoid low performance components and hasty construction practices.

The other side of the system planning process makes you think about what you want to do with your computer. Indeed, there is a certain logic to planning what to put in to a system based on what you want out of it, but many computer hobbyists have not been especially concerned with this question. So let's not address the end use, but think instead in terms of the high level programming languages you'd like to have running on your system. Several are available for your Altair's 8080 CPU, and the amount of memory and I/O needed to support a particular language can provide you with a first cut at some goals for planning your system. A popular starting point that makes a lot of sense is the BASIC language because it can start small and grow with your system, from Tiny BASIC to the 4, 8, and 12K versions available from MITS and others. Even if you want to eventually be running FORTRAN or APL, one of the smaller BASICs is a good idea for a start. It will come in very handy later on for that large applications program that won't fit after one of the larger language interpreters soaks up its share of the available memory space.

Getting In . . . and out

Two kinds of input/output will be needed before your system is ready to run. First, some means to get to and from a non-volatile mass memory medium, such as audio cassette or paper tape, will be required to take advantage of available machine-readable software and to provide a way to save software you've written or modified yourself. Second, you will want a keyboard for program and data entry, and an output display of some kind. There are many possible solutions to performing these functions because of the many products on the market today that were designed to work in your Altair, and because of the variety of both new and used equipment originally intended for commerical systems.

In deciding what to get for my own system an important factor was the flexibility that could be squeezed out of each board. A board that can be made to serve more than one use saves both time and money that might otherwise go to buying and building another board and adding another slot (maybe even another mother board, too!). I made my decision and purchases several months ago and am quite pleased with my choices, but your circumstances and goals may be very different from mine and there is certainly more to select from now. While I hope the specifics of my system are helpful, you'll have to extract and modify to suit your own needs.

The input/output interfaces in my system consist of three boards: the MITS four port parallel board, the Processor Technology video display module (VDM-1), and the MITS audio cassette recorder board. The advantage of this configuration is that two of the three devices to be interfaced are already available in most homes: a black-and-white television set and a cassette recorder. Only the keyboard has to be bought, and one that is encoded for ASCII (American Standard Code for Information Interchange) can usually be had for under $50 on the surplus market, while keyboards with other encoding schemes cost a lot less but might have to be translated to ASCII in software. (It only takes 512 bytes to go both ways for an 8-bit code, but if you don't want to spare the memory, stick to an ASCII keyboard.)

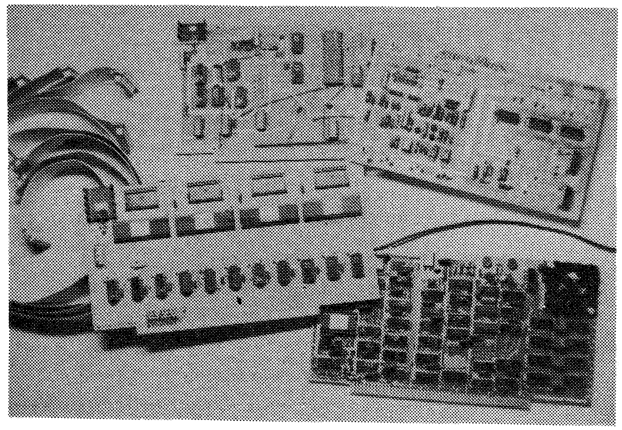
The parallel board was purchased initially with just one port to connect to a surplus keyboard. Using the Motorola 6820 Peripheral Interface Adapter (PIA) chip, it is an extremely simple, tremendously versatile piece of hardware. But there's a catch — in order to get the thing to work at all you have to configure the little beast with software that is more than a little complex. I thought it was my own stupidity that was the cause of my problems with it until I started getting phone calls from other members of our local computer club who were having the same kinds of trouble. The inadequacy of the documentation that came with the kit was eventually recognized by the presentation of expanded documentation in Computer Notes. Hopefully the explanation supplied with the kit has also been revised by now.

The Video Display Module presented the opposite problem. The instructions for using it were quite adequate, but the assembly manual is confusing in the great amount of detail concerning intermediate checks and tests, some requiring an oscilloscope which I don't have. So instead, I simply put the board together carefully without stopping for any of the intermediate checks, and winding up with a board that seemed to work fine. I attributed the difficulty I was having adjusting the picture to the TV set I was using until I found out that two diodes had been left out of the kit and the assembly manual. Installing the diodes solved the problem, and the current version of the VDM has corrected this ommission.

Modifying an old TV set to work with the VDM seemed beforehand like it would be a fairly formidable task, but following the guidance provided in Don Lancaster's *Byte* article on television interfacing (reprint provided with kit) proved to be quite straightforward. Getting the parts and the schematic for an old TV delayed things somewhat. The Sam's Photofact for *my* set was temporarily out of stock, as usual, and there is no point in even opening the set's cabinet until you've studied the circuit diagram and the component layout. The few parts that are needed were not available in the indicated type numbers at several parts sources, but after running a few poor stockmen ragged I was able to get equivalent substitutes for one of the three suggested circuits. A useful technique in such situations is to have a list of part numbers for each alternative made up ahead of time and when one of them can be filled completely note down beside each part the designation on the substitute part. That way when you get home you won't have trouble figuring out what was a substitute for what.

The audio cassette recorder interface consists of two separate circuit boards: a TTL-level serial I/O port and a modem (modulator-demodulator) board. Assembly of the boards presents no particular difficulties. The assembly instructions, however, say to mount the two of them together, but that way the unit will take up the space of two slots inside your Altair case. Then, too, I had another reason for not mounting them together. For audio cassette recording and playback the modem at that time operated on the same 2025/2225 Hertz band that a standard "103" style originate modem uses to receive data. Of course, if the modem will only be used with magnetic tape, just the 2025/2225 band is needed. However, I also wanted to use mine as a 103 data set so for this it had to be able to send on 1070/1270 Hertz and receive on 2025/2225 Hertz.

The ACR modem has the capability to do both jobs by changing the six jumper wires that control the modulator frequencies. To have both a standard 103 originate modem and an audio cassette recorder modem all in the same board (but not both at the same time) simply replace those six jumpers with a non-shorting six pole, two position rotary switch, connecting the numbered row to the "B" row for audio cassette operation and to the "A" row for use with an acoustic coupler to your telephone handset.



This dual capability of the modem board is no accident, by the way. It is used as a standard 103 style originate modem in the MITS COMTER 256 terminal. However, in order to make the modem less sensitive to the speed variations of low quality cassette recorders MITS announced a modem modification changing the width but fortunately not the center of the operating frequency band. That modification would make it much more difficult to use the same modem board for both applications. As I was having no trouble with speed variation on my cassette recorder I decided not to implement the modulator portion of the change, but only to widen the lock range on the phase locked loop of the demodulator. Thus I can read tapes written with either the old method or the new method, plus easily switch over to full 103 style operation. The only wrinkle that has been introduced into my plan by the modification is the potential problem that might come up if I wanted to give a tape written on my system to someone using a poor quality recorder, but even that would only require a brief loan of my recorder to get the data into the other system one time to then be dumped out using that other system's equipment alone.

". . . you forgot to tell him it's an antique computer. He thinks it's ours . . ."

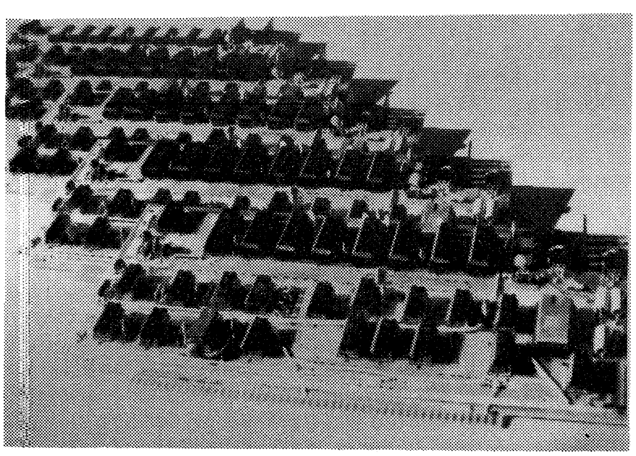
Deciding About Memory

Some of the toughest decisions you'll face with your system will have to do with internal system memory. You have more different memory boards to choose from than any other type of board. Most of this memory will be random access read/write (RAM), but at least some read-only memory (ROM) can save you a lot of work each time you turn the power on. Programmable ROM of the erasable variety is the best bet in the long run, because you can count on wanting to make changes from time to time. But it is RAM that requires the greatest care since it is needed in relatively large amounts.

Cost is certainly a most important factor, but do not neglect the relative advantages and disadvantages associated with the options you have in access speeds, board densities, and rates of power consumption. In some cases you may also want to give special consideration to some of the design features which may or may not be available on particular memory boards. These features include battery backup capability, wait state flexibility, switch selectable addressing, and memory protect features. When you get to the bottom line, however, you'll see that what really drives the decision process is your answer to the question, "How much memory do I want in my system when I'm all done?" You will also want to consider, "How will this memory board perform if I someday want to upgrade my ALTAIR from the 8080 CPU to some new processor chip?" (Selected chips of the Z-80, for example, can run at a clock rate of 4 MHZ, which means memories with access times longer than 250 nanoseconds would need a wait state to work in such a system.)

Well, there it is, again! We're back to the need for a good system plan, including a pretty solid idea of the use to be made of the system. But how else can you make an intelligent decision? If you'll need a lot of memory you had better plan to conserve power consumption, while a smaller system memory would allow you to use some of the really low cost but power-hungry memory boards. Then, too, memory can be added in increments of either 4, 8, or 16K bytes with readily available kits, and although the 4K boards may be easier to afford, the higher density boards may be a better buy in terms of cost and power consumption per byte. They will certainly take fewer slots for the same total memory size, and adding slots to your ALTAIR takes both time and money.

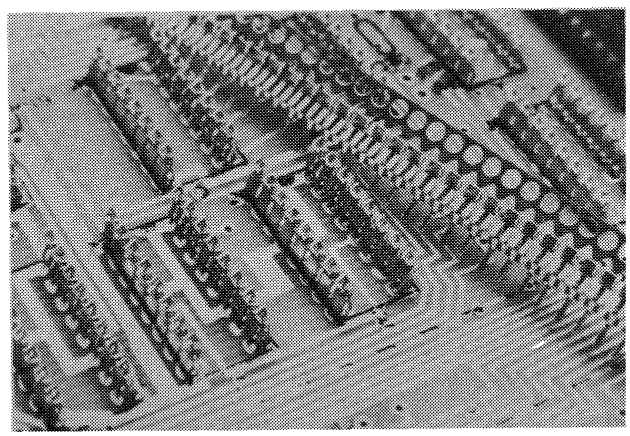
In my own system I decided in favor of high speed memory and started off with 20K of dynamic RAM in five 4K boards from MITS. There was a prolonged delay in getting them up and running because the boards had to be recalled for factory rework and replacement of defective parts. In the meantime MITS lowered its price on that kit and gave a substantial credit on a board-for-board basis for earlier purchases. So even though I could have done better with higher density boards now available based on the regular prices, the credit tilted the decision in favor of five more 4K boards.



Yes, that's a lot of memory, but my system plan includes applications needing that much. In particular, I have become very fond of the APL programming language, including an extensive library of programs that I've written over the past few years. When APL becomes available for the 8080 it is expected to require 24K bytes for the interpreter, on top of which memory will be needed for user workspace. Since APL has the greatest memory requirement of the uses I have for my system, it serves to determine the amount of memory I need.

Plan for Repairability, Too

As you implement your system plan you will be pouring in a lot of money and you can expect it to work for a long time before any of its parts fail. But eventually something's going to go bad, and it is now while you're putting it together that you should allow for the future event. My plan for repairability consists of never soldering an integrated circuit directly into a printed circuit board. I use sockets supplied with the kit and socket strips for all other ICs. If a board doesn't work because of a bad IC somewhere, it is then quite easy to pull and replace one IC at a time until the culprit is found.



I have used over 5,000 of these strip sockets and recommend them on three grounds in addition to the fact that I've never had any problems with them. First, they are relatively inexpensive when purchased in strips of 1,000, costing about a penny a pin. Second, they allow you to get at the printed circuit traces under the IC, which regular sockets don't. This feature saved me a lot of trouble when MITS announced that a trace had to be cut that was under the memory protect flip-flop on my 4K boards. Third, while they may not last through as many insertions and removals as a regular socket, if it becomes necessary to replace them it's easy because each is a separate piece to be unsoldered. Removing a regular socket would be as difficult as removing an IC.

Finally, while I have you thinking about repairs, you should make a simple addition to your ALTAIR that will lessen the likelihood of its needing repair. Add a metal oxide varistor (GE-750) across the power line at the terminal block to protect your system from damage due to high voltage transients. With hundreds or thousands of dollars of hardware inside your ALTAIR case, you can hardly afford not to invest the three dollars or so that it takes for this protection.

A Creative Computing Equipment Profile . . .

ODYSSEY VIDEO GAMES

by David H. Ahl

Ah, the fascinating path of product development. The original Odyssey game from Magnavox (1972) didn't beep. It didn't even keep score. It was expensive (in the multi-hundred dollar range). It came with overlays to put on the face of the TV set and little program cards which allowed you to play a fair number of games — skiing, submarine chase, shooting gallery, and, of course, tennis/ping pong.

However, it wasn't nearly as good as Atari's Pong (the original one to appear in taverns, arcades, and shopping centers). Magnavox responded by filing a suit against Atari claiming that they (Magnavox) held the patents on ping-pong/tennis ball bouncing algorithms for video displays. Panic in Atari-land until it was determined (with the aid of yours truly) that CRT ball-bouncing algorithms had been around since 1957 at places like MIT and CMU and other homes of computer hackers. Having won their case, Atari added insult to injury by putting together a home version of Pong, lining up Sears to distribute it, and pow(!) the battle for the home video game market was off and running.

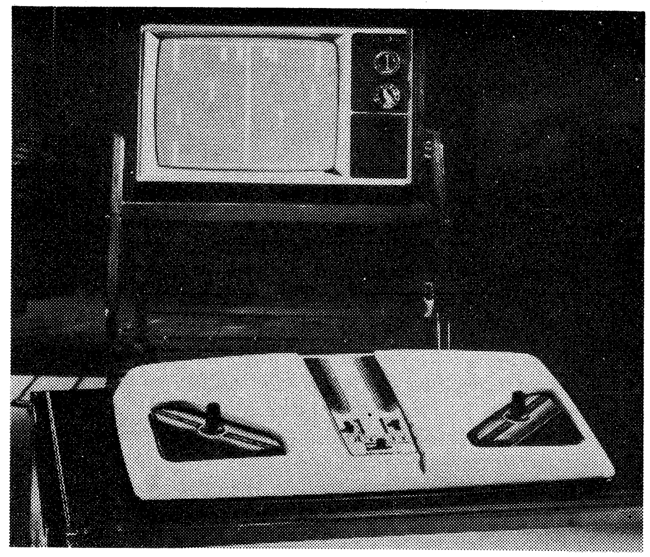
Magnavox, not to be outdone, introduced the Odyssey 200. In all fairness, I should point out that this product would have been introduced whether or not the suit was won. The 200 was, in most ways, an improvement over the first Odyssey. No overlays, no program cards, and it kept score and beeped. However, it had only three games — tennis, hockey, and "smash."

Tennis is the standard CRT tennis. Hockey has a wall on each side of the screen with an opening in it. The puck bounces on the wall; the object is to get it in the opening (goal). Smash is a wall ball game. There can be only two players, although in tennis and hockey each player can control a main paddle and a second, drone paddle (in tennis it's a doubles partner, in hockey it's a goalie).

A significant improvement over earlier TV games is the horizontal as well as vertical paddle control as well as ball control. Once you hit the ball, you can control its vertical path until it is hit by the opposing player. Sports purists will object that this is not an accurate simulation of the real thing, however, it makes for a much more interesting game. And why shouldn't electronic games be different, or even unique (heaven forbid)? There is also a ball speed control.

Because each player has three controls and only two hands, (at least those from this planet), the game can be very exciting or very frustrating. Since there are essentially two variables and one constant per player, there are many possible playing strategies. It is conceivably possible to have two or three players per side. This is not particularly satisfactory for casual play although things improve after a few beers and/or as team-mates get to know each other's moves.

This year there are three Odyssey games. The 300 is a basic unit with Hockey, Tennis and Smash. It has only one control and drones (goalies) only in hockey. The main additions this year seems to be an automatic serve and 3-position skill switch. It keeps score, beeps, and has automatic English. Retail price is around $69.



Odyssey 400 is essentially an updated 200 as described above with mainly an addition of automatic serve. Retail price is around $100. However, if you don't need automatic serve, you might want to look for a 200; in this area they're being heavily discounted to make room for the newer 400.

Odyssey 500 is the top-of-the-line model with colored playing fields and players. It also has a fourth game, soccer.

Magnavox loaned us an Odyssey 300 for testing. It seemed rugged and, as they say in *Road and Track*, the controls fell nicely to hand. It connected to the VHF antenna terminals easily. The only pain was finding 6 "C" batteries to make it work (my DC power supply was busy elsewhere). Since the unit was a loaner I didn't explore the insides too extensively to see if any interesting mods were possible. I wouldn't encourage this type of thing anyway; it doesn't use a microprocessor or memory so it wouldn't have much general purpose versatility. (Please don't take that statement as a challenge to prove me wrong.)

Summary: if you like eye-hand coordination games and have $70 (or more) burning a hole in your wallet, Odyssey might be opere pretium.

If you're a comparison shopper, you should be aware that Atari/Pong now have four games on the market this year starting with the bottom-of-the-line PONG IV ($65) and ranging to a 4-player, 5-game super model ($100) available through Sears. J.C. Penney has still another model ($60). Unisonic's Tournament 2000 plays 6 games and includes a separate electronic pistol ($88 in K-Mart). Also several electronics parts vendors are selling kits in the definitely non-bargain price range of $50 - $90.

A Creative Computing Equipment Profile . . .

HEWLETT PACKARD HP-25 CALCULATOR

"The Minimum Computer"

by James Blodgett

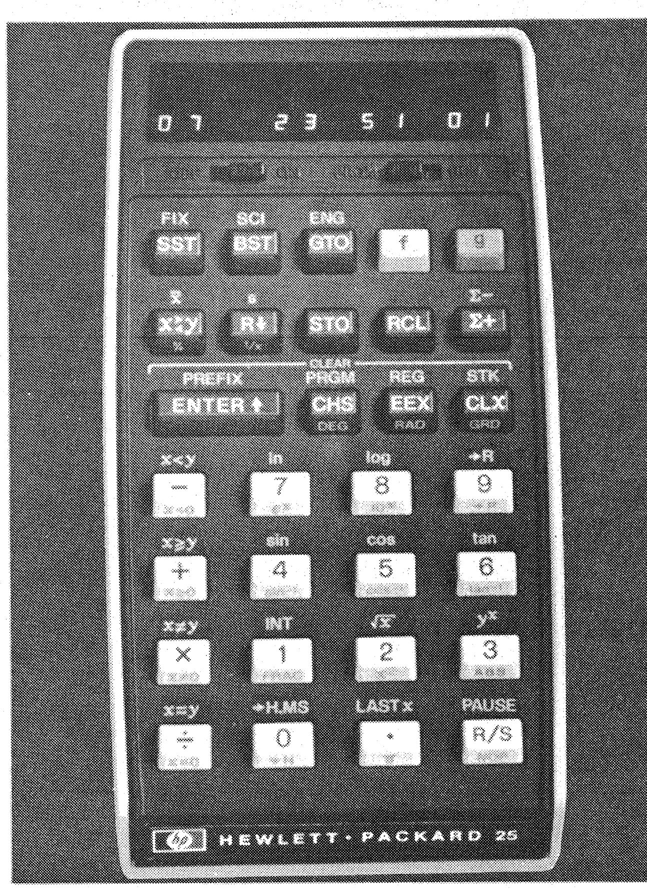
The Hewlett-Packard HP-25 programmable calculator is the cheapest complete computer system on the market today. This statement may raise debate. For one thing, I am saying that the HP-25 is a computer while several cheaper programmable calculators are not. Second, despite its simplicity I am saying that it is a complete system. Both points are matters of definition and debate, but my definition makes good sense.

What, after all, makes a computer? At what level of complexity does a programmable calculator become a computer? At present common usage gives no precise distinction. But since loops, nested loops, and alternate sequences of operations are fundamental to programming, I would like to suggest that a system is a computer if it can do them, and is not otherwise. In general, this means that it must be able to do some form of conditional branching. Besides, I have tried programmable calculators without conditional branching, and they are simply not much fun. They can do only one short sequence of steps in one way, so the complexity of their programs is strictly limited. Conditional branching allows complexity. And the HP-25 is the cheapest programmable with conditional branching.

(This is not to say that a cheaper programmable might not be the best buy for many people. For some practical applications, a little programming goes a long way. And on some models, it is possible to simulate simple loops by pushing some version of the "run" button over and over.)

Another possible point of contention is my calling the HP-25 a complete system. I call it complete because it computes, by itself, and it is all you have to buy. At this writing (Feb '76) there are more ambitious computers advertised in the price range of the HP-25, but they are by no means all you have to buy. For example, the Altair 680 CPU board in kit form lists for the same price as the HP-25 at retail ($195), although the 680 is not yet discounted while the HP-25 is. But the 680 CPU board is a "complete computer" only to those who are ready to wire it into something else—and that something else had better include some memory. Even the 680 "complete" kit with memory, front panel and case for $345 is a complete computer only to those who are willing to enter and read out both data and machine language instructions in binary, do this each time the machine is turned on, and flip switches forever. This might provide interesting recreational computing for some people, but the machine by itself is certainly not very practical. Most Altair users buy at least a terminal, the Basic language (available as of Feb. '76 only on the Altair 8800) and the extra memory necessary to support Basic, and these additions put the system in the $2,000 range.

One might also question my statement that the HP-25 is a complete computer because of features which it lacks, or because of the general limits of the system. Hewlett-Packard, for example, calls the HP-25 only a "programmable calculator," while their HP-65 is "fully programmable" because it can store its programs on small magnetic cards. Such storage is certainly a useful feature, but it is not a necessary feature for programming to intellectually "feel like" computer programming. The HP-25 feels like a computer, and it doesn't need a card to do so.



(The new Texas Instruments SR-52 also has a card, and twice the memory and program capacity of the HP-65 for half the price. Readers contemplating purchase might well consider the SR-52; the HP-25 is the cheapest but not necessarily the best choice.)

The HP-25 is in fact quite limited for a computer: it has 49 steps of programming and 8 memory registers, plus 5 registers in its operational stack. Refusing to call it a computer because of its limits, however, is mainly a matter of preference for larger systems. I would call the HP-25 a computer despite its limits. The system it defines is not at all trivial. It can do enough so that it is fun to program, and it is also useful, and will do a reasonable fraction of the personal computing that a mathematically oriented user would do with a much larger system.

Indeed, the very fact that the system is limited has both practical and recreational advantages. A larger system can be very impractical when one is seduced into wasting days solving what had at first appeared to be a simple problem, writing hundreds of steps, debugging, and improving the output. If one has an HP-25 and a problem can't be solved on it, one knows that the problem will take a substantial amount of programming on a larger system, and this is a very good point at which to ask whether the solution is worth the trouble.

Another recreational and educational advantage of a limited system is that one is much more quickly forced to optimize programs. Optimal programs are much better esthetically than sloppy programs, and there is a real feeling of craftsmanship in knowing a system well, trying every trick in the book, and finally being able to shoehorn a complicated problem into a limited number of steps.

An example of a recreational problem may give some idea of the possibilities and limits of the HP-25. I have been trying since I bought the machine to make it play ticktack-toe. I am almost certain that this is impossible in such a limited machine, but it is less impossible than one might think at first.

For example, there are nine positions in a ticktacktoe board—how can one store the contents of these positions in only eight memory registers? Well, the machine can take a number with a decimal point and throw away either the fractional or the integral component. Thus it is possible to extract a specific digit in a larger number by placing the decimal point in front of the digit, taking the fractional component of the number, then multiplying this fractional component by ten and looking at the integral component. It is also possible to place a digit in any decimal position by multiplying the digit by ten-to-the-power of the desired decimal position and adding the resulting number to the number in a storage register. (Unfortunately the ten-to-the-power function is slightly off in the last decimal place for ten-to-the-seventh and higher powers, but if necessary this can be corrected by a few steps of programming.) Since one can both take out and also put a number in any decimal position, the ten decimal places in one register can be used as ten different memory locations. Thus if necessary a representation of the ticktacktoe board can be stored in only one register.

A more fundamental limitation for implementing ticktacktoe is that the small number of branches which are possible in 49 steps seems much lower than the number of decisions a computer must make in playing ticktacktoe. The number of branches possible for the HP-25 is much less than 49 because a branch requires from one to five steps depending on what one is doing. In order to branch it may be necessary to bring the two numbers to be compared into the two appropriate registers, specify the logical test to be applied, and specify the two different directions to go to depending on the outcome of the test, and this totals five steps.

My latest thought is to sequentially peel off digits from a sequence of memory registers and use these digits as a sort of higher-level programming language to specify the sequence of application of a group of subroutines, thus in a sense expanding the programming capacity. I am fairly sure that this will not work, since the peeling-off and branching program alone will take quite a few steps, and there might be room for only about four or five very short subroutines.

Whether or not it can be implemented, the ticktacktoe problem is an example of the complexity and richness of strategy possible in what seems at first to be a very limited system. And despite the difficulty with ticktacktoe, by no means are all games excluded. For example, the *Applications Programs* book that comes with the HP-25 includes a moon-landing program and a version of nim. And I have written an ESP testing game, a ball-bouncing game, and others. Games which can be implemented tend to be relatively simple, however. The best game is the system itself.

Hints on Buying a Used Teletype

by David Ahl

If you've been following the Teletype ads, you know that the ASR 33 has been sold for well over 10 years. It was, for many years, the workhorse of the TWX and Telex networks (although the real heavy-duty workhorses of the networks are the Models 28, 35, and 37).

The ASR-33 is Teletype Corporation's most popular product in history with over 600,000 delivered as of early 1976. Many of these are now on the surplus market available both "as is" and reconditioned. Here are some hints if you're considering buying a used unit.

Serial Numbers 1-200,000 are likely to be dogs. In "as is" condition they're worth $250 or less. They're generally tough to refurbish because so many parts must be replaced. (By the way, the Serial Number is hard to find—you have to take off the cover and look for a little plate, generally covered with dirt and oil, in the back corner).

Serial Numbers 200,000-450,000 may be OK as is, particularly if they've been under a regular maintenance contract. As is price should be $350-$400 or so.

Serial Numbers 450,000 and higher should be in good shape. Nevertheless, it's worth seeking out units that have been under a maintenance contract.

Reconditioning generally adds $250 or more to the price; generally a reconditioned unit of any age is going to cost $700-$900. While this is only $100-$300 under the new price ($969 plus), sometimes a reconditioned unit is actually better — all the initial bugs have been worked out and it should be in good adjustment.

Of course, another alternative is to buy a new one. Base price direct from Teletype Corp. is $969. However, for most minis and/or micro systems, you're going to have to add about $75-$200 worth of bits and pieces from the computer vendor to make it all work.

The other big question—where do you get one? Many, perhaps most, second-hand computer or terminal dealers will not sell you an ASR-33. The reason, as a salesman from American Used Computer told me is that dealers can make a lot more money renting them than selling outright. Hence, you're going to have to make a few phone calls to terminal vendors listed in the Yellow pages or from ads in Computerworld, Computer Hot Line, etc. Chances are you'll find someone with a temporary overstock and maybe even someone who cares about hobbyists or schools and will sell you a '33. After six phone calls in the metro New York area I found 3 dealers willing to sell me a '33 for prices between $775 used and cleaned to $875 reconditioned. I eventually decided to wait out the 8 to 12-week delivery cycle and get one for $969 through the MITS-Teletype Corp. deal. In theory, delivery today is better, but don't bank on it.

There are many models of the '33, but by far the most common model in use by schools and hobbyists is the ASR (Automatic Send Receive) i.e., it has paper tape reader and punch, friction feed (as opposed to sprocket or pin feed on the paper). The order number is: 33 ASR/TC or 3320/3JA.

A Creative Computing Equipment Profile . . .

HEWLETT PACKARD 9815A PROGRAMMABLE CALCULATOR

by David Ettel
6015 36th Ave. NE
Seattle, WA 98115

A person unfamiliar with or not interested in how to build his/her own computer system may be interested in a programmable calculator (a what?! — a programmable calculator — you can't hold it in your hand; it can be programmed; it has a few memory registers, a printer, and can be hooked up to many different peripheral devices). Recently I had the pleasure of evaluating (playing) with the brand new, just off the shelf, HP 9815A programmable calculator and immediately fell in love with it (some jealousy has arisen in certain quarters). It is, in the words of my boss, 'darling' — by which I understand her to mean — sleek (13 x 13 inches - 4 inches high) and compact (13 pounds). The 16 character/line thermal printer (numeric, alpha, and a few special characters) and cartridge system, both of which are built in, add to the character of the machine.

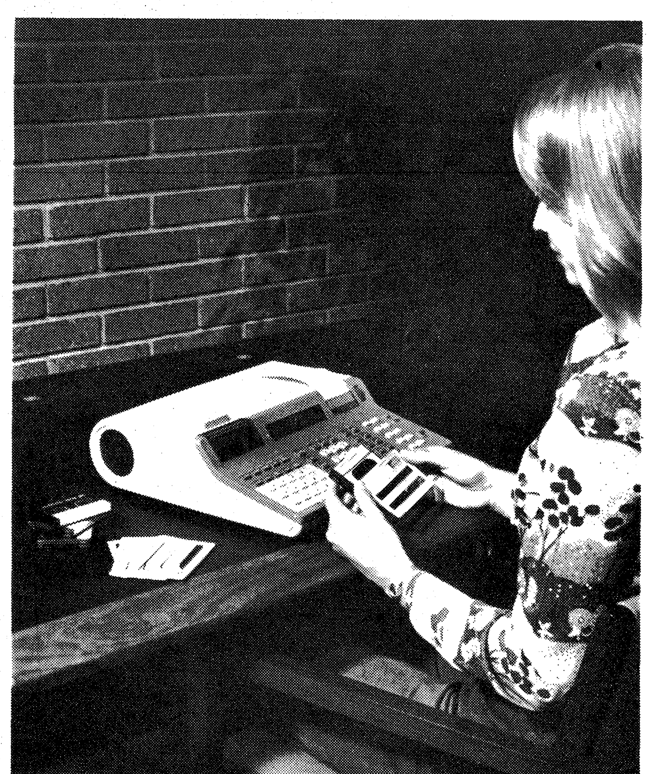
The 9815A uses a keyboard language (i.e. commands are entered by using the proper key on the keyboard) which is more effecient (fewer steps are required to do something) than any other programmable calculator I have seen. Some of the commands are:

- GO TO
- GO SUB / RETURN
- FOR NEXT loops
- STORE into a register
- RECALL from a register
- IF statements for conditional branching

24 scientific functions are also included along with the four basic arithmetic keys. Once entered,programs may be edited or stepped through. Editing includes inserting, deleting and changing of instructions. GO TO's and GO SUB's using absolute addresses are automatically updated when inserting or deleting.

The programs are stored in the program memory. The basic machine contains 472 program steps with an option to have that increased to 2008 steps. Storage registers are created by assigning 8 steps of program memory to be a main storage register. Thus memory can be divided between program memory and data register memory depending on the application.

472 steps or even 2008 steps may seem like a serious constraint, but this problem is taken care of by the unique cartridge system. Using cartridges which can hold 96,000 program steps or 12,000 data registers, the effective memory size of the 9815A is enlarged enormously. The cartridge system uses bi-directional search to locate a specified data set. Search speed is 60 inches/second. Read/write speed is 10 inches/second. The compact cartridge contains 140 feet of tape. Two tracks are used for recording information. With the cartridge system programs



can be segmented - one program can call another program from the tape and pass control to that program. Called programs can overlay the calling program. Also, data may be stored and recalled from the tape. Program size and data register memory size are therefore not severly limited.

Peripherals include a plotter, a digitizer, paper tape reader, thermal page printer, and paper tape punch. The plotter would be nice for games.

It seems to me that the programmable calculator may be the closest thing to a computer system (core, operating system, software, printer, off-line storage) for a reasonable price - reasonable for the HP 9815A being $2900. I'd be interested in knowing what a comparable do-it-yourself system would cost - then I could make an intelligent decision about what goes on my Christmas list next year.

(More information on the 9815A including brochure and technical data sheets are available from Inquiries Manager, Hewlett Packard, 1501 Page Mill Road, Palo Alto, CA 94304 or from your local HP office.

A Creative Computing Equipment Profile . . .

TEKTRONIX 4051 GRAPHICS SYSTEM

by Stephen B. Gray
Gray Engineering Consultants
Darien, CT

The Tektronix 4051 is one of several hard-wired BASIC computers (which can be programmed only in BASIC), but it is the only one that also offers graphics, as well as the ability to function as a terminal. I had the good fortune to be able to borrow a 4051 for a month, and I'd like to tell you about this fascinating desk-top (or pedestal-mounted) graphics computer. First, a look at the hardware, then later the software.

The Hardware

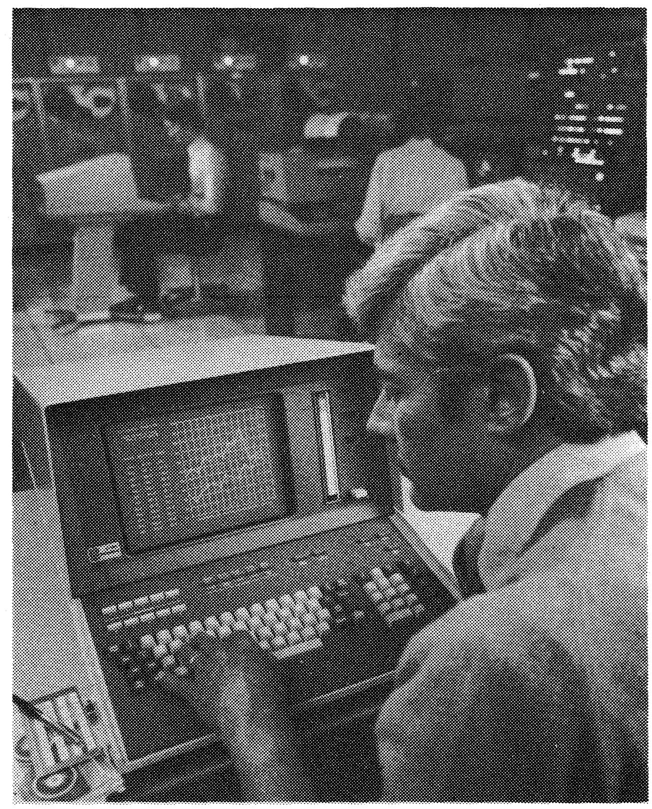
The main keyboard offers the full 128-character ASCII set with both upper and lower case; 96 printing characters and 32 control characters. Several additional built-in character fonts permit the use of accented vowels and special signs such as the British pound sign. To the right of the main keyboard is a numeric keypad which, when used with nearby keys for the four arithmetic functions plus parentheses and exponentiation, permits fast calculator-type computations, without line numbers or any programming at all. The keypad is also a help when entering numeric data into a BASIC program.

A group of ten user-definable keys, used in conjunction with the shift key, allows up to 20 pre-defined program subroutines to be individually selected for injection into the main program. The first 99 line numbers are reserved for these User Definable keys; each key transfers program control to the first of four line numbers, enough for a *very* short subroutine or, more likely, enough for a GOSUB to a larger subroutine. If the last of the four lines is not a RETURN (or END or STOP), the system keeps executing statements in sequence, in the user-definable routines, until it finds a RETURN, STOP or END, or until it continues into the main program beginning at line 100. These keys are very handy for menu selection, and a plastic overlay card you can write on keeps track of what you've got them programmed for.

At top center of the keyboard, five keys provide ten functions for program editing, permitting characters to be changed, deleted or added. A nearby Auto Number Key will provide a line number automatically for each BASIC statement before it is entered on the keyboard, starting with 100 and incrementing by 10; the initial line number and increment can be changed to whatever is desired. A Step Program key permits executing the current BASIC program one step at a time, starting at the first line or, by using a GOTO, at any line desired.

At top right are three keys for peripheral control. The Auto Load Key causes the internal magnetic-tape unit to load the first program found on the current tape cartridge. The Rewind key will rewind the tape cartridge back to the beginning, and Make Copy causes the optional Hard Copy Unit, if one is attached, to make a paper copy of the information on the display.

The vertical slot at the right of the display holds a 300-kilobyte magnetic-tape cartridge; more about that when we get into software.



The display uses an 11-inch-diagonal direct-view storage crt, with 72 characters per line, 35 lines per page, 1024 x 780 addressable graphic points. Clever feature: the data on the screen dims to a lower (but still readable) level after a few minutes, to conserve energy; pressing the shift key restores the original brightness.

The MPU used in the 4051 is a 6800; a 32k ROM holds the operating system. The 4051 is supplied with an 8-kilobyte RAM for work space; memory is expandable in 8k options up to 32k maximum.

Hardware options include the 4631 Hard Copy Unit for dry-process copies; 4924 Magnetic Tape Unit for 300k *external* bytes; 4662 Digital Plotter for X-Y plotting and digitizing on a 10-by-15-inch work area; 4952 Joystick for positioning the graphics cursor in interactive graphics.

Communications Option

A communications option, which adds an RS-232C interface, lets the 4051 act, in terminal mode, like a Tektronix 4012 Computer Display terminal, with keyboard inputs going direct to whatever computer the 4051 is connected to, and with returning data and graphics going up on the screen. The same option lets the 4051, in communications mode, use the internal tape unit to send or receive data at speeds up to 2400 baud. An output-only RS-232C interface, for line printers, is also available. Either option is in the form of a plug-in ROM package, about the size of two cigarette packs, which fits into a receptacle in the back of the 4051. There are three others, which I haven't seen: matrix, editor, and binary-loader ROM packs; more are in the works.

Software and Manuals

Four manuals and two pre-recorded tapes come with the 4051. The two manuals on BASIC will be covered in a future review. The tapes are duplicates; one is to be put "in a safe place; it exists to minimize down time in the event the operational tape is accidently erased." Each tape contains these programs: system verification, tutorial, Y-only data plot, X-Y data plot, histogram plot, and function plot. The "4051 Graphic System Operator's Manual" contains all the information that's on the tape, with amplification, plus an introduction, sections on "keys, buttons and switches" and routine maintenance, appendixes on error messages, specifications, and installation, a glossary, and an index. The fourth manual is the "4051 Graphic System Reference Manual."

System Software Tape

For the beginner, the first of the seven programs to use would be the Graphics System Tutorial, which has a menu with these six alternatives: the whole tutorial, keyboard operations, demonstration of graphic software, programming primer, graphics commands, index (for picking and choosing).

System verification, which is meant to be used when the system is first received, and "at any later time when system performance is in doubt," takes less than 10 minutes, and is mostly automatic. The Software Verification part requires the user to press keys as requested, and the system responds with various phrases and test patterns. The Firmwave Verification checks out system memory, and runs all by itself in less than a minute.

The two best features about all the canned plotting programs is that they're automated as much as possible, and there is a great variety of statements that let you do just about anything you can think of, plus some that might not have occurred to you. Let's look at a function plot to see how it all works. A function can be plotted in a very short time, if the user chooses to let the system set certain parameters automatically. If he/she wishes, the user can override these "default values" and, for instance, move either axis and/or change its length, change the data range on either axis, choose either a line or point plot, and choose from five point-plot symbols.

After the user has chosen Function Plot from the master menu, he/she is asked to enter the function, either single-variable in Y or double-variable in X and Y. The screen then shows the Function Plot menu, listing all the parameters one can enter. Most of these can be skipped by letting the 4051 set most of them with default values, but there are of course several that must be entered. This is done by selecting the first menu item, Display Function, which will then cause the 4051 to ask the user to enter only three numbers: beginning and ending independent variables, and increment.

After the increment is entered, and the user presses Return, the screen displays the function, complete with axes and labels. If the plot is satisfactory, that's it. But if the user wants to smooth out the curve, he can enter a smaller increment and replot in a few seconds. By calling up the Function Plot menu, he can make further changes as desired. If he forgets what changes he's made, selecting List Parameters from the menu will provide a reference chart.

As a BASIC computer, the 4051 is excellent. All the necessary statements are here, plus some I'd never heard of before which proved not only fascinating but useful. FUZZ decides just how close a comparison is to be made between two non-zero numbers; FUZZ 10, for instance, compares two numbers to ten digits. This gets around the problem of what are often necessarily imprecise mathematical operations. SECRET permits a program to be executed only; "it can never be listed, saved, or in general output from memory." SUM "returns the algebraic sum of the elements in a specified array."

Graphics

Now to the specialty of the 4051, graphics. With its amazing variety of statements, the 4051 can create just about anything you've got in mind, as far as static display goes. VIEWPORT and WINDOW determine which part of a curve will be shown where on the screen. AXIS produces an X-Y axis, with tic marks as desired. After you specify an initial point with MOVE and a pair of coordinates to indicate how far from the lower-left corner of the screen you want the point to be, a DRAW statement with coordinates will specify a line. A square can thus be drawn by using one MOVE and four DRAW statements. This can be simplified with arrays to a single DRAW statement by using DIM, four pairs of coordinates in DATA, READ X,Y and DRAW X,Y.

RMOVE and RDRAW "free the programmer from having to figure out the absolute coordinates of each data point" by interpreting the numeric constants as relative increments to the position of the cursor. ROTATE will move a pattern, or single vector, through a specified angle, in either direction, once or more than once (with FOR/NEXT), and is a main ingredient in creating one type of graphic art.

All this information on graphics is given in the Reference Manual, which also has sections on language elements (constants, variables, operators, strings, DIM, LET), environmental control (presets for degree/radian/grad, trace/normal, initializing the system, fuzzy comparisons, etc.), system control (CALL, COPY, HOME, PAGE), memory management (DELETE, MEMORY, SPACE), controlling program flow (END, FOR/NEXT, GOSUB/RETURN, GOTO, IF/THEN, RETURN, RUN, STOP), handling interrupts, input/output operations (DATA, OLD, PRINT, READ, SAVE, SECRET, etc.), math operations (ABS, COS, INT, LOG, PI, SQR, etc.), character strings (ASC, DIM, INPUT, LEN, VAL, etc.), programming editing, debugging and documentation (DELETE, LIST, REMARK, RENUMBER, SET), language syntax (rules for line numbers, keyboards, data items, etc.), and appendixes on error messages, tables (ASCII character values, character priority, fonts, etc.), interfacing information, and glossary.

The two manuals are very good, except that the examples in the reference-manual sections on the various statements are too skimpy in many places.

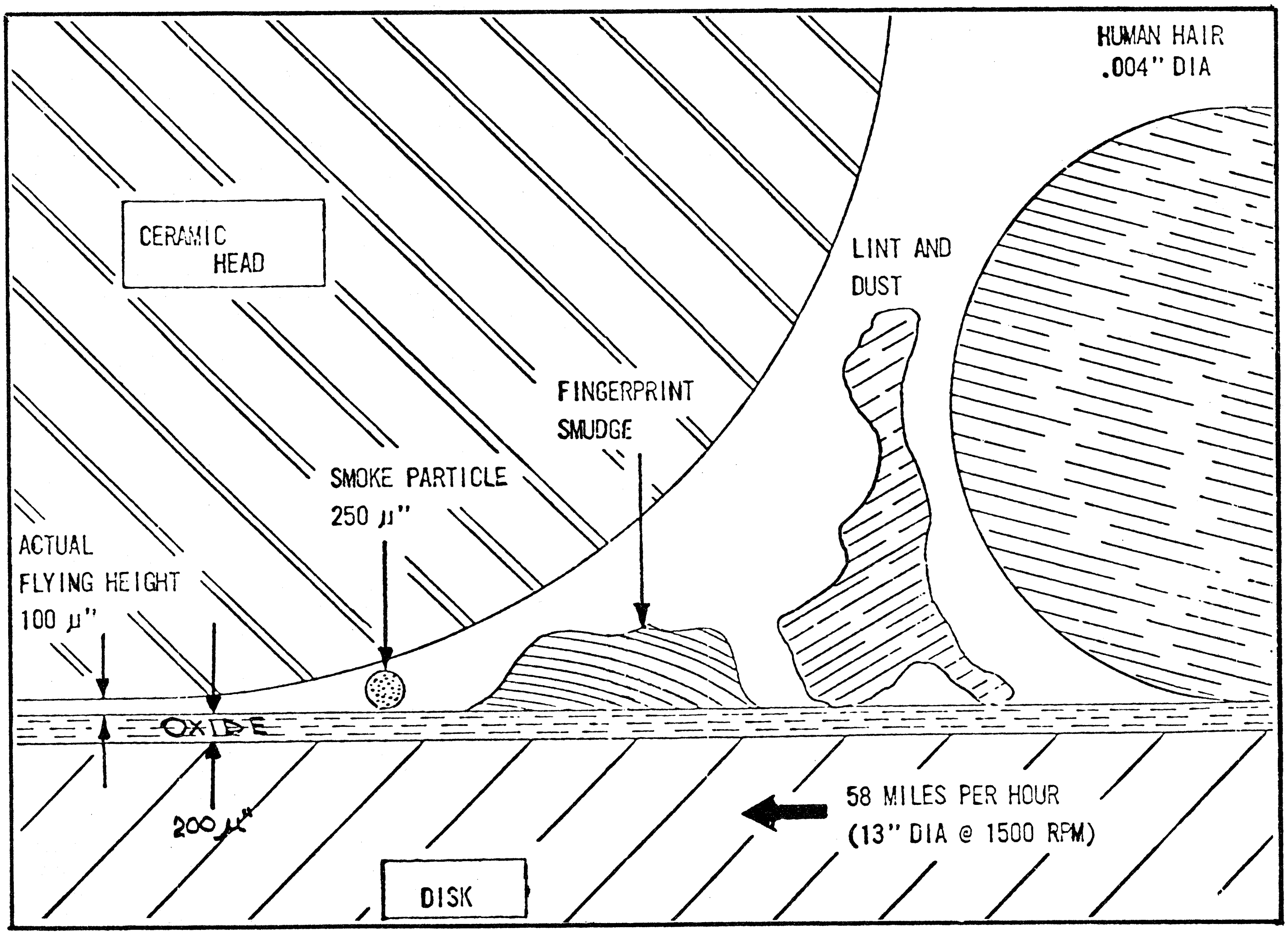
There's a lot more to say about the 4051, but if you're interested in this computer, a demonstration is of course much better than any number of words. Tektronix has several fine demo tapes, such as the one I had, which gives examples of applications in mechanical engineering (conic analysis), electrical engineering (filter design), business (financial analysis, depreciation, savings and loan), mathematics (integration), statistics (regression analysis), plus sections on peripherals, available character sets, stock-market bar chart, Crosshair Simulation (a sort of draw-it-yourself), three-dimensional plot, winding up with a Capability Demonstration, which describes the 4051, shows a wide variety of graphs, and ends by drawing the Tektronix logo.

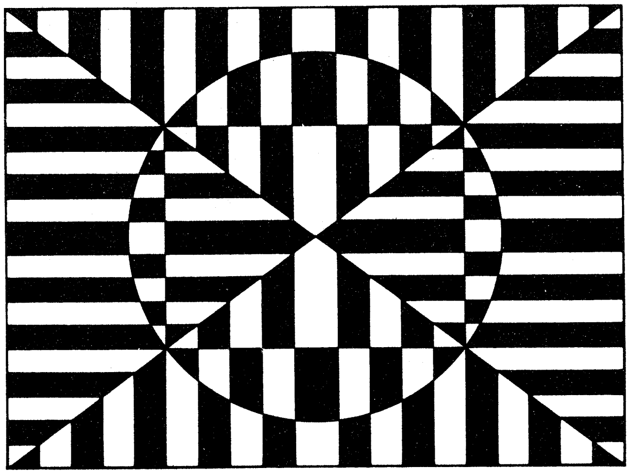
If there's anything Tektronix left out, I wasn't able to discover it in four weeks of using this marvelous machine, the 4051 Graphics System.

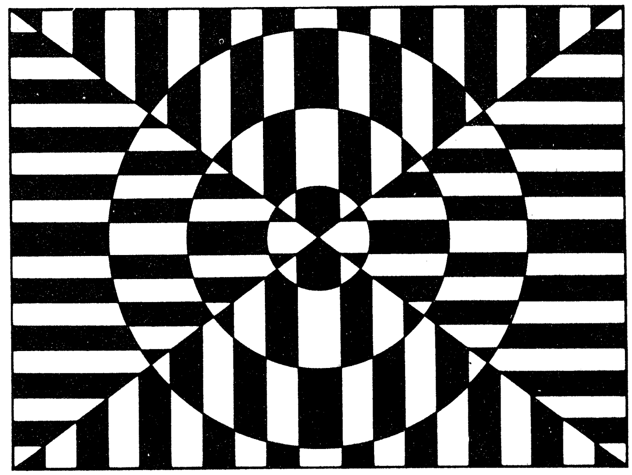
DISK DESTRUCTION MADE SIMPLE

by Bill Thorne, London, England

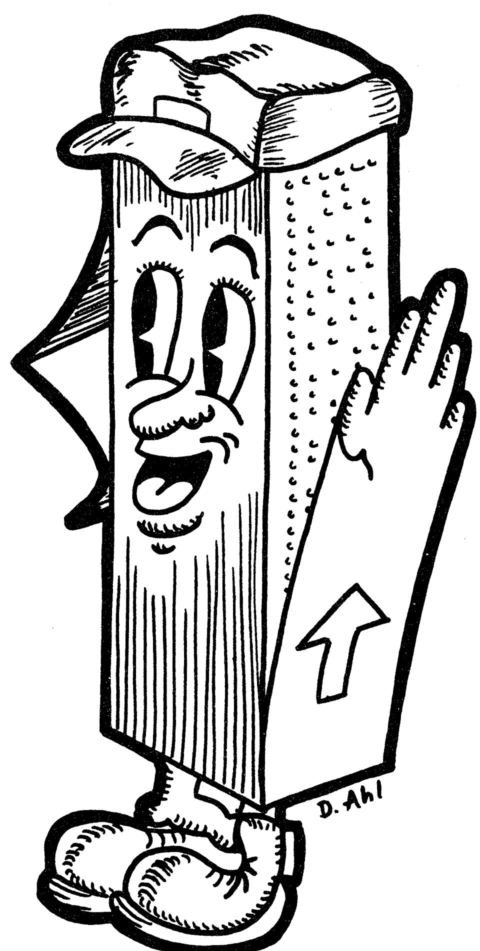
The diagram below vividly illustrates the effects that lint, dust, finger prints, smoke particles, etc., can have when they are on a disk surface and meet flying heads of a typical cartridge disk drive. The results, as you might imagine, can be very damaging to reliable disk performance. By storing disk cartridges more carefully and keeping them in a clean environment, reliability can be dramatically improved.







Reviews



Punchy Paper Tape

CREATIVE COMPUTING
Feature Review
34 Books on BASIC

Stephen Barrat Gray
Gray Engineering Consultants
260 Noroton Ave.
Darien, Conn. 06820

Installment Number 5

This group review of 34 books on BASIC started in *Creative Computing,* Vol. 1, No. 3. Five to seven reviews appear in each issue. More next issue.

For a future group review of books on applications of BASIC, I would appreciate information concerning such publications. This would include not only books such as Peckham's *Computers, BASIC and Physics,* but also applications books not oriented toward any particular language, but which could be used with BASIC, such as Gruenberger & Gaffrey's *Problems for Computer Solution.*

Also appreciated is information about books on BASIC in languages other than English.

22. *Fundamental Programming Concepts*, by Jonathan L. Gross and Walter S. Brainerd. Pub. Apr. 1, 1972, by Harper & Rowe, New York, N. Y., 304 pages, 6 x 9¼, $9.95 (hardcover).

Some excellent portions, but too many difficult areas unexplained, far too much extraneous or overly complex material. Rating: for classroom use, B+; for the solitary reader, C–

On the dust jacket, some of the letters of the title are filled in, generating a subtitle: Fun Programming. This book could be quite a lot of fun, in the hands of a good teacher who could explain the hard parts, and cut through the material that doesn't belong. But in the hands of the solitary reader, this text provides more exasperation than fun.

The preface says "This book is the appropriate length for a concentrated semester course and could also be used as a self-teaching guide for a motivated person who has a time-sharing service available." Highly motivated, that is, and with an IQ above 150.

There are ten chapters: Elementary BASIC; Advanced BASIC; Information Processing; Character Strings; Mathematical Methods (maxima and minima, approximating roots, area, graph plotting, matrix operations); Simulation; Two Puzzles (algebraic, colored cubes); Artificial Intelligence; Machine Language, Assemblers, and Compilers; Programming Languages.

The beginning is good, a five-line program that adds two constants, using three LET statements, on page 2. The next page has a five-liner that finds the product of three constants, using READ, DATA and LET.

Page 6 begins a section on using a terminal, with 11½ pages on remote access, and on writing and running programs. The writing is simple and straightforward, with a rare sense of humor: "The computer does not have extrasensory perception, so it cannot detect errors of meaning, like adding the wrong number or naming the wrong variable." Or later, "On some systems the more euphemistic word UNSAVE is used instead of KILL."

But by page 20 the flaws begin to show. There are not enough examples; none at all in some cases, as in the paragraphs on relation symbols. Flowcharts are introduced early, but there are only 13 in the whole book.

The first major flaw is on page 25. Line 130 in the PRIME program is not explained; it is not readily apparent why I is compared with N-1 rather than just N.

If this were the only unexplained program line, then little harm would have been done. But there are many such puzzlers that only a professional programmer could easily understand, and which leave the beginner confused and exasperated. On page 73, line 340 of the BOOK program reads PRINT B;"   "; M;D without an explanation that quotes around blank spaces are one way of formatting output. On page 58, BKWRDS prints 20 numbers both forwards and backwards; there is no explanation of a key line, 190 FOR I = 1 to 10, although there are 20 items in the list, not 10. This is the introduction to subscripted variables, which need not be so complex, with 23 lines, and with one line that raises more questions than the program answers. Lines 200 to 220 are also not explained, other than in the REM line, EXCHANGE ELEMENTS. In general, there is not enough explaining of programs, usually only a general description, no explanation of the lines that are different from what has been previously encountered, except in a few very rare instances.

There is a group of problems after most of the sections in each chapter, although without answers. Some of these problems are for high-IQ types only, such as the ninth one of page 45, which asks the reader to "write a program that allows a person to play the game of matchsticks against the computer."

In many places, the authors get into areas that are off the track. In the example given for GOSUB and RETURN, the book goes off into an explanation of a return-address stack. Although some readers may want to know what's inside the machine, it is extraneous to the subject at hand, as is the section on "subdivisions of the scratch area."

By page 54, the reader has become conscious of a writing style that is dry and often stilted: "It is emphasized that the computer creates the data stack during compilation, before it begins running the program."

There are too many tricks that may only confuse the uninitiated, such as using the line READ X,Y,X,Y,X in explaining RESTORE. This is fine for the high-IQ readers curious about the byways of BASIC, but not the average reader.

A 37-line program that computes the charges for long-distance telephone calls is used to demonstrate double-subscripted variables. Not too difficult, but why use such a long program every time?

By page 69, the authors are through describing BASIC, and go into applications; the problems get much harder in this part of the book. Problem 2 on page 76 asks the reader to use data such as 10010, which must somehow be broken down into the individual binary digits, but there is nothing up to this point (or anywhere else in the text) to indicate how to do this.

Many other things are left for the reader to figure out. The only explanation of how bubble-sorting works is one brief sentence and one REM line.

The authors are stingy with examples. Built-in functions, such as ABS and SGN, are introduced without any examples of why, how and when to use them. There is only one example of a table. There are no examples of PRINT with and without comma and semicolon.

Most of the application programs are much too long and complex to be useful to the beginner who has learned what BASIC he knows from this book. There is little point in using BOOKS, a long library-book-availability program that requires such a long explanation, and which does little if anything to teach BASIC. There are seven pages on cryptography, which must be a favorite subject of one or both authors; this is usually and end-of-book subject, but here it is used as an application of strings, for other than which it is rather trivial. Page 173 begins four pages of explanation for an archeology program. Only one line is explained in a 99-line program on baseball simulation.

The most curious program is in Appendix C, which contains "a complete description and a listing of the BRAIN simulator," a "mythical computer" introduced to help in "developing a little programming skill in machine language." This is quite interesting, but certainly does not belong in a beginning text. It requires 24 pages in the main text, and all ten pages of Appendix C, including a three-page flowchart and a four-page listing of the 285-line program.

The title of the book should be more on the order of "Beginning BASIC with Applications." However, this is a good book for advanced work, or for browsing in the applications portion.

There are some unique portions. These are the only authors to distinguish between the use of parentheses and brackets in BASIC, noting that "arithmetic expressions use parentheses to indicate the order of evaluation. Function arguments are enclosed in parentheses. Subscripts and dimensions are enclosed in brackets. However, many versions of BASIC allow the programmer to use parentheses and brackets interchangeably."

The authors are the only ones to say, "Usually, the main concern with a BASIC program is getting it to run successfully, not necessarily most efficiently. Therefore, a higher premium is placed on clarity than upon brevity." They are alone in showing how to generate random numbers "if there is no random-number generator in the language or if the one available is not satisfactory," and in giving "suggested reading" throughout the text, after many of the groups of exercises. They are among the very few to get into the economics of programming: "The cost of computing is going down and the cost of programming is going up, which means that there are fewer and fewer occasions which justify the use of assembly language programming." They are among the very few who discuss the infinite loop. There are five fine pages on debugging, going into hand simulation, tracing with PRINT statements, and sample runs.

The last problem in the book is a short one, "Write a program that will produce rock and roll music." How can a beginner do this after reading only one page on "a program that composes music"? This is very reminiscent of the last section in Kemeny & Kurtz (2), "Harmony in Music," which has over three pages on a more difficult area of music, with a long program and also a long and careful explanation of it. The last problem in the Kemeny & Kurtz book is much more reasonable: "Try the program MUSIC on several melodies and suggest improvements in the program based on your experience."

Chapter 10, on Programming Languages, has a unique and interesting feature: an 8-line BASIC program is presented, followed by discussions of its equivalent in FORTRAN, ALGOL, and PL/1. That's fine, but why follow it with eight pages on SNOBOL?

In trying to be all things to all readers, the authors end in exasperating most of them.

* * *

23. *Programming Time-Shared Computers in BASIC*, by Eugene H. Barnett. Pub. Apr. 14, 1972, by Wiley-Interscience div. of John Wiley & Sons, New York, N. Y., 366 pages, 6 x 9, $12.00 (hardcover).

Packs much information into a well-designed book, with many unique features. Rating: A

One of the best organized of these books, this is meant to be used (according to the dust jacket) in high schools, colleges, in-house courses, and for self-study. It should do admirably for any of these.

A short program is presented on page 4, after a discussion and a flowchart.

The subsequent presentation of programs proceeds slowly and surely, and although the program on page 40 is 61 lines long, it is a very simple program that is easily understood. This payroll program, which includes employees Mike and Deborah Barnett (nepotism or child labor?), is modified and enlarged upon up to halfway through the book, by which time it has taken on a very new look.

There are eight chapters: Introduction; Elementary BASIC Programming; Programming Hints; Matrices, Input, Random Numbers, and Other Good Features; Advanced Examples; Advanced BASIC Features; Data Files: Input/Output; Computer Features. The book ends with solutions to selected exercises.

There are several sets of exercises in each chapter, presented right after each major statement or group of statements. These exercises are given in a nice variety: at the end of the section on GO TO, for instance, there are exercises on interest, test-score averages, center of gravity, vectors, calculating area under a curve, and standard deviation, among others, and this by only page 60. Some of the exercises are quite far-out, such as the one that asks the reader to "computer the (x,y,z) position of a satellite," given half a dozen parameters.

By page 73, after IF-THEN, the author figures it's time for the reader to get a better idea of how problems are solved with the computer, so there are 18 pages of "more sample problems," with programs and full discussions on subjects such as loan interest, largest number, depreciation, moving average, linear regression, quadratic equations, and binomial probability.

Over 21 pages are devoted to A Bag of Tricks, which are "little gimmicks" of the type the reader is urged to "consciously develop." The author discusses first the accumulator, which he believes to be "the most important single trick in computer programming," then goes into counters, interactive techniques, recognizing the end of data, step functions, and several others. One of the best is that of using employee numbers as line numbers, and then letting the computer sort them into numerical order.

This is the only book other than Sass (21) to use MAT arithmetic for business applications, such as accumulating total sales for each retailer, salaries, production and distribution methods, product test data, etc.

Random numbers is an area to which many authors devote a minimum of space; Barnett has eight very nice pages.

By page 231 the author decides it's time for advanced examples, and presents 24 pages of problems, programs and discussion, covering, among others, moving averages, sorting, survey analysis, simultaneous equations, Monte Carlo simulation of a business decision, random walk, and even simultaneous differential equations, a nice batch of sophisticated applications.

The last chapter contains one of the few sections on editing in these books, and the only one to discuss time-sharing contracts.

A unique feature is a preview, right up front on page xiii, telling the reader what he can look forward to, with a list of Program Subjects, "The following major subjects are illustrated by various examples and exercises throughout this text." There are 54 subjects listed, under the headings of business and financial, sociology, engineering, mathematics and operations research, and miscellaneous. A very nice pre-sell to show the reader what big, exciting things are ahead of him.

There are very few things to say against this book, and all are relatively trivial. A few statements are somewhat confusing, such as "An important aspect of the READ statement is that the computer has the ability to never read the same DATA entry twice (except under special conditions . . . .)." True, but the wording is nevertheless confusing. On the same page, ". . . the READ statement can read as many variables simultaneously as desired." Simultaneously?

Barnett is the most realistic of all these authors. In a forward to the first set of "more sample problems," he writes: "It is the student's immediate task to: simply understand how each program does, in fact, implement each problem; begin to understand some of the trickery, chicanery, and subterfuge used to accomplish real-life tasks . . . ."

* * *

24. *Introducing BASIC,* by Theodore R. Blakeslee, II. Pub. Aug. 1972 by Educomp Corp., 298 Park Road, West Hartford, Conn. 06119, 162 pages, 8½ x 11, $4.95 (paperback).

Although there are many good parts, the coverage is uneven and the writing often stilted. Rating: C

Although this book is not published by a book company, it was, according to the preface, "written as a general primer for BASIC, and may be used easily with any computer system supporting the BASIC language, [although] the book was designed around the Edusystem line of computers marketed by Educomp." At the time this book was written, both Educomp and Digital Equipment Corp. were using "Edusystem" for their computer systems designed for schools. Then DEC copyrighted the name EduSystem, so Educomp now uses EDUCOMPUTER.

There are eleven chapters: Introduction, Fundamentals, Communicating (INPUT, PRINT, TAB), Assignment Statements, Control Statements, Arrays, Functions, Subroutines, Teletype Graphics, Errors, and "Pity the Poor User." There are exercises at the ends of chapters 3 through 10, without answers.

The preface notes that "Emphasis is placed on two important aspects, the random number generator and terminal graphics." Actually, the emphasis is on Teletype graphics (a 20-page chapter), not on the random-number generator (three pages in the chapter on functions).

The cost of the book has been kept down by typing the text, leaving the right margin unjustified, and doing all the artwork freehand (lines around boxes, exponential arrows, etc.). So the cost has been kept below $5.00, but the appearance suffers. This is the only book without consecutive page numbering; each chapter starts with page number one, such as 7-1.

This book is not very well written; it tries to be conversational, but still has too much of the pedagogue, so the beginner won't find the going very easy in a number of places. On the second page: "The computer language called BASIC is, itself, a computer program. It is a very complex program written in a different kind of computer language ("machine" language), one that requires great detail when writing a program." This on page 1-2 of a book that is designed to "let you obtain a working knowledge" of BASIC "with a minimum of teacher assistance."

Page 1-3 notes that every statement starts with a line number, but gives no examples. Nor are there examples of commands, which "use no line numbers." When writing about string constants, this is the only author who refers to elsewhere in the book for examples. There just aren't enough examples, and in some places, none at all.

Many sentences leave unanswered questions and therefore tend to be cryptic, such as "You should also note that the characters "12345" in a string (remember the quotation marks) are not the same as the numeric constant 12345 (twelve thousand three hundred forty-five)." And on page 7-5, "In addition, much of this mathematics also gives the constant e raised to a power as the result of manipulations (mystical, magical, mathematical manipulations)."

That alliterative phrase is a part of a curious and often awkward writing style that includes, "A discussion of plotting (nothing undercover) is postponed until Chapter 9." In writing about quotation marks in a PRINT statement, "No data is processed, just character-by-character regurgitation." A confession: "The author has used both kinds of LET statements and prefers the implied LET but must confess to a highly developed sense of laziness." Pedantic: "The LET statement resembles the algebraic equation but its operation is one of evaluation and assignment." Confusing: "Try using just a leading comma and no space" (without explaining what a leading comma is). Cute (heading for a printout): "RANDOM NUMBERS (SPECIAL SALE)." Or, on page 11-3, ". . . for use by people not at all familiar with the mysterious art of programming."

Some of the explanations are not too clear: "The comma will control the position of the type element in 14-position line segments." The author is the only one to use technical terms such as *embed* and *hierarchial.*

This is an uneven book, too generous in some places (nine pages on precedence), much too skimpy in others (only one flowchart in the book, and that a "verbal" one).

Program 5.06 as shown will not produce the given output, because the STEP of 1/2 is missing from the FOR statement. And if this is supposed to be an enlargement of the previous program, the step should be I/2, not 1/2. Had the author included the needed explanation, he would probably have realized that he left out the step.

There is no mention of the standard table dimension; page 6-3 notes that a program using the given list would have to include DIM A(6). Program 6.01 has a DIM A(9).

The explanation of program 6.02 says "The first 10 numbers in line 360 specify how many numbers are contained in each group." Not true; they are one less than the required value. Later the text notes that line 340 uses B(I)+1 and asks the reader why, but never tells him. This is an important but unexplained point.

The section on bubble-sorting, on the other hand, is long and very well explained.

The author often gets too far ahead of the reader, with complex programs to demonstrate something that could be done much better with a short program. The author then fails to answer all the questions the complex program brings up by virtue of its complexity.

The chapter on Teletype graphics has sections on "more on the TAB function," a simple graph, plotting with negative values, using Teletype graphics to solve an equation, and plotting multiple functions, this last with a program of 124 lines, five subroutines and 29 GOSUB statements to plot three functions.

Chapter 10, on program errors, covers debugging, with a seven-page example of debugging a program. The last chapter, "Pity the Poor User," covers the documenting of programs with flowcharts, generalization of programs, and output format. The opening states: "This chapter will be regarded by some as a sermon and an unnecessary sermon at that. However, having examined many student programs . . . ."

The writing is often on the Tom Swift level: "With fingers crossed and a feeling of dread we type RUN – AND THERE IT IS!!" Or this one, "A dummy variable, Z, is introduced, the changes entered, and one more hope rides high."

Appendix A is on Teletype terminals; Appendix B lists the BASIC statements, edit and control commands, and functions, and which of the five EDUCOMPUTERS uses each.

There is much emphasis on neat, simple and understandable program output.

* * *

25. *Computing With the BASIC Language,* by Fred Gruenberger. Pub. Aug. 1972 by Canfield Press, San Francisco, Calif., 140 pages, 7 x 10, $5.95 (paperback).

Good for reading after one or two others on BASIC, or after having some programming experience. Otherwise, too much extraneous (although interesting) material. Rating: as a first book; C–; as a second or third book, B

A disappointing book, when compared with others by Gruenberger. Although there are many interesting sidelights, they don't belong in a 140-page book "suitable for a one-semester introductory course in computing."

On the plus side, these sidelights are of most interest to someone reading this after having gone through one or more other books on BASIC, or having had some programming experience. The author is the only one to tell how to find out if the program is in an endless loop. He explains in more detail than most why the conversion from decimal to binary is not always exact. He gives the title and author of the first article on flowcharting. He is the only author to show what should *not* be done in flowcharting.

Gruenberger is the only one to show how to make a timing run. He alone asks the reader to run a program that will demonstrate the range of some of the built-in functions. He is alone in asking "What should we compute?" – and takes 3½ pages to answer. There is a fine section on "The Limitations of BASIC." He has one of the few write-ups on machine, assembly and high-level language. The first appendix gives three excellent pages on the internal workings of a BASIC time-sharing system.

The main text is set entirely in a typewriter font, with headings, programs and captions in five other kinds of type, resulting in an appearance that is less than handsome. There are exercises at the end of each chapter; the last exercise is number 310, but there are not 310 exercises; they are numbered 5, 10, 15, 20 . . . on the basis that this provides "for the insertion of new exercises"; there are actually 62 exercises.

The exercises are not set off from the text; on page 26, for example, the exercise runs into the text, and so this portion of the text, for one, could easily be skipped over if the reader is not careful. Actually, the reader may not always be able to tell what is text, or part of an exercise. If the latter, then the exercise contains important information that should be in the text, so as not to be missed by the hurried reader who skips the exercises.

There are eleven chapters: A First Look at BASIC; Data, Numbers and Variables; Flowcharting, More About Writing Programs; Looping; Functions; Subroutines; Problem Solving; Debugging, Testing and Good Computing; Extended BASIC Features, and A Final Word (The Elements of Computing, Your Future in Computing). There are six appendixes: About BASIC Itself, Things to Check For, Error Messages, Library Programs, Eight Advanced Problems (perpetual calendar, elapsed days, change maker, etc.), and a Commentary on Selected Exercises, with answers to only a couple of them.

There are only 97 pages of main text, so the information has to be poured on, with no space for investigating in detail, yet the author keeps straying from the subject at hand, going into areas that should be beyond the scope of such a text.

On a sink-or-swim basis, the author starts right off with an 11-line program on page 2, followed by a paragraph of explanation for each program line. This program would be difficult for many readers to understand.

Gruenberger is another author who emphasizes the nicety of not allowing a program to end in OUT OF DATA ON LINE XXX, saying this is "not the most graceful way to terminate a program," adding "but it works and is commonly used."

In several places he is cryptic: ". . . powers of ten and powers of two are incompatible," without further explanation. Some programs contain lines that are not explained at all, but whose importance is far from obvious.

The text begins to get a little murky on page 20, at which point the author gets into probability to illustrate flowcharting, with an example that is neither interesting nor simple.

As an example of the terseness of this text, the description of formatting with comma and semicolon takes only four sentences and two program lines.

The author writes as though he had first made an outline, and then filled it in as sparsely as possible. The style is rather dry, with only an odd item now and then to brighten up the desert.

This text seems to be another example of an expert writing for beginners without realizing that he's writing over their heads, that they may know absolutely nothing about programming.

Gruenberger doesn't seem comfortable in BASIC. For one thing, on page 36, he starts using the DO concept in describing a loop.

The chapter on Problem Solving with BASIC begins with bracketing, which is too much for many beginners, and is poorly explained, as are the three pages on Euclid's algorithm, which are very confusing; only a very clever reader could follow the text here.

* * *

26. *Business Programming With BASIC,* by George Diehr. Pub. Oct. 20, 1972, by Becker and Hayes, subsidiary of John Wiley & Sons, New York, N. Y., 344 pages, 8½ x 11, $8.95 (paperback).

A few good features, but mainly a jumbled, distracting mixture of solid masses of text and too many interrupting problems. Rating: C–

The basic idea of this text sounds good: it has a "semi-programmed" format, with questions throughout each chapter, usually related directly to the text. They are multiple-choice and fill-in questions, with the answers at the right side; the author suggests covering the answers with "an opaque card while you read the question."

But the questions come so thick and fast, interspersed in the text, that it's often difficult to figure out just which is which. The questions have a line before and after them, but there are often so many on a page that the separation isn't all that obvious.

Chapter 4, for example, is 23 pages long, followed by a five-page review quiz, and then four pages of quiz answers. Of the 23 pages, only 13 are text; the rest is taken up with 43 questions and answers. The entire book contains 330 questions in text, and 66 review-quiz questions. Some of the latter provide space for writing in the answer, so some pages have only a couple of lines on them.

Each chapter begins with a half-page outlining the chapter's contents. There are eight chapters: Computers and Terminal Systems; Introduction to BASIC; Conditional Transfers, PRINT Statement, Iteration; FOR-NEXT, Library Functions; String Manipulation, Input/Output Extensions; Subscripted Variables; Advanced Uses of Lists and Tables; Subroutines, Simulation. The three appendixes are on: extensions to BASIC, from file processing to PRINT USING; a summary of BASIC; and a table of system-dependent features and their parameters for nine time-sharing systems, from the B5500 to the XDS 5,7.

The first chapter is rather messy, jumbling together a too-long program, a description of terminal use, examples on two systems, error correction, and too many questions and answers, plus two pages on commands and editing features, for both CALL/360 and GE MARK II.

The writing is pedestrian, and sometimes confusing, such as "A character is one of the letters from our alphabet, a single digit, or one of several symbols such as * + = . . . Roughly speaking, when a number is stored in memory, two digits can be stored in the space required for a single character."

The text is typewritten, in a type so small that it isn't easy to read, especially since there are some very solid masses of type, seven inches wide and several inches deep. This is perhaps the worst feature of the book: the solid chunks of small type.

The author gets off to a bad start on the very first question, on page 3, asking who developed BASIC. The choices are: IBM, Kemeny and Kurtz, or "the creators of Screaming Yellow Zonkers." The rest of the 396 questions are all serious, but this first one sets an unfortunate tone.

There are 19 removable pages at the end of the book, to be taken out "so you can refer to one page while answering on another page." Since a duplicate of each of these is included in the relevant chapter, the redundancy seems unnecessary.

The first chapter presents READ and PRINT, but then, before making use of either, goes into a page and a half about a problem involving discount, interrupted by arithmetic questions intended to test the reader's knowledge of basic mathematics.

The first program is nine lines long, on discount off gross cost, with a fairly good explanation of the various steps involved, but rather long for a first program. On this same page the author goes right away into compilation, with a long paragraph on source programs, compilers, and object programs.

Four pages are taken up to show the discount program (now called a billing program) as run, corrected, saved and unsaved on *both* the IBM and GE systems, which seems wasteful.

There is some confusion as to terminology: on page 43, "After each line-number there is always an 'English' word. This word, called a 'Command,' describes the function or purpose of the statement." This author is the only one to use the terms "destructive read-in" and "non-destructive read-out" in describing the use of READ, PRINT and LET, and is the only one to refer to dummy data as a "trailer" value, to indicate end of DATA.

The text constantly refers to "figures" that explain various points such as constants, DATA, variables, etc. But these "figures" are just plain ordinary text with a line above and below, not drawings or charts at all, which is somewhat misleading. The style is rather dull, such as in this heading for the READ "figure": "Syntax and semantics of the READ statement."

The book's title is also misleading: the main text has very little to do with business programming. There are about two dozen complete programs in the book. Half a dozen are billing programs; the others are on assigning letter grades for a school, compound interest, optimal order quantity by search, finding values less than the mean, finding values less than the average, frequency tabulation, federal income tax, the "newsboy problem," etc. There is also a program that simulates a game of dice, and one that writes a song ("Old Macdonald's Farm") to illustrate the use of strings. Not much in the way of business programming in these programs, but there is in the two or three "programming problems" that accompany each chapter, which somehow seems a rather odd way of teaching business programming. As the preface puts it, "Each chapter concludes with a set of suggested programming assignments of varying difficulty. These assignments are designed to put the student's learning to the acid test. The programming assignments include data and correct answers for the problem." These programming problems are on computing unit price of supermarket items, monetary conversion, future values of investment, assigning letter grades, payroll program, depreciation, true annual interest, mean and standard deviation, rate of return on an investment, total receipts for a week, sales report, bubble sort, optimal replacement policy, maximum and minimum GPA, production costs, simulation program for inventory problem.

The sixth program is 44 lines long, but it's not too hard; it assigns letter grades depending on the student's numerical scores, and is mainly IF and DATA statements. The 17th program is quite complicated: 50 lines on an extended grading problem. Yet it does little other than determine the student's grade, and then list the various grades according to grade groupings.

This is one of the five BASIC books without an index, making it difficult or impossible to look up anything, except by spending much time searching the contents pages.

The jumble of text and questions is very hard to follow without getting lost or bored to death, or both. The book is needlessly confusing, and should have been heavily edited, set in book type, the padding removed, and the dull text improved. But would it be worth the trouble?

* * *

27. *Entering BASIC,* by John R. Sack and Judith L. Meadows. Pub. Dec. 1972 by Science Research Associates, Chicago, Ill., 133 pages, 7½ x 9¼, $4.95 (paperback).

One of the best, if the reader is very knowledgeable. But not enough detail for a beginner. Rating: for a programmer, A; for a beginner, C

A mixed bag, with some fine portions, and a preface that reads beautifully: "... BASIC is an ideal first step towards more complicated programming languages such as FORTRAN, COBOL, and ALGOL. Like the Roman god Janus who faces both ways, BASIC faces the needs of those just entering the world of automated data processing as well as those departing for its more stratified plateaus."

This book complements C. W. Gear's *Introduction to Computer Science* (also from SRA, 1973); "His book is frequently referred to for detailed discussions of specific points that go beyond the scope of this book. The two books are, however, mutually independent."

There are twelve chapters: Fundamentals of Computing, Interactive Processing, Introduction to BASIC, Assigning Values to Variables, Input/Output, Program Control, BASIC Functions, Subroutines, Array Handling, String Manipulation, Advanced Features, Debugging Aids. There are two appendixes: a Summary of BASIC Features, and Solutions to Selected Exercises. A 31-item glossary ends the book.

The book opens with a very readable style and a nice flow to the language. There is an excellent section, possibly the best, on Rules of Preference for arithmetic operators. The authors are among the few (9 of the 34 books) to introduce string constants and variables along with numeric constants and variables, rather than much later in the text, or not at all.

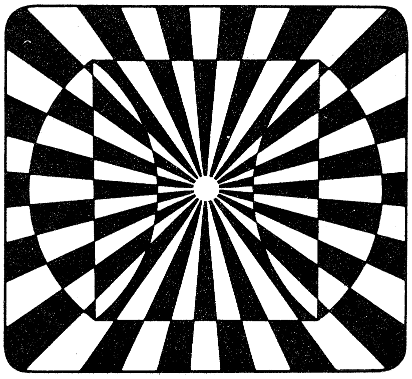
However, early in the book things begin to go a little too fast for comprehension, and without explaining terms. For example, "assembly language" is never explained adequately, in a section that would be difficult for most beginners, although not for a programmer. And yet the preface indicates that the book is for "the newcomer to computer science." The authors jump into executive commands without suitable preparation or explanation, without a word on how to enter these commands, or on what.

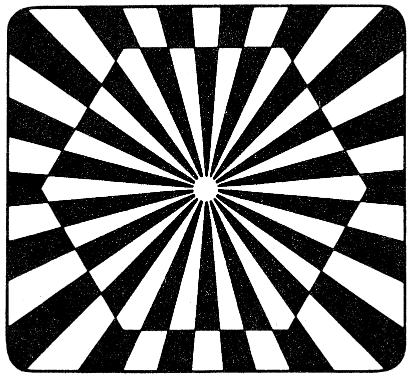
The first exercise is rather silly, on page 19: "To apply your newly learned wisdom, the next time you hear people talking computer jargon, casually drop the following: 'The other day I was communicating via teleprocessing lines with our time-sharing system in ————. I logged on with the system executive, ————, entered my program, and compiled it using the BASIC compiler . . . .'" Or are they pulling the reader's leg?

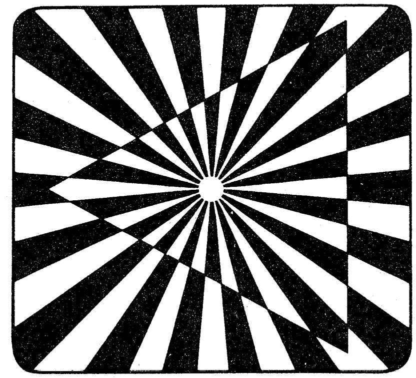
The first program, on page 21, is seven lines long, and prints the square roots of 1 through 5. The explanation of those seven lines is mixed in with a four-page general discussion of the elements of BASIC, and gets lost in the shuffle.

The program on page 39, on average rainfall, is too complex for many beginners to read; too much too soon.

Because of its lack of detail, and inadequate preparation for the programs presented, this book is not suitable for beginners, unless a teacher is available for filling in the gaps and answering questions. On the other hand, this is one of the best books for the knowledgeable reader, such as a programmer looking for a quick introduction to BASIC.







This is the 6th and final installment of this comprehensive review of 34 Books on BASIC. Several books have been published since this review was written; they will be reviewed in an Epilog in a future issue. Also, if there is sufficient demand, we will reprint the entire Group Review as a separate booklet.

28. *My Computer Likes Me* (*when i speak in BASIC).* Pub. 1972 by Dymax, P. O. Box 310, Menlo Park, Calif. 94025, 64 pages, 8¼ x 10¼, $1.49 (paperback).

The most elementary of all these texts. Based on a single idea (population), with too many programs using the computer just for printing. Rating: C

The text of this pulp-paper booklet is identical with (although printed more neatly than) the 1971 Dymax booklet *My Computer Understands Me.* But someone else already had the rights to that title, so it was changed. The text was written by Robert Albrecht, one of the principals of Dymax, and author of several other BASIC books (14,32).

The first thing one notices about the text is the variety of type and the ornaments used. Most of the text is Teletype print; the rest is in roman serif type. There are subheadings and notes in sans-serif, and major headings in large boldface fonts, including Old English. The ornaments run from decorative borders to pointing hands and large asterisks. Featured in several places is the almost-forgotten Kilroy symbol, with head peering over wall.

The first program is a two-liner based on PRINT "MY HUMAN UNDERSTANDS ME", with two pages of explanation. The second program is based on printing 7 + 5, with and without quote marks around the numbers, with a page of explanation. The text explains that when 7 + 5 is enclosed in quotation marks, it is a string; this is the only definition of string. The next page uses PRINT to perform the arithmetical functions.

The fourth chapter is on "mistrakes" and shows how to correct them. Four pages on Shorthand are about scientific notation. The next section, on Boxes, is about storing numbers.

The first sentences of the book, on page 1, say "This book is about people, computers and a programming language called BASIC. We will communicate with a computer, in the BASIC language, about population problems." The second mention of population is on page 9, under Shorthand, showing how to print the population of the U. S. The third time is on page 15, under Division of Labor, which is about computing population in N years, based on initial population, growth rate, and number of years. With the section GO TO, starting on page 19, the reader learns how to make the computer perform this operation on successive groups of data. The section on Demography starts with a program that simply prints what is in the DATA statements, which is present population of the world's major regions, and their growth rates. On page 25, the reader is asked to write his first program, to compute the population in the year 2001 for each of these regions, ending with a grand total.

And so the book goes, enlarging on the population program. The World of IF section is on mathematical relationships, with a program that stops after computing the hundredth year of population growth. Race to Oblivion makes the program more general, inputting two populations and their growth rates, and computing the year in which one overtakes the other. In the example, the population of Latin America overtakes that of Europe in 1994. A section on Beware of Mathematical Models warns against making unwarranted assumptions that can lead to unreasonable computer results.

The section on Count to N is about FOR-TO and NEXT, and introduces a 16-country data base of populations and growth rates; the accompanying program only prints out the data directly, with headings, but without any computing. Subscripted Variables has a 22-line program, but uses the variables merely to print out the data directly, as is done with subroutines in the next section on Building Blocks. Information Retrieval picks population and growth rate from the data base, when the region number is entered. The text notes that it would be nice if the name of the region could be entered instead of its number, and goes on to say that this involves asking someone (or checking the reference manual) about strings.

Page 55 has a variation of the program illustrating subroutines: it asks for two region numbers, then computes when the population of one will overtake the other. Page 60 has a "things to do" section, which gives the same data base as before, but adds birth and death rates, then says "think up ways to use the data and write programs to do so."

Page 61 borrows an idea from Sack & Meadows (27), with a page on Janus, the god with two faces, who "looks backwards . . . and forwards." The text notes that "Now you can speak a little BASIC. But you aren't yet fluent. We have introduced only a primitive form of BASIC. . . ." and then, "This is the end of the beginning. Look ahead . . . and one more thing . . . please recycle this book."

Page 62 recommends the books by Coan (11) and Kemeny & Kurtz (2) "if you want to learn more about BASIC and you like math." For those whose math is "a little wobbly (or non-existent!)" the book by Sharpe & Jacob (17) is recommended.

The typography makes the book look grade-schoolish, but the language is often at senior high-school level.

There are exercises within each chapter, to be run on the computer.

In comparing these 64 pages with the 64 of the NCTM book (4), there is much too much space spent here on using the computer as a printer, and not enough on actual computation. The one-track population theme severely restricts the author to what would be terribly monotonous if the text were any longer. The young beginner deserves a better break.

This text was originally written for Digital Equipment Corp. as *Populution: A Self-Teaching BASIC Primer,* copyrighted in 1972 and priced at $2.00. But before DEC published it, Dymax, as permitted by their contract, published a rewritten version, *My Computer Understands Me,* copyrighted in 1971. Dymax sold 25,000 copies in two years, according to the author.

The main difference in the DEC book is that it has none of the eye-catching headings in large gothic (and other) typefaces, decorative borders, pointing fingers, and a variety of ornaments, including huge quotation marks around don't-forget items (changed to "thought clouds" in the DEC version, as in comic books). DEC does keep the Kilroy head, except for the last one, a full page in the Dymax book, with a huge Kilroy admonishing the reader to "Experiment!!"

The DEC book adds a page on the Teletype keyboard, changes the section on E-notation considerably (with different examples), has a new section on PRINT, adds a chapter on Computer Power, goes into more detail on loops (with a more complex trace output), drops the whole sorcerer's apprentice bit (used to explain infinite loops), adds a program to demonstrate IF-THEN (tracing the program in detail), adds two pages explaining FOR and NEXT, etc.

Both books recommend, at the end, other books on BASIC. The DEC book also gives four sources of information on population, plus a list of DEC publications, under the heading "First, get help from your trustworthy computer supplier, DEC."

The DEC book is typewritten, using roman and italic type, and is much neater, without the flamboyance and flair of the Dymax book, and without the fun.

* * *

29. *Elements of BASIC*, by R. Lewis and B. H. Blakeley. Copyright 1972, by National Computing Centre, Ltd., London, England. 83 pages, 12 x 8½, £1.80 UK (hardcover), $9.00 USA (International Publications Service, 114 E. 32 St., New York, N. Y. 10016).

Succinct and thorough, covers more BASIC than any other book its size. Rating: A

British teenagers are more sophisticated than their American counterparts, to judge from this text, whose introduction says that "In schools, teachers may find that the first four chapters can form the basis of a course for 13-16 year olds." Yet I found myself saying several times while reading, "Are they bringing this up already?" They are indeed, and with quite adequate preparation, providing the reader pays close attention, as there is a minimum of repetition.

Nine chapters make up the book: Introducing BASIC, Extending BASIC Statements, Character Handling, Loops and Arrays, Numerical Methods, Procedures and Subroutines, File Handling, Matrices, A Problem Miscellany.

The first chapter starts right out with a page each on flowcharts, the computer, and the program, and then goes into entering and running a program on a terminal.

At the end of each chapter is a summary of the main points, including the statements introduced. There are exercises within each chapter (without answers); three sets in Chapter 1, for a total of 20 exercises in that chapter. The first exercise is on converting a sum of money in pounds and pence into dollars and cents. The exercises soon grow quite sophisticated: one in the third chapter goes like this: "Write a program which prints out the day of the week for a particular date, given one day and date in the same month."

By page 25, there is no longer an explanation of the program except for the ever-present flowchart. Thus the reader is forced to learn to read and understand flowcharts if he wants to be able to understand the programs.

The book moves quickly to longer programs, and by page 33 has a 28-liner, which asks the user questions about his grades, and then comments on them.

This is one of ten books that introduce INPUT before DATA, perhaps to emphasize the interactive aspect of on-line time-sharing.

By page 35, the book notes that "The end of the data list is indicated by a dummy item, in this case, zero, which is used to terminate the program execution." Nobody else says that much about the dummy zero in just the one sentence.

By page 38, the authors are into an inventory-records program, by page 43 a program with triple loops, and on page 53, computing the area under a curve.

A neat simile is used to explain arrays: "For example, a milkman could note down the amount of milk required for each house in a street in an array such as (1, 6, 2, 3, 1, 0, 4 . . . .)."

Bubble-sorting is explained more thoroughly than most other subjects, with half a page devoted to showing the exact steps involved in sorting six numbers.

REM is mentioned on page 30, although it is never used, perhaps due to space limitations, although there is enough white space in the book to permit dozens of REM statements. On the other hand, statements within a loop are indented in the programs, but no reason is given.

Another terse item: "Thus, if we write Y = RND(X), Y takes on a value between 0 and 1. X can have any value. It is in fact a dummy number which may have some significance in some systems."

Chapter 7, on File Handling, refers to cards in a pocket at the back of the book, which show Filed Program Manipulation on nine different systems (6 US, 3 UK) and the same five programs as run on each; all are inventory programs, such as creating, updating, etc. It would seem to be much easier to put the material on these cards in an appendix.

The last chapter, A Problem Miscellany, contains advanced exercises in eleven areas, with several programs to be written for each. For banking procedures, programs are required for setting up accounts, deposit and withdrawal, money transfer, etc. The last exercise is to write a program for a guessing game such as Hangman. This is the only indication that the book is for schoolchildren.

The longest program is a 57-liner, to sort a list of names of people, first by height and then by weight, as an illustration of subroutines. All programs are in Teletype printout; nearly all are very legible, although a couple are somewhat faint. Nearly all should have been printed larger.

There are very few things that would give an American any difficulty in reading this book; most of the differences are easily understood. "Swop" is used where we would write "swap." One of the computer units is the "store." The decimal point is above the line. And although the text and flowcharts use the dollar sign for strings, the programs themselves use the pound sign; the text says one can use either. Only two items may puzzle the non-British: the gameboard for "snakes and ladders," and the exercise that requires calculating the average runs scored by all the batsmen in a cricket club, including the number of times "not out."

There is a mystery connected with this book: why did the authors (or publishers?) decide to use such an odd size, 12 inches wide and 8½ inches high? The book is printed two columns wide per page, and could easily have been made into a book half as wide, and thus much easier to handle. However, the wider book does stay open more easily.

This is a no-nonsense book, without any sign of the "gee-whiz" attitude found in some American books.

Some good advice on the practical side of programming is given in the last chapter: "The reader should not expect well defined programs and rarely is there any indication of the features of BASIC which should be used."

* * *

30. *A Visual Approach to BASIC,* by Robert E. Smith. Pub. 1972, by Control Data Corp., Minneapolis, Minn., 278 pages, 7 x 10¼, $5.00 (paperback) invoiced, $4.75 cash.

Much too difficult for all but top minds. Rating: A for geniuses, D for the rest of us.

At first glance, this is a book for children. The opening line is "The Lady Cataswank Case really began the night Reggie Cataswank made his way down to Runnymede to seek the help of the Inspector." The drawings show people as caricatures, just as in some children's books.

By page eight, the reader begins to wonder if a child could understand the text, and by page 37 he knows that only a child in the top two percent of the class could. Page eight introduces the flowchart, but too much is presented, and too soon, to the reader who has no teacher to guide him, yet this is supposed to be "a self-instructional guide to computer programming." Page 37 has the first problem for the reader: write a program from a flowchart. Only the brightest reader could do this, with so little previous knowledge provided by the text. There is a difficult problem at the end of each chapter, with answers at the book's end.

The title is somewhat misleading. There are three types of figures in the book: programs, flowcharts, and drawings. Only the drawings are unique; they help tell seven stories, one per chapter. Each story is a "case" involving an inspector; the solution to each case requires a computer program. The author claims that "each unit emphasizes an interesting 'story-type' approach so that the reader is motivated to discover the outcome of each situation, as well as the corresponding programming concepts involved." Although this type of motivation is usually aimed at children, very few children could possibly write the program specified at the end of each chapter. For instance, the flowchart on page 104 requires a program that will, in part, sum the digits of a number. No hints are given on how to do this, which would be difficult for some experienced programmers; in most BASIC systems it has to be done the hard way, with a number of INT statements.

Smith seems to believe, and his previous book (10) lends further weight, that a reader learns much in trying to figure out a program already written, and without any REM statements. A great many readers will give up trying to solve the problem, and turn to the program in the back of the book. Whether or not they will learn anything from this depends on how good they are at digging, and how persistent.

This book could be used by a teacher, if he were to explain thoroughly and fill in with the required lead-in material, and not just aim at the top students in the class, which this text would facilitate. This might be a good book for good students who have already studied BASIC and who want a challenge.

Whether the book is for children or adults, there is no point in using "The Sinfoo Atrocity Case" to set up a problem, with public executions in both text and drawings.

Chapter 4 requires a program to help find eight chests of gold. A very complicated procedure eventually leads, after a dozen pages, to a comparatively simple formulation that is nevertheless not simple to program and which uses much computer time. A great deal of the book is taken up, not in teaching BASIC, but in problem-solving, in working through a number of complex problems to get to a program. The author has let his fascination for such problems obscure the main point.

The most incredible problem involves 25 pages on finding out on which day of the week a person was born, given the date. Those 25 pages could have been put to much better use.

If this book were for children, the author might continue the use of LET in the assignment statement beyond page 18, but he does not.

It may well be true that a story-type approach can motivate the reader to discover the outcome of each situation, buy why make the crux of the story so complex that only a high-IQ reader can get it? Recommended only for Mensa members and others in the top-IQ bracket.

* * *

31. *BASIC, A Computer Programming Language, with Business and Management Applications,* by C. Carl Pegels. Pub. Jan. 15, 1973, by Holden-Day, San Francisco, Calif., 198 pages, 5¾ x 9, $6.95 (paperback).

A useful book, although so terse it's more of a summary of BASIC than a teaching text. Rating: B

The preface calls this a book "intended for those people who would like to be able to use computers to solve problems and help in making decisions – either immediately or some time in the future. Thus, it is written both for practitioners and for students." However, it seems to be more for a programmer than for a beginner, not only because page one says it is for "the programmer who wants to become proficient in programming" and it is "also intended as a reference for the occasional programmer." The author assumes a rather intelligent reader, because there are many sections where much is covered in only a few sentences, giving the book the flavor of a reference manual rather than a text. There is a minimum of the repetition used in many of the other BASIC books to reinforce the impressions made on the reader.

After a rough start, the book improves. This is an example of an author who presents the introductory material somewhat briefly (the first 38 pages) so he can devote most of the book to the applications. If the reader has been able to obtain a firm foundation in those 38 pages, it's much easier from there on, as the writing is better, and there are more explanations and more examples. After presenting a program, the author explains it in full detail.

The author tightens up again, however, for Chapter 6, on programming with data files, page 55. This chapter is written so succinctly, without full explanations and examples, that it would be very hard to understand except by an experienced programmer.

The first program is on page four, and consists of six lines, calculating the cost of a quart of milk. Most of the programs are between 10 and 15 lines long. Pegels is one of the few to go into FILES, and the only one to introduce it as early as page 55. Nearly all the authors assume the reader has some knowledge of vectors and matrices; Pegels includes an eight-page appendix on the subject.

There is a small error in the program on page 23, illustrating loops, that could easily confuse a beginner, who might not be able to figure out that line 150 should read LET X = X + I, rather than LET X = X + 1. So the answer to the program as written is 5 rather than 15. Such an error could cause a misconception on the use of loops that might last for many chapters, and perhaps never be fully dispelled.

There is an oddity on page 32, in a program concerning averages. The average weight of the twelve students is 160.5; the average height is 70.3. If the units are centimeters and kilos, the average student is 63 inches tall and weighs 156 pounds!

There are twelve chapters: Introduction; Elements of BASIC; Looping, Subscripted Variables and More About PRINT; Flowcharting; Functions, Subroutines, Input and String Variables; Programming With Data Files; Matrix Operations; and five chapters on applications, beginning on page 75 with a chapter on statistical problems, including means, deviations, and probabilities. Chapter 9, on business and economic problems, deals with payroll, depreciation, average and marginal cost, breakeven and compound interest. Chapter 10, on production management, covers order point and quantity, ratio scheduling, and learning curves. Chapter 11, on random numbers and simulation, involves simulating simple processes, a "junior merchant's problem simulation," and a queuing simulation. Chapter 12 on corporate financial models gives one program that is rather skimpy on explanations.

There are exercises at the end of each chapter, and complete solutions at the end of the book. The two appendixes are on vectors and matrices, and the teletypewriter terminal.

A unique feature is the presentation of two illustrative programs, on page 20, without explanation; the reader is asked to analyze them. This would be very difficult for a beginner, after only 20 pages, especially since neither program seems to have a practical use.

The text is so brief as to be puzzling in some places. The style is rather choppy.

An uneven book, and a compact one, better as a second book than as a first one for a beginner. The applications portion contains many (39) programs of a wide variety.

32. *BASIC,* by Robert L. Albrecht, LeRoy Finkel, and Jerald R. Brown. Pub. Feb. 9, 1973, John Wiley & Sons, New York, N. Y., 324 pages, 8½ x 11, $3.95 (paperback).

Obviously the result of much teaching of BASIC to young people. Every difficult point is nicely explained in full detail. Rating: A

One of a series of "self-teaching guides," this book says on its front cover, "Teach yourself the quick proven way with programmed instruction." Well, it's not quite programmed instruction. Rather, there are blanks to be filled in, with words or program lines, or even entire programs. Each chapter ends with an excellent "self-test," which also has blanks to be filled in, and is immediately followed by the answers in full, including programs and runs.

There are ten chapters: Getting Started, Warming Up, Decision Making, FOR-NEXT Loops, Functions, Subscripted Variables, Double Subscripts, Subroutines, Advanced BASIC, and Files. The book ends with a final self-test; all answers are given. completes the chapter.

The book proceeds quite simply and slowly, taking 41 pages to teach, in the first chapter, the use of SCRATCH, PRINT, END, RUN and LIST. The reader is slowly led into writing programs, and by page 51 is writing six-line ones. The authors go into greater detail than any others in teaching decision-making, subscripted variables, and DIM; there are 9 pages on inequalities. The section on string variables is detailed and fine.

The writing is informal and easy-going, at the high-school level, even though the preface suggests the book is suitable for college students and adults, who however might find the slow-but-sure method rather drawn-out.

There are very few criticisms of this book. A great deal of white space is used, especially at the bottom of pages; the book could have been several dozen pages shorter. The program on page 128 has data that couldn't possibly produce the computed result.

There are eight pages on statistics, going into variance and standard deviation. Although the text says the reader not familiar with statistics can skip over this section, it does seem out of place in a book so obviously written for high-school students, even though the statistics are used only to demonstrate subroutines. Two pages are used to describe the TYP function, as though it were common, yet Waite & Mather (1) is the only other book that mentions it.

There is a clever program to generate five-letter words with a random selection of letters.

This is the only book other than Dwyer & Kaufman (33) to include something on computer art: three programs provide simple patterns.

The authors use the idea of "little boxes inside the computer" to help teach storage of variables; Dwyer & Kaufman (33) use "mailboxes." Most authors rely on words alone to teach this.

New material sometimes appears in a self-test, but without explanation. On page 261, the reader is asked to write a sort program, without any previous information other than two little hints. As the authors put it, this is a "real programming challenge."

As for content: although only three dozen BASIC statements are covered (including six for files and four for matrices) those three dozen are covered in great detail, giving the reader a solid buildup.

* * *

33. *A Guided Tour of Computer Programming in BASIC,* by Thomas A. Dwyer and Michael S. Kaufman. Pub. July 3, 1973, by Houghton Mifflin, Boston, Mass., 156 pages, 8½ x 11, $3.60 (paperback).

The best of the introductory texts, bright and sparkling. Rating: A

The first thing you notice when opening this book is the engaging illustrations in red and black (by Mark Kelley), cheerful but not cute, 67 of them, adding a nice touch of sparkle to the words. The next feature that hits the eye is the many callouts to the programs, outlined in red, and with a red line pointing to the line or lines they explain. Computer-generated program lines are overprinted in red. These features, plus a text by authors who have given a great deal of thought to the necessary amount of detail required, make a most attractive book.

There are four parts. Getting Ready for the Journey covers the basics of LET, PRINT and END. The Economy Tour introduces six more statements. Techniques for the Seasoned Traveler, nine more statements plus library functions. Far Away Places presents nine applications programs. This book covers 20 statements, 3 commands, and the library functions.

The first part contains a unique section, on How to Recognize a Computer, with pictures and information on minis and "large machines." Page 9 slides neatly into an introduction of flowcharts by showing a "final checklist for time-sharing users." This book is meant to be used with a terminal; the sections in which the reader is asked to use the terminal are indicated by a repeated "ON-LINE" alongside, in red, and sideways.

Page 16 is a perfect model of what an Example of a Normal Session should look like (even grown-up readers would profit greatly from it), with all errors and corrections fully explained with boxed callouts and red-overprinted program lines.

There are several sets of exercises within each of the four parts, and at the end of the book are Selected Answers and Hints for Exercises. At the end of each section in each part is a review of the material covered.

Another excellent section teaches LET with a blackboard, mailboxes, and many examples; they've been used before, but this is the best so far. There is a unique section that compares looping with two different statements, IF-THEN and FOR-NEXT.

The book begins to get a little difficult at page 75, when the reader is asked to write a program that will create a bar graph.

The section on Storing Programs on Paper Tape is the best of all those seen, with several fine figures. The method of introducing subscripted variables is quite clever: an airline-seat reservation system.

The authors are the only ones to provide the proper pronunciation of subscripted variables, by noting that M(8) is pronounced "M sub 8." A map is cleverly used to teach two-dimensional arrays, by asking the reader to obtain statistics on the number of accidents at each intersection.

READ and DATA are not introduced until page 100, as an indication of the stress the authors put on interactive computing. There is a unique section on three ways to get your computer to provide a different random output every time.

Part 4 has five application areas with nine programs, for a hotel reservation system, airline reservation system, generating brand names for a soap, generating menus to choose dishes from in an automated restaurant, slot-machine game (cherries, lemons, oranges), buried-treasure game, monthly installment payments on a loan, same with interest only on the unpaid balance, and payroll.

The last page contains a summary of BASIC, with statements and commands, and for each its name, page where explained, purpose, and an example, more complete than in any other book.

There are few features on the debit side of the ledger. Most authors who wish to indicate that certain idiosyncracies are due to the particular time-sharing system being used, do so in a brief statement early in the book. Dwyer and Kaufman do it in the text, not once, but ten times.

There could be a little more explanation of how to declare array sizes using DIM, of the use of INT in rounding, and of the program for tabulating questionnaire data. There is no explanation at all of the bubble-type sorting program, nor of the buried-treasure program; each needs one.

The authors use both parentheses and brackets, as do Gross & Brainerd (22) but without explaining the difference in usage.

This is, then, a fine book, mainly for young people, but of value for anyone, full of detail, many examples, with much thought having been given to the use of graphics in teaching. This is the best of the introductory texts on BASIC.

* * *

34. *Principles of Data Processing,* by Robert A. Stern and Nancy R. Stern. Pub. Feb. 9, 1973, by John Wiley & Sons, New York, N. Y., 630 pages, 7¼ x 9¼, $12.95 (cloth).

Workbook for Principles of Data Processing, pub. Apr. 30, 1973, 325 pages, 8½ x 11, $5.50 (paper).

BASIC Supplement to accompany Principles of Data Processing, pub. Mar. 27, 1974, 131 pages, 8½ x 11, $4.50 (paper).

The main text and the workbook are excellent for learning all about the fundamentals of business programming, in great detail, and rate an A. However, the BASIC supplement, despite some fine portions, is not up to the same standard, and rates a C.

The main text, the Principles, is a handsome book, very well produced, with excellent use made of color, and with a great many fine illustrations (flowcharts, forms, photographs, card layouts).

There are 20 chapters in three sections. Section One is on fundamental concepts of data processing, with six chapters: overview, business organization and the role of the data-processing department, punched card and printed report, processing of data by computer, input/output devices, a guide to terminals and time-sharing. Section Two, on Concepts of Computer Programming, has nine chapters: steps in programming, flowcharting, introductions to COBOL, RPG, FORTRAN, PL/1, BASIC, software: control and optimization of computer capability, common programming techniques. Section Three, on Systems Analysis and Design, has five chapters: systems analysis, systems design, two case studies (accounts receivable, inventory system), and management information systems and other decision-making techniques.

The preface says that "this textbook differs significantly from the data processing texts currently in use. It introduces the concepts of data processing as they actually relate to the business world .... We do not attempt to idealize data processing." All very true.

At the end of the first chapter is a Self-Evaluating Quiz, with 16 completion-type and true-and-false questions, with the answers following immediately. There is such a quiz at the end of each chapter, and also in the middle of the longer chapters.

Chapter 2, on business organization, contains organization charts for retail and manufacturing companies to help illustrate "the flow of information within a typical company." Chapter 5, on input/output devices, looks into the card read/punch, printer, tape drives (and details such as labels, and file protection rings), disks, specialized equipment (such as MICR, OCR, paper tape, offline devices). An 8-page table compares the characteristics of 26 computers, from the Burroughs B3500 to the XDS Sigma 6. Chapter 8, on Program Flowcharting, has examples in inventory, banking, accounts receivable, and updating.

Chapter 9, Introduction to COBOL Programming, shows filled-in coding sheets, card formats, output tape format, flowcharts, and sample programs for payroll, accounts receivable, salesmen's commissions, and is 30 pages long. The chapter on RPG is similar, with the same three types of programs, 38 pages long. The chapter on FORTRAN has a payroll program, then compares FORTRAN with COBOL; 13 pages. Chapter 12, on PL/1, has the three programs; 20 pages. The 12-page BASIC chapter contains five illustrative programs of 5 to 10 lines (simple calculation, temperature conversion, weekly wages, transaction amount, monthly sales report).

Chapter 16 has a hefty title: The Interaction Between the Analyst and the Businessman: Systems Analysis, with information on the basic elements, collecting data, analysis of current system costs, and problem definition.

Chapter 18, a case study of accounts receivable, is in detail, with 18 illustrations of reports, memos, flowcharts, card formats, and disk layouts.

Chapter 20, on Management Information Systems and Other Decision-Making Techniques, contains brief examples of typical companies using MIS: Westinghouse, Boeing, Chemical Bank of New York, and sections on CPM and PERT charts, simulation, linear programming.

The book ends with a 44-page appendix on Numbering Systems and Their Significance in Computer Processing, with sections on binary, octal and hexadecimal numbers, and representation of characters in storage.

The *workbook* chapters parallel those of the main text, complementing them very well, providing excellent information, and asking many good questions. As the preface puts it, "The workbook . . . includes applications in specified business areas, designed as realistic illustrations of data processing applications and as a vehicle for classroom discussion."

"Each chapter is subdivided into the following topics: define the following terms, answer true or false, multiple choice, fill in the missing blanks, and applications. [The first chapter aks for 15 definitions, 20 true-or-false, 10 multiple-choice, and 20 fill-ins.] The emphasis, throughout the Workbook, is on . . . Applications. This section contains data processing forms and layouts to be completed by the student in an effort to familiarize him or her with professional standards in this area."

The Applications portion of Chapter 1 contains two articles from *Business Automation,* which one reads and then answers eleven essay-type questions. One question is, "Don't you think that the system described above is an invasion of privacy? Explain your answer." Chapter 2 follows the same scheme; the applications portion discusses the acquisition of a computer by a Long Island department store, and considers the operations and organizations of each department.

Subsequent applications sections look into applications (payroll, banking, sales, accounts receivable), evaluation of a computer set-up (Honeywell 3200), a mini-computer application, OCR, online application (law enforcement, Burroughs), credit cards, betting (Varian), inventory system (Honeywell), personnel, etc. Most of these are based on application reports from computer manufacturers.

The chapter on systems analysis has a section on "the interview technique for collecting data," complete with three dialogues.

So far the authors have done beautifully. But then we come to the *BASIC supplement,* which is one of four; the others are on COBOL, FORTRAN, and RPG. There is also an instructor's manual that includes answers to all workbook questions. These four language supplements, by some Procrustean device, are each 150 pages long, according to the back cover of the one on BASIC.

The preface notes that "The [supplement] is written in the style of a programmed-instruction text. Each unit is followed by a series of self-evaluating questions that test the student's understanding of the material presented . . Each question is followed by five asterisks (*****) which provide a signal that the solution follows. It is recommended that the student read each question, using a card or sheet to cover the solution."

"Each chapter is followed by a series of Review Questions and Problems for which solutions are not provided. These may be used as homework assignments or as the basis for classroom participation."

There are six chapters: Terminal Processing With the Use of BASIC, Essential Elements of a BASIC Program, Writing Simple Programs in BASIC, The PRINT Statement and Its Options, Loops and Arrays, Functions and Subroutines. This supplement covers 16 statements, 5 commands, and 11 predefined functions.

By page 4 there have already been five questions. Chapter 1 has eight groups of 42 questions in all. So a third of the first chapter is question-and-answer, a rather high percentage. Counting the Review Questions at the chapter's end, the percentage of Q&A is 39%. This reinforcement technique may appeal to many; others may prefer fewer questions and more teaching text. Also, the Q&A method used takes much space: on page 8, some 39 square inches are used to present only four Q&A.

The text continually tells of different ways that different types of terminals might handle a particular situation; this is done nicely, not so much as to be boring or confusing.

The actual BASIC text is only about 63 pages in length, less than half the 131 numbered pages.

There are 21 programs in the book, with very few runs. The first program is a two-liner that prints square roots. The next three programs expand on that, using INPUT, READ, and DATA. The fifth program is 11 linches long, on calculating salesmen's bonuses, and is explained thoroughly. The sixth program computes averages, and is one of the few with output. Most of the programs are 4 to 8 lines long; the 19th, which computes exam averages for five students, is 15 lines long. The last program is 21 lines long, demonstrates subroutines, has no RUN.

The explanation of user-defined functions is possibly the best in all these books. There is some nice detail in describing the technique of looping: "initialize a field . . ., test the value of that field . . ., modify that field . . ., return to the beginning of the sequence." This book contains the best (most explicit) explanation of incrementing a counter, as when taking an average. It is one of the most painstakingly thorough in explaining each new statement.

For some odd reason, the only terminal shown is a CRT model; no Teletype is shown. There is nothing on rounding, RANDOM, etc. A curious error is the period at the end of a PRINT line on page 60.

The biggest fault of the book, for some readers, will be the constant annoyance of questions and answers appearing so often. For instance, Chapter 4, on PRINT, has 1½ pages of text, then ½ page of Q&A, a page of text, 2 of Q&A, 1½ text, 2 Q&A, ½ text, 2 Q&A, 1½ text, 2 Q&A, 2 text, 1 Q&A, and a page of review questions. That's fine for the reader who likes constant reinforcement, but not for the reader who prefers to ask his own questions.

Each chapter begins with a page devoted to section headings, which wastes much space, especially the first page of Chapter 6, which has only eleven words and two numbers on it.

Certain important points are boxed, but so are a couple of programs in the first chapter, so there is no apparent overall scheme.

Although there are a number of partial programs in the book, to explain things such as STOP, there are all too few complete programs, and almost no runs.

* * *

Interactive Computing in BASIC, by Peter Sanderson. Pub. 1973 by the Butterworth Group, London, £4.00 (hardcover), £2.00 (paperback).

(Review copy requested but not received.)

* * *

Simplifield BASIC Programming: With Companion Problems, by Lisa and Judah Rosenblatt. Pub. June 1973 by Addison-Wesley, Reading, Mass., 313 pages, 6¼ x 9¼, $3.95 (paper).

This is one of several books not being reviewed here because they are much more oriented toward applications than toward the teaching of BASIC. This particular one, for instance, has 82 pages on BASIC, followed by 172 pages of applications in the fields of mathematics, business, social science, and calculus (and 51 pages of solutions). With two-thirds of the book being on applications, and especially since there isn't a single word about BASIC in those 172 pages, it seems to fit better into my next group review, which (God and the publishers willing) will cover books on applications of BASIC.

CREATIVE COMPUTING Reviews

Reviews Editor: Peter Kugel, School of Management, Boston College, Chestnut Hill, MA 02167.

Readers: Want to be a reviewer? Write to the Reviews Editor directly. Publishers: send materials for review to the Reviews Editor.

ANS COBOL. Ruth Ashley. 242 pp. *$4.95.* John Wiley & Sons, Inc. (Wiley Self-teaching guides), New York, 1974.

Ruth Ashley's American National Standard COBOL is an acceptable self-teaching book and classroom text for those who do not have access to a computer terminal and do have previous programming experience. A general understanding of programming and flowcharting is assumed. The author displays her knowledge of the elementary errors students have when beginning to program the most widely used computer language in business today.

Each chapter starts with a list of behavioral objectives and proceeds in the standard programmed text method of frames followed by multiple choice questions. The answers which directly follow the questions are supplemented with detailed and clear reasons for those answers. The student is provided with strips of COBOL coding forms for writing his or her programs. Throughout the book, much attention is given to spelling, punctuation, and format as is required in this language.

The author suggests that students try to complete each of the ten chapters in one or two study sessions of 1 to 3 hours each. The first chapter is devoted to the different divisions. The final question asks the student to write a complete small program. The second and third chapters cover MOVE, COMPUTE, GO TO, IF and arithmetic operations. The different types of files and editing are covered in the next four chapters. Chapter eight clearly presents PERFORM options. The last two chapters on tables and desk files complete a well planned text for anyone who is interested in an inexpensive text for high school or college students beginning COBOL. Teachers will find a complete lesson plan prepared by a teacher who understands student errors.

Paul A. Chase
Leicester, MASS.

[Available from the Creative Computing Book Service. Catalog No. 7H. Use order form on inside back cover.]

COMPUTER POWER AND HUMAN REASON

Computer Power and Human Reason: From Judgement to Calculation, by Joseph Weizenbaum, W.H. Freeman & Co., 1976, 300 pp., $9.95.

Computer Power and Human Reason: From Judgement to Calculation is probably the most significant computer book issued in 1976, if not for the last decade. It is basically an account of the impact of scientific technology on man's self-image. It is a distinguished computer scientist's probe of the limits of computer power and of scientific rationality itself. Above all, it is a defense of the sanctity of the human spirit.

Presented here are three reviews of the book. The first one by *Creative Computing* Contributing Editor John Lees examines mostly the human freedom and ethical issues. The second by John McCarthy of Stanford focuses on inconsistencies in the book and defends various people and institutions that were attacked by Weizenbaum. In addition, McCarthy introduces some new issues on "What worries about computers are warranted?". The third review by *Creative Computing* Reviews Editor Peter Kugel focuses on the questions "What should one think and what should one work on?"

We expected to have Weizenbaum's rebuttal to these reviews in this issue, but alas it did not arrive by the typeset deadline. Maybe next issue.

The book itself is available through the *Creative Computing* Library for $9.95 plus 75¢ shipping (USA) or $1.75 (elsewhere). Creative Computing Library, P.O. Box 789-M, Morristown, NJ 07960.

If, after reading the book and reviews, you wish to correspond with Weizenbaum or any of the reviewers, here are their addresses. Please send a carbon of any correspondence to us at *Creative*, too. —DHA

John Lees
P.O. Box 1543
Rolla, MO 65401

John McCarthy
Artificial Intelligence Laboratory
Stanford, CA 94305

Peter Kugel
School of Management
Boston College
Chestnut Hill, MA 02167

Joseph Weizenbaum
Computer Science Lab, Room 522
545 Technology Square
Cambridge, MA 02139

A Return to Freedom and Dignity

The twentieth century has been a century of the triumph of science and technology. The attempt has been made to bring all things in our universe, if not under the control, at least within the scientific understanding of man. In 1971, with the publication of *Beyond Freedom and Dignity*, B.F. Skinner tried to reduce all of human affairs to the scope of a technology of behavior; a culmination of the viewpoint that technology can and *should* be used to exercise complete control of the world. In the realm of computer science, the Artificial Intelligentsia are claiming that man can be understood totally as an information processing system, an attitude which has lead to an increasingly mechanical view of man and society, with no place remaining for responsible ethics. In his book, Joseph Weizenbaum, Professor of Computer Science at MIT, explores two major questions: whether there is a difference between man and machine, and whether there are tasks for which computers *ought* not to be used, even if it is possible for computers to be used for those tasks.

In the 1960's Professor Weizenbaum composed a computer program, named ELIZA, which could hold a "conversation" in English with a human partner. Supplied with a "script" for playing psychiatrist, ELIZA, known as DOCTOR, became somewhat famous as one of the first programs of its kind. As is too often the case, many people mistook a clever demonstration as evidence that the computer was a "person who could be appropriately and usefully addressed in intimate terms." When some psychiatrists suggested that improved versions of ELIZA might do much of the work of a human therapist, Professor Weizenbaum was awakened to what Michael Polanyi has called "a scientific outlook that appears to have produced a mechanical conception of man," and he was disturbed by the "enormously exaggerated attributions an even well-educated audience is capable of making, even strives to make, to a technology it does not understand." From these feelings and misgivings came the desire for clarification which led to *Computer Power and Human Reason*.

For some purposes, man can be thought of as an information processing system, but intelligent machines will not be human machines.

Concerned with where computers fit into human society and with how the use of computers is changing our view of ourselves, Professor Weizenbaum believes that there *are* differences between man and machine and that this means that there are certain tasks which machines ought not to perform. He points out that "however much intelligence computers may attain, now or in the future, theirs must always be an intelligence *alien* to genuine human problems and concerns." He does not discount the possibility that we may someday succeed in constructing very intelligent machines, nor does he disagree that, for some purposes, man can be thought of as an information processing system, but *intelligent* machines will not be *human* machines; it is not right for machines to perform tasks which are in essence human, and it is demeaning to deal with humans in the same terms in which one deals with machines.

Professor Weizenbaum does not give a rigorous argument for his belief that there are fundamental differences between man and machine, but he is not writing as a metaphysician. It is in any case doubtful that man will be mature enough to answer that question with finality for many thousands of years. We now know almost nothing about such things as human neurophysiology and what is really going on in the brain. The task of translating from one natural language to another, once believed simple, has proved to be astoundingly complex, requiring a machine which can deal with context and meaning as seen by human beings belonging to different cultures. It is obvious that machines with human capabilities, even if possible, are so far off as to be practically impossible.

The real question is *why do we want machines with human capabilities?* Are we trying to bring about the Trafalmadorian situation of asking our machines what we are good for and receiving the reply that we are good for nothing since the machines

If we treat each other like machines, if we act like machines, then we will become no longer human.

can do everything a human can do and more? Professor Weizenbaum believes the desire to manufacture machines which can perform such human functions as therapist or judge to be odious and obscene: "The point is that there are some human functions for which computers *ought* not to be substituted. It has nothing to do with what computers can or cannot be made to do. Respect, understanding, and love are not technical problems."

This is the point he strives to make throughout the book and particularly in his concluding chapter. Not all of our problems have technical solutions. Human problems require human solutions, not because no other solutions are possible, but because we are in danger of losing our humanity if we do not act to retain our humanity. If we treat each other like machines, if we act like machines, then we will become no longer human. We are human not because we are born with hands and faces; we are human through our *actions*, through our ability to make eithical decisions of right and wrong, and because we can perceive higher purposes in our lives than being cogs in a machine.

So many of the world's ills can be traced to seeing the world as a machine and men as machines within a machine. That attitude turns forests into lumber, lakes into sewage, land into property, and people into consumers; it allows otherwise sane men to calmly do research leading to improved methods of killing people, without ever questioning the morality of their actions; it allows data processors to treat people like objects which must be standardized, optimized, and made to conform; it allows schools to 'produce' graduates for the job market who have skills, but lack the necessary understanding of the power and limitations of their tools.

The mechanical conception of man all too often is used as an excuse for taking no action where action is needed. Too many see their lives as simply the inevitable result of a process which they cannot affect. This is a seriously dangerous attitude, and Professor Weizenbaum calls for people to rise above it and assert their freedom, their dignity, and their humanity:

"It is a widely held but a grievously mistaken belief that civil courage finds exercise only in the context of world-shaking events. To the contrary, its most arduous exercise is often in those small contexts in which the challenge is to overcome the fears induced by petty concerns over career, over our relationships to those who appear to have power over us, over whatever may disturb the tranquility of our mundane existence.

"If this book is to be seen as advocating anything, then let it be a call to this simple kind of courage. And, because this book is, after all, about computers, let that call be heard mainly by teachers of computer science."

John Lees
Rolla, MO

*****

An Unreasonable Book*

This moralistic and incoherent book uses computer science and technology as an illustration to support the view promoted by Lewis Mumford, Theodore Roszak, and Jacques Ellul, that science has led to an immoral view of man and the world. I am frightened by its arguments that certain research should not be done if it is based on or might result in an "obscene" picture of the world and man. Worse yet, the book's notion of "obscenity" is vague enough to admit arbitrary interpretations by activist bureaucrats.

IT'S HARD TO FIGURE OUT WHAT HE REALLY BELIEVES ...

Weizenbaum's style involves making extreme statements which are later qualified by contradictory statements. Therefore, almost any quotation is out of context, making it difficult to summarize his contentions accurately.

The following passages illustrate the difficulty:

"In 1935, Michael Polanyi", [British chemist and philosopher of science, was told by] *"Nicolai Bukharin, one of the leading theoreticians of the Russian Communist party, ...* [*that*] *'under socialism the conception of science pursued for its own sake would disappear, for the interests of scientists would spontaneously turn to the problems of the current Five Year Plan.' Polanyi sensed then that 'the scientific outlook appeared to have produced a mechanical conception of man and history in which there was no place for science itself.' And further that 'this conception denied altogether any intrinsic power to thought and thus denied any grounds for claiming freedom of thought.'"* - from page 1. Well, that's clear enough; Weizenbaum favors freedom of thought and science and is worried about threats to them. But on page 265, we have

"Scientists who continue to prattle on about 'knowledge for its own sake' in order to exploit that slogan for their self-serving ends have detached science and knowledge from any contact with the real world". Here Weizenbaum seems to be against pure science, i.e. research motivated solely by curiosity. We also have

"With few exceptions, there have been no results, from over twenty years of artificial intelligence research, that have found their way into industry generally or into the computer industry in particular". - page 229. This again suggests that industrial results are necessary to validate science.

"Science promised man power. But as so often happens when people are seduced by promises of power ... the price actually paid is servitude and impotence". This is from the book jacket. Presumably the publisher regards it as a good summary of the book's main point.

"I will, in what follows, try to maintain the position that there is nothing wrong with viewing man as an information processor (or indeed as anything else) nor with attempting to understand him from that perspective, providing, however, that we never act as though any single perspective can comprehend the whole man." - page 140. We can certainly live with that, but

"Not only has our unbounded feeding on science caused us to become dependent on it, but, as happens with many other drugs taken in increasing dosages, science has been gradually converted into a slow acting poison". - page 13. These are qualified by

"I argue for the rational use of science and technology, not for its mystification, let alone its abandonment". - page 256.

In reference to the proposal for a moratorium on certain experiments with recombinant DNA because they might be dangerous, we have *"Theirs is certainly a step in the right direction, and their initiative is to be applauded. Still, one may ask, why do they feel they have to give a reason for what they recommend at all? Is not the overriding obligation on men, including men of science, to exempt life itself from the madness of treating everything as an object, a sufficient reason, and one that does not even have to be spoken? Why does it have to be explained? It would appear that even the noblest acts of the most well-meaning people are poisoned by the corrosive climate of values of our time."* Is Weizenbaum against all experimental biology or even all experiments with DNA? I would hesitate to conclude so from this quote; he may say the direct opposite somewhere else.

"Those who know who and what they are do not need to ask what they should do." - page 273. Let me assure the reader that there is nothing in the book that offers any way to interpret this pomposity. The menace of such grandiloquent precepts is that they require a priesthood to apply them to particular cases, and would-be priests quickly crystallize around any potential center of power. A corollary of this is that people can be attacked for what they are rather than for anything specific they have done. The April 1976 issue of *Ms.* has a poignant illustration of this in an article about "trashing".

"An individual is dehumanized whenever he is treated as less than a whole person". - page 266. This is also subject to priestly interpretation as in the encounter group movement.

"The first kind [of computer application] *I would call simply obscene. These are ones whose very contemplation ought to give*

*This review is appearing simultaneously in the ACM SIGART Newsletter and originally "appeared" in a public file on the ARPA net.

rise to feelings of disgust in every civilized person. The proposal I have mentioned, that an animal's visual system and brain be coupled to computers, is an example. It represents an attack on life itself. One must wonder what must have happened to the proposers' perception of life, hence to their perceptions of themselves as part of the continuum of life, that they can even think of such a thing, let alone advocated it". No argument is offered that might be answered, and no attempt is made to define criteria of acceptability. I think Weizenbaum and the scientists who have praised the book may be surprised at some of the repressive uses to which the book will be put. However, they will be able to point to passages in the book with quite contrary sentiments, so the repression won't be their fault.

BUT HERE'S A TRY AT SUMMARIZING

As these inconsistent passages show, it isn't easy to determine Weizenbaum's position, but the following seem to be the book's main points.

1. Computers cannot be made to reason usefully about human affairs. This is supported by quoting over-optimistic predictions by computer scientists and giving examples of non-verbal human communication. However, Weizenbaum doesn't name any specific task that computers cannot carry out, because he wishes *"to avoid the unnecessary, interminable, and ultimately sterile exercise of making a catalogue of what computers will and will not be able to do, either here and now or ever."* It is also stated that human and machine reasoning are incomparable and that the sensory experience of a human is essential for human reasoning.

2. There are tasks that computers should not be programmed to do. Some are tasks Weizenbaum thinks shouldn't be done at all—mostly for new left reasons. One may quarrel with his politics, and I do, but obviously computers shouldn't do what shouldn't be done. However, Weizenbaum also objects to computer hookups to animal brains and computer conducted psychiatric interviews. As to the former, I couldn't tell whether he is an anti-vivisectionist, but he seems to have additional reasons for calling them "obscene". The objection to computers doing psychiatric interviews also has a component beyond the conviction that they would necessarily do it badly. Thus he says, *"What can the psychiatrist's image of his patient be when he sees himself, as a therapist, not as an engaged human being acting as a healer, but as an information processor following rules, etc.?"* This seems like the renaissance era religious objections to dissecting the human body that came up when science revived. Even the Popes eventually convinced themselves that regarding the body as a machine for scientific or medical purposes was quite compatible with regarding it as the temple of the soul. Recently they have taken the same view of studying mental mechanisms for scientific or psychiatric purposes.

3. Science has led people to a wrong view of the world and of life. The view is characterized as mechanistic, and the example of clockwork is given. (It seems strange for a computer scientist to give this example, because the advance of the computer model over older mechanistic models is that computers can and clockwork can't make decisions.) Apparently analysis of a living system as composed of interacting parts rather than treating it as an unanalyzed whole is bad.

4. Science is not the sole or even main source of reliable general knowledge. However, he doesn't propose any other sources of knowledge or say what the limits of scientific knowledge is except to characterize certain thoughts as "obscene".

5. Certain people and institutions are attacked. These include the Department of "Defense" (sic), *Psychology Today*, the *New York Times* Data Bank, compulsive computer programmers, Kenneth Colby, Marvin Minsky, Roger Schank, Allen Newell, Herbert Simon, J.W. Forrester, Edward Fredkin, B.F. Skinner, Warren McCulloch (until he was old), Laplace and Leibniz.

6. Certain political and social views are taken for granted. The view that U.S. policy in Vietnam was "murderous" is used to support an attack on "logicality" (as opposed to "rationality") and the view of science as a "slow acting poison". The phrase *"It may be that the people's cultivated and finally addictive hunger for private automobiles ..."* (p. 30) makes psychological, sociological, political, and technological presumptions all in one phrase. Similarly, *"Men could instead choose to have truly safe automobiles, decent television, decent housing for everyone, or comfortable, safe, and widely distributed mass transportation."* presumes wide agreement about what these things are, what is technologically feasible, what the effects of changed policies would be, and what activities aimed at changing people's taste are permissible for governments.

THE ELIZA EXAMPLE

Perhaps the most interesting part of the book is the account of his own program ELIZA that parodies Rogerian non-directive psychotherapy and his anecdotal account of how some people ascribe intelligence and personality to it. In my opinion, it is quite natural for people who don't understand the notion of algorithm to imagine that a computer computes analogously to the way a human reasons. This leads to the idea that accurate computation entails correct reasoning and even to the idea that computer malfunctions are analogous to human neuroses and psychoses. Actually, programming a computer to draw interesting conclusions from premises is very difficult and only limited success has been attained. However, the effect of these natural misconceptions shouldn't be exaggerated; people readily understand the truth when it is explained, especially when it applies to a matter that concerns them. In particular, when an executive excuses a mistake by saying that he placed excessive faith in a computer, a certain skepticism is called for.

Colby's (1973) study is interesting in this connection, but the interpretation below is mine. Colby had psychiatrists interview patients over a teletype line and also had them interview his PARRY program that simulates a paranoid. Other psychiatrists were asked to decide from the transcripts whether the interview was with a man or with a program, and they did no better than chance. However, since PARRY is incapable of the simplest causal reasoning, if you ask, "How do you know the people following you are Mafia" and get a reply that they look like Italians, this must be a man not PARRY. Curiously, it is easier to imitate (well enough to fool a psychiatrist) the emotional side of a man than his intellectual side. Probably the subjects expected the machine to have more logical ability, and this expectation contributed to their mistakes. Alas, random selection from the directory of the Association for Computing Machinery did no better.

It seems to me that ELIZA and PARRY show only that people, including psychiatrists, often have to draw conclusions on slight evidence, and are therefore easily fooled. If I am right, two sentences of instruction would allow them to do better.

In his 1966 paper on ELIZA (cited as 1965), Weizenbaum writes,

"One goal for an augmented ELIZA program is thus a system which already has access to a store of information about some aspect of the real world and which, by means of conversational interaction with people, can reveal both what it knows, i.e. behave as an information retrieval system, and where its knowledge ends and needs to be augmented. Hopefully the augmentation of its knowledge will also be a direct consequence of its conversational experience. It is precisely the prospect that such a program will converse with many people and learn something from each of them which leads to the hope that it will prove an interesting and even useful conversational partner." Too bad he didn't successfully pursue this goal; no-one else has. I think success would have required a better understanding of formalization than is exhibited in the book.

WHAT DOES HE SAY ABOUT COMPUTERS?

While Weizenbaum's main conclusions concern science in general and are moralistic in character, some of his remarks about computer science and AI are worthy of comment.

1. He concludes that since a computer cannot have the experience of a man, it cannot understand a man. There are three points to be made in reply. First, humans share each other's experiences and those of machines or animals only to a limited extent. In particular, men and women have different experiences. Nevertheless, it is common in literature for a good writer to show greater understanding of the experience of the opposite sex than a poorer writer of that sex. Second, the notion of experience is poorly understood; if we understood it better, we could reason about whether a machine could have a simulated or vicarious experience normally confined to humans. Third, what we mean by understanding is poorly understood, so we don't yet know how to define whether a machine understands something or not.

2. Like his predecessor critics of artificial intelligence, Taube, Dreyfus and Lighthill, Weizenbaum is impatient, implying that if the problem hasn't been solved in twenty years, it is time to give up. Genetics took about a century to go from Mendel to the genetic code for proteins, and still has a long way to go before we will fully understand the genetics and evolution of intelligence and behavior. Artificial intelligence may be just as difficult. My

What would it mean for a computer to hope for love? The answer depends on being able to formalize (not simulate) the phenomena in question.

current answer to the question of when machines will reach human-level intelligence is that a precise calculation shows that we are between 1.7 and 3.1 Einsteins and .3 Manhattan Projects away from the goal. However, the current research is producing the information on which the Einstein will base himself and is producing useful capabilities all the time.

3. The book confuses computer simulation of a phenomenon with its formalization in logic. A simulation is only one kind of formalization and not often the most useful — even to a computer. In the first place, logical and mathematical formalizations can use partial information about a system insufficient for a simulation. Thus the law of conservation of energy tells us much about possible energy conversion systems before we define even one of them. Even when a simulation program is available, other formalizations are necessary even to make good use of the simulation. This review isn't the place for a full explanation of the relations between these concepts.

Like *Punch's* famous curate's egg, the book is good in parts. Thus it raises the following interesting issues:

1. What would it mean for a computer to hope or be desperate for love? Answers to these questions depend on being able to formalize (not simulate) the phenomena in question. My guess is that adding a notion of hope to an axiomatization of belief and wanting might not be difficult. The study of *propositional attitudes* in philosophical logic points in that direction.

2. Do differences in experience make human and machine intelligence necessarily so different that it is meaningless to ask whether a machine can be more intelligent than a machine? My opinion is that comparison will turn out to be meaningful. After all, most people have no doubt that humans are more intelligent than turkeys. Weizenbaum's examples of the dependence of human intelligence on sensory abilities seem even refutable, because we recognize no fundamental difference in humanness in people who are severely handicapped sensorily, e.g. the deaf, dumb and blind or paraplegics.

IN DEFENSE OF THE UNJUSTLY ATTACKED — SOME OF WHOM ARE INNOCENT

Here are defenses of Weizenbaum's targets. They are not guaranteed to entirely suit the defendees.

Weizenbaum's conjecture that the Defense Department supports speech recognition research in order to be able to snoop on telephone conversations is biased, baseless, false, and seems motivated by political malice. The committee of scientists that proposed the project advanced quite different considerations, and the high officials who made the final decisions are not ogres. Anyway their other responsibilities leave them no time for complicated and devious considerations. I put this one first, because I think the failure of many scientists to defend the Defense Department against attacks they know are unjustified, is unjust in itself, and furthermore has harmed the country.

Weizenbaum doubts that computer speech recognition will have cost-effective applications beyond snooping on phone conversations. He also says, *"There is no question in my mind that there is no pressing human problem that will be more easily solved because such machines exist."* I worry more about whether the programs can be made to work before the sponsor loses patience. Once they work, costs will come down. Winograd pointed out to me that many possible household applications of computers may not be feasible without some computer speech recognition. One needs to think both about how to solve recognized problems and about opportunities to put new technological possibilities to good use. The telephone was not invented by a committee considering already identified problems of communication.

Referring to *Psychology Today* as a cafeteria simply excites the snobbery of those who would like to consider their psychological knowledge to be above the popular level. So far as

There is a whole chapter attacking "compulsive computer programmers" or "hackers." This mythical beast lives in the computer laboratory, is an expert on all the ins and outs of the time-sharing system, elaborates the time-sharing system with arcane features that he never documents, and is always changing the system before he even fixes the bugs in the previous version.

I know, professional and academic psychologists welcome the opportunity offered by *Psychology Today* to explain their ideas to a wide public. They might even buy a cut-down version of Weizenbaum's book if he asks them nicely. Hmm, they might even buy this review. (No, they didn't.)

Weizenbaum has invented a *New York Times Data Bank* different from the one operated by the *New York Times* — and possibly better. The real one stores abstracts written by humans and doesn't use the tapes intended for typesetting machines. As a result the user has access only to abstracts and cannot search on features of the stories themselves, i.e. he is at the mercy of what the abstractors thought was important at the time.

Using computer programs as psychotherapists, as Colby proposed, would be moral if it would cure people. Unfortunately, computer science isn't up to it, and maybe the psychiatrists aren't either.

I agree with Minsky in criticizing the reluctance of art theorists to develop formal theories. George Birkhoff's formal theory was probably wrong, but he shouldn't have been criticized for trying. The problem seems very difficult to me, and I have made no significant progress in responding a challenge from Arthur Koestler to tell how a computer program might make or even recognize jokes. Perhaps some reader of this review might have more success.

There is a whole chapter attacking "compulsive computer programmers" or "hackers." This mythical beast lives in the computer laboratory, is an expert on all the ins and outs of the time-sharing system, elaborates the time-sharing system with arcane features that he never documents, and is always changing the system before he even fixes the bugs in the previous version. All these vices exist, but I can't think of any individual who combines them, and people generally outgrow them. As a laboratory director, I have to protect the interests of people who program only part time against tendencies to over-complicate the facilities. People who spend all their time programming and who exchange information by word of mouth sometimes have to be pressed to make proper writeups. The other side of the issue is that we professors of computer science sometimes lose our ability to write actual computer programs through lack of practice and envy younger people who can spend full time in the laboratory. The phenomenon is well known in other sciences and in other human activities.

Weizenbaum attacks the Yale computer linguist, Roger Schank, as follows — the inner quotes are from Schank *"What is contributed when it is asserted that 'there exists a conceptual base that is interlingual, onto which linguistic structures in a given language map during the understanding process and out of which such structures are created during generation (of linguistic utterances)? Nothing at all. For the term 'conceptual base' could perfectly well be replaced by the word 'something.' And who could argue with that so-transformed statement?"* Weizenbaum goes on to say that the real scientific problem "remains as untouched as ever." On the next page he says that unless the "Schank-like scheme" understood the sentence *"Will you come to dinner with me this evening?"* to mean *"a shy young man's desperate longing for love,* then the sense in which the system "understands" is "about as weak as the sense in which ELIZA "understood." This good example raises interesting issues and seems to call for some distinctions. Full understanding of the sentence indeed results in knowing about the young man's desire for love, but it would seem that there is a useful lesser level of understanding in which the machine would know only that he would like her to come to dinner.

Contrast Weizenbaum's demanding, more-human-than-thou attitude to Schank and Winograd with his respectful and even obsequious attitude to Chomsky. We have *"The linguist's first task is therefore to write grammars, that is, sets of rules, of particular languages, grammars capable of characterizing all and only the grammatically admissible sentences of those languages, and then to postulate principles from which crucial features of all such grammars can be deduced. That set of principles would then constitute a universal grammar. Chomsky's hypothesis is, to put it another way, that the rules of such a universal grammar would constitute a kind of projective description of important aspects of the human mind."* There is nothing here demanding that the universal grammar take into account the young man's desire for love. As far as I can see, Chomsky is just as much a rationalist as we artificial intelligentsia.

Chomsky's goal of a universal grammar and Schank's goal of a conceptual base are similar, except that Schank's ideas are further developed, and the performance of his students' programs can be compared with reality. I think they will require drastic revision and may not be on the right track at all, but then I am pursuing a rather different line of research concerning how to represent the basic facts that an intelligent being must know about the world. My idea is to start from epistemology rather than from language, regarding their linguistic representation as secondary. This approach has proved difficult, has attracted few practitioners, and has led to few computer programs, but I still think it's right.

Weizenbaum approves of the Chomsky school's haughty attitude towards Schank, Winograd and other AI based language researchers. On page 184, he states, *"many linguists, for example, Noam Chomsky, believe that enough thinking about language remains to be done to occupy them usefully for yet a little while, and that any effort to convert the present theories into computer models would, if attempted by the people best qualified, be a diversion from the main task. And they rightly see no point to spending any of their energies studying the work of the hackers."*

This brings the chapter on "compulsive computer programmers" alias "hackers" into a sharper focus. Chomsky's latest book *Reflections on Language* makes no reference to the work of Winograd, Schank, Charniak, Wilks, Bobrow or William Woods to name only a few of those who have developed large computer systems that work with natural language and who write papers on the semantics of natural language. The actual young computer programmers who call themselves hackers and who come closest to meeting Weizenbaum's description don't write papers on natural language. So it seems that the hackers whose work need not be studied are Winograd, Schank, et. al. who are professors and senior scientists. The Chomsky school may be embrassed by the fact that it has only recently arrived at the conclusion that the semantics of natural language is more fundamental than its syntax, while AI based researchers have been pursuing this line for fifteen years.

The outside observer should be aware that to some extent this is a pillow fight within M.I.T. Chomsky and Halle are not to be dislodged from M.I.T. and neither is Minsky — whose students have pioneered the AI approach to natural language. Schank is quite secure at Yale. Weizenbaum also has tenure. However, some assistant professorships in linguistics may be at stake, especially at M.I.T.

Allen Newell and Herbert Simon are criticized for being overoptimistic and are considered morally defective for attempting to describe humans as difference-reducing machines. Simon's view that the human is a simple system in a complex environment is singled out for attack. In my opinion, they were overoptimistic, because their GPS model on which they put their bets wasn't good enough. Maybe Newell's current *production system models* will work out better. As to whether human mental structure will eventually turn out to be simple, I vacillate but incline to the view that it will turn out to be one of the most complex biological phenomena.

I regard Forrester's models as incapable of taking into account qualitative changes, and the world models they have built as defective even in their own terms, because they leave out saturation-of-demand effects that cannot be discovered by curve-fitting as long as a system is rate-of-expansion limited. Moreover, I don't accept his claim that his models are better suited than the unaided mind in "interpreting how social systems behave," but Weizenbaum's sarcasm on page 246 is unconvincing. He quotes Forrester, "[desirable modes of behavior of the social system] *seem to be possible only if we have a good understanding of the system dynamics and are willing to endure the self-discipline and pressures that must accompany the desirable mode."* Weizenbaum comments, *"There is undoubtedly some interpretation of the words 'system' and 'dynamics' which would lend a benign meaning to this observation."* Sorry, but it looks ok to me provided one is suitably critical of Forrester's proposed social goals and the possibility of making the necessary assumptions and putting them into his models.

Skinner's behaviorism that refuses to assign reality to people's internal state seems wrong to me, but we can't call him immoral for trying to convince us of what he thinks is true.

Weizenbaum quotes Edward Fredkin, former director of Project MAC, and the late Warren McCulloch of M.I.T. without giving their names. pp. 241 and 240. Perhaps he thinks a few puzzles will make the book more interesting, and this is so. Fredkin's plea for research in automatic programming seems to overestimate the extent to which our society currently relies on computers for decisions. It also overestimates the ability of the faculty of a particular university to control the uses to which technology will be put, and it underestimates the difficulty of making knowledge based systems of practical use. Weizenbaum is correct in pointing out that Fredkin doesn't mention the existence of genuine conflicts in society, but only the new left sloganeering elsewhere in the book gives a hint as to what he thinks they are and how he proposes to resolve them.

As for the quotation from (McCulloch 1956), Minsky tells me "this is a brave attempt to find a dignified sense of freedom within the psychological determinism morass." Probably this can be done better now, but Weizenbaum wrongly implies that McCulloch's 1956 effort is to his moral discredit.

Finally, Weizenbaum attributes to me two statements — both from oral presentations — which I cannot verify. One of them is *"The only reason we have not yet succeeded in simulating every aspect of the real world is that we have been lacking a sufficiently powerful logical calculus. I am working on that problem."* This statement doesn't express my present opinion or my opinion in 1973 when I am alleged to have expressed it in a debate, and no-one has been able to find it in the video-tape of the debate.

We can't simulate "every aspect of the real world," because the initial state information is never available, the laws of motion are imperfectly known, and the calculations for a simulation are too extensive. Moreover, simulation wouldn't necessarily answer our questions. Instead, we must find out how to represent in the memory of a computer the information about the real world that is actually available to a machine or organism with given sensory capability, and also how to represent a means of drawing those useful conclusions about the effects of courses of action that can be correctly inferred from the attainable information. Having a *sufficiently powerful logical calculus* is an important part of this problem — but one of the easier parts.

The second quotation from me is the rhetorical question, *"What do judges know that we cannot tell a computer."* I'll stand on that if we make it "eventually tell" and especially if we require that it be something that one human can reliably teach another.

A SUMMARY OF POLEMICAL SINS

The speculative sections of the book contain numerous dubious little theories, such as this one about the dehumanizing effect of the invention of the clock: *"The clock had created literally a new reality; and that is what I meant when I said earlier that the trick man turned that prepared the scent for the rise of modern science was nothing less than the transformation of nature and of his perception of reality. It is important to realize that this newly created reality was and remains an improverished version of the older one, for it rests on a rejection of those direct experiences that formed the basis for, and indeed constituted the old reality. The feeling of hunger was rejected as a stimulus for eating; instead one ate when an abstract model had achieved a certain state, i.e. when the hand of a clock pointed to certain marks on the clock's face (the anthropomorphism here is highly significant too), and similarly for signals for sleep and rising, and so on."*

However, when home terminals become available, social changes of the magnitude of those produced by the telephone and automobile will occur.

This idealization of primitive life is simply thoughtless. Like modern man, primitive man ate when the food was ready, and primitive man probably had to start preparing it even further in advance. Like modern man, primitive man lived in families whose members are no more likely to become hungry all at once than are the members of a present family.

I get the feeling that in toppling this microtheory I am not playing the game; the theory is intended only to provide an atmosphere, and like the reader of a novel, I am supposed to suspend disbelief. But the contention that science has driven us from a psychological Garden of Eden depends heavily on such word pictures.

By the way, I recall from my last sabbatical at M.I.T. that the *feeling of hunger* is more often the *direct social stimulus for eating* for the "hackers" deplored in Chapter 4 than it could have been for primitive man. Often on a crisp New England night, even as the clock strikes three, I hear them call to one another, messages flash on the screens, a flock of hackers magically gathers, and the whole picturesque assembly rushes chattering off to Chinatown.

I find the book substandard as a piece of polemical writing in the following respects:

1. The author has failed to work out his own positions on the issues he discusses. Making an extreme statement in one place and a contradictory statement in another is no substitute for trying to take all the factors into account and reach a considered position. Unsuspicious readers can come away with a great variety of views, and the book can be used to support contradictory positions.

2. The computer linguists — Winograd, Schank, et. al. —— are denigrated as hackers and compulsive computer programmers by innuendo.

3. One would like to know more precisely what biological and psychological experiments and computer applications he finds acceptable. Reviewers have already drawn a variety of conclusions on this point.

4. The terms "authentic," "obscene," and "dehumanization" are used as clubs. This is what mathematicians call "proof by intimidation."

5. The book encourages a snobbery that has no need to argue for its point of view but merely utters code words, on hearing which the audience is supposed to applaud or hiss as the case may be. The *New Scientist* reviewer and Daniel McCracken in *Datamation* certainly salivate in most of the intended places.

6. Finally, when moralizing is both vehement and vague, it invites authoritarian abuse either by existing authority or by new political movements. Imagine, if you can, that this book were the bible of some bureaucracy, e.g. an Office of Technology Assessment, that acquired power over the computing or scientific activities of a university, state, or country. Suppose Weizenbaum's slogans were combined with *the bureaucratic ethic* that holds that any problem can be solved by a law forbidding something and a bureaucracy of eager young lawyers to enforce it. Postulate further a vague *Humane Research Act* and a "public interest" organization with more eager young lawyers suing to get judges to legislate new interpretations of the Act. One can see a laboratory needing more lawyers than scientists and a Humane Research Administrator capable of forbidding or requiring almost anything.

I see no evidence that Weizenbaum forsees his work being used in this way; he doesn't use the phrase *laissez innover* which is the would-be science bureaucrat's analogue of the economist's *laissez faire,* and he never uses the indefinite phrase "it should be decided" which is a common expression of the bureaucratic ethic. However, he has certainly given his fellow computer scientists at least some reason to worry about potential tyranny.

Let me conclude this section with a quotation from Andrew D. White, the first president of Cornell University, that seems applicable to the present situation — not only in computer science, but also in biology. — *"In all modern history, interference with science in the supposed interest of religion, no matter how conscientious such interference may have been, has resulted in the direst evils both to religion and to science, and invariably; and, on the other hand, all untrammelled scientific investigation, no matter how dangerous to religion some of its stages may have seemed for the time to be, has invariably resulted in the highest good both of religion and of science."* Substitute *morality* for *religion* and the parallel is clear. Frankly, the feebleness of the reaction to attacks on scientific freedom worries me more than the strength of the attacks.

WHAT WORRIES ABOUT COMPUTERS ARE WARRANTED?

Grumbling about Weizenbaum's mistakes and moralizing is not enough. Genuine worries prompted the book, and many people share them. Here are the genuine concerns that I can identify and the opinions of one computer scientist about their resolution: What is the danger that the computer will lead to a false model of man? What is the danger that computers will be misused? Can human-level artificial intelligence be achieved? What, if any, motivational characteristics will it have? Would the achievement of artificial intelligence be good or bad for humanity?

1. Does the computer model lead to a false model of man? Historically, the mechanistic model of the life and the world followed animistic models in accordance with which, priests and medicine men tried to correct malfunctions of the environment and man by inducing spirits to behave better. Replacing them by mechanistic models replaced shamanism by medicine. Roszak explicity would like to bring these models back, because he finds them more "human," but he ignores the sad fact that they don't work, because the world isn't constructed that way. The pre-computer mechanistic models of the mind were, in my opinion, unsuccessful, but I think the psychologists pursuing computational models of mental processes may eventually develop a really beneficial psychiatry.

Philosophical and moral thinking hasn't yet found a model of man that relates human beliefs and purposes to the physical world in a plausible way. Some of the unsuccessful attempts have been more mechanistic than others. Both mechanistic and non-mechanistic models have led to great harm when made the basis of political ideology, because they have allowed tortuous reasoning to justify actions that simple human intuition regards as immoral. In my opinion, the relation between beliefs, purposes and wants to the physical world is a complicated but ultimately solvable problem. Computer models can help solve it, and can provide criteria that will enable us to reject false solutions. The latter is more important for now, and computer models are already hastening the decay of dialectical materialism in the Soviet Union.

2. What is the danger that computers will be misused? Up to now, computers have been just another labor-saving technology. I don't agree with Weizenbaum's acceptance of the claim that our society would have been inundated by paper work without computers. Without computers, people would work a little harder and get a little less for their work. However, when home terminals become available, social changes of the magnitude of those produced by the telephone and automobile will occur. I have discussed them elsewhere, and I think they will be good — as were the changes produced by the automobile and the telephone. Tyranny comes from control of the police coupled with a tyrannical ideology; data banks will be a minor convenience. No dictatorship yet has been overthrown for lack of a data bank.

One's estimate of whether technology will work out well in the future is correlated with one's view of how it worked out in the past. I think it has worked out well — e.g. cars were not a mistake — and am optimistic about the future. I feel that much current ideology is a combination of older anti-scientific and anti-technological views with new developments in the political technology of instigating and manipulating fears and guilt feelings.

3. What motivations will artificial intelligence have? It will have what motivations we choose to give it. Those who finally create it should start by motivating it only to answer questions and should have the sense to ask for full pictures of the consequences of alternate actions rather than simply how to achieve a fixed goal, ignoring possible side-effects. Giving it human motivational structure with its shifting goals sensitive to

The achievement of above-human-level artificial intelligence will open to humanity an incredible variety of options.

physical state would require a deliberate effort beyond that required to make it behave intelligently.

4. Will artificial intelligence be good or bad? Here we are talking about machines with the same range of intellectual abilities as are possessed by humans. However, the science fiction vision of robots with almost precisely the ability of a human is quite unlikely, because the next generation of computers or even hooking computers together would produce an intelligence that might be qualitatively like that of a human, but thousands of times faster. What would it be like to be able to put a hundred years thought into every decision? I think it is impossible to say whether qualitatively better answer would be obtained; we will have to try it and see.

The achievement of above-human-level artificial intelligence will open to humanity an incredible variety of options. We cannot now fully envisage what these options will be, but it seems apparent that one of the first uses of high-level artificial intelligence will be to determine the consequences of alternate policies governing its use. I think the most likely variant is that man will use artificial intelligence to transform himself, but once its properties and the consequences of its use are known, we may decide not to use it. Science would then be a sport like mountain climbing; the point would be to discover the facts about the world using some stylized limited means. I wouldn't like that, but once man is confronted by the actuality of full AI, they may find our opinion as relevant to them as we would find the opinion of *Pithecanthropus* about whether subsequent evolution took the right course.

5. What shouldn't computers be programmed to do. Obviously one shouldn't program computers to do things that shouldn't be done. Moreover, we shouldn't use programs to mislead ourselves or other people. Apart from that, I find none of Weizenbaum's examples convincing. However, I doubt the advisability of making robots with human-like motivational and emotional structures that might have rights and duties independently of humans. Moreover, I think it might be dangerous to make a machine that evolved intelligence by responding to a program of rewards and punishments unless its trainers understand the intellectual and motivational structure being evolved.

All these questions merit and have received more extensive discussion, but I think the only rational policy now is to expect the people confronted by the problem to understand their best interests better than we now can. Even if full AI were to arrive next year, this would be right. Correct decisions will require an intense effort that cannot be mobilized to consider an eventuality that is still remote. Imagine asking the presidential candidates to debate on TV what each of them would do about each of the forms that full AI might take.

Reference:

McCulloch, W.S. (1956) Toward some circuitry of ethical robots or an observational science of the genesis of social evaluation in the mind-like behavior of artifacts. *Acta Biotheoretica,* **XI,** *parts 3/4, 147-156*

John McCarthy
Stanford, California

He worries that the power of the computer reinforces the widespread view that science is all powerful and that all unscientific ways of knowing are, therefore, suspect, fuzzy, vague, and not worthy of our trust.

The Computer's Role vs Man's Role

This book is about computers and their impact on human beings, but don't let that put you off. This is not another one of those dreary, guided tours through computerland or one of those windy sermons on the social significance of the (Gasp!) computer. This is an important book about an important subject. Why, Weizenbaum asks in this book, are we so ready to take the computer so seriously as a complete model for the human mind? And what, if any, are the dangers in doing so? Whether or not one agrees with his answers, they are important answers, well presented.

Weizenbaum tells us that he wrote this book as a result of the reactions he observed to the work for which he probably is best known, namely a computer program called ELIZA and her possibly even better known "son" the DOCTOR. Both of these programs (the latter being the former provided with a script that many find amusing) are based on some rather clever ideas. One of the most important of these is one that we have probably all used—the idea that you can keep your side of a conversation going without listening to the other by either repeating some of the other person's words or by occasionally muttering appropriate trivialities. The DOCTOR program uses this trick to fake what we might call a Rogerian psychiatrist who then encourages responses from patients by using something akin to (but very different from) this conversationalist trick. What surprised Weizenbaum, and led him to write this book, was that people were so eager to take this rather trivial program seriously. Some psychiatrists (e.g. Colby) took it to be a model, not only of the psychiatrist, but also of his patient. (In this connection, you might be interested in the rather amusing response of Weizenbaum's to Colby's simulation or paranoia in a letter in a recent issue of "The Communications of the ACM.") Some people interacting with the program rapidly began to take it seriously and the program even passed a version of Turing's test when a majority of the subjects in an experiment by Quarton and others said that, when conversing with Weizenbaum's program, they thought that they had been dealing with a real person. And other people began to think (as many had, years earlier, after IBM demonstrated "machine translation" at the 1939 New York World's Fair) that the problem of getting machines to handle natural languages had, to all intents and purposes, been solved.

These responses led Weizenbaum to ask why people were so ready to accept this very simple model of their own (and their friends') minds as complete. This book is Weizenbaum's attempt to answer this question.

Weizenbaum recognizes that people's tendency to think of themselves as machines of one sort or another is nothing very new nor, necessarily, anything very bad. We understand machines that we build better than the minds that we have and the similarities between our minds and machines can help us to better understand the former. What worries Weizenbaum is not that we use the computer as a model for ourselves but that we may be taking it too seriously in this role.

The introduction of the computing machine, with its marvelous ability to manipulate symbols rapidly and accurately has simply accelerated man's tendency to take a "rationalistic view of his society and a mechanistic view of himself." Weizenbaum suggests that the question of "whether or not human thought is entirely computable" is merely a sharpening up of a question that has attracted attention for millenia—namely whether or not every aspect of human thought is reducible to some sort of precise formalism. Weizenbaum then addresses two basic questions: "Can human thought be reduced to calculation (or computation)?" and "Even if it can be, ought it to be?" He worries that the power of the computer reinforces the widespread view that science is all powerful and that all unscientific ways of knowing are, therefore, suspect, fuzzy, vague, and not worthy of our trust.

Weizenbaum begins by considering how people tend to think of their tools as a metaphor for their world. The view that man is some kind of machine predates the invention of the computer. But the impact of the computer as methaphor for the mind has

Weizenbaum points out that, contrary to what many have claimed, to be able to program something for a computer is not the same thing as to be able to understand it.

been much greater than that of other machines, and Weizenbaum goes on to explain why. The computer is, in a sense that he describes in two very well written chapters, a universal symbol processing machine. Although Weizenbaum tells us we can read this book without reading these two chapters on "Where the Power of the Computer Comes From" and "How Computers Work," I found them utterly delightful.

These two chapters demonstrate that Weizenbaum's argument comes from somebody who understands the computer. Weizenbaum explains Turing's Universal Machine Theorem in terms that should be accessible to many readers who shy away from anything mathematical, although some may find the explanation a bit wordy. The Turing machine itself is exemplified in terms of stones and the perfect real-life counterpart of the Turing machine's infinite tape divided into squares—a roll of toilet paper.

He goes on to explain logical design and computer languages in a style that should appeal to many people with a shaky understanding of these areas and should be enjoyable to those with solid backgrounds who like to see things they know said well. Some of the remarks he makes are marvelously apt. Thus in explaining the power of languages, he refers to Maslow's comment that, "to the man who has only a hammer, the whole world looks like a nail" to suggest how one's programming language can shape one's view of the world. Weizenbaum concedes that the computer is impressive and it is a suitable model for some thinking but it is not a total model.

Weizenbaum points out that, contrary to what many have claimed, to be able to program something for a computer is not the same thing as to be able to understand it. It helps us to understand something better when we write a program to do it, but it also helps us when we try to describe it in English. Programming something is one way of understanding it, and a good way. What Weizenbaum objects to is thinking that it is the only way.

The computer program, running under the control of a stored program, is an abstraction from the real world (with all its messy details) that many find more appealing than the real thing. It is so comfortable and neat that it can give rise to what Weizenbaum calls the "compulsive programmer" who sits, bleary eyed, in front of his console, totally absorbed by it and the technical problems that it presents to him or her. "The compulsive programmer," writes Weizenbaum, "is merely the proverbial mad scientist who has been given a theatre, the computer, in which he can, and does, play out his fantasies."

There is, according to Weizenbaum, a continuum, ranging from the compulsive programmer at one hand, who seeks to fit the world and its people into his (or her) rather narrow perceptions of what is real, to the fuzzy minded humanist at the other extreme who seeks to understand the human being from all possible perspectives and finds no framework adequate to that purpose. I personally find both extremes somewhat distressing, but Weizenbaum focusses his guns on the compulsive programmer.

Science proceeds by simplifying reality, but Weizenbaum worries that we can overdo such simplification. He compares the computer scientist, who is trying to account for all human thought in terms of the computer, to the drunk who is looking for his keys under the street light, not because that is where he dropped them, but because the light is better there. Scientists do, indeed, tend to look where the light is better. That may be a mistake if they also assume that they are looking in the only possible place, but it is surely not a mistake per se. It is a mistake to think that "man is merely an information processing machine." This limiting view is like a magical system in that it is detached from some (but not necessarily all) human experience.

It is seductive precisely because it gives the person who holds it the illusion of having power that he does not really have. It is bad because it tries to capture all the world in a framework that is too weak to hold it.

There are some things we know only by virtue of having bodies, being children, and being treated as human beings by other human beings.

Weizenbaum argues that there is "nothing wrong with viewing man as an information processor (or indeed as anything else), nor with attempting to understand him from that perspective, providing, however, that we never act as though any single perspective can comprehend the whole man. Our theories are dangerous if we think of them as being like encyclopedias in which we can look up what we want to know rather than as maps or guides that help us to understand but are not total guides to the territory they comprehend. The computer is a powerful instrument in helping us understand our world which is precisely why it is so important to remember that it is not the only instrument we have."

Weizenbaum considers various fields that use the computer metaphor to understand man. He devotes chapters to "Computer Models in Psychology," "The Computer and Natural Language" and to "Artificial Intelligence." The first of these focuses primarily on the ideas of Simon and Colby, the second on ELIZA and the work of Schank. Like other writers on this general subject, Weizenbaum quotes extensively from the rather optimistic writings of Simon whose rather rambunctious claims, having been given with explicit time limits, have proven to be overly optimistic and now provide fine fodder for critics. Weizenbaum is arguing that there is more in heaven and earth than is dreamt of in our computational philosophies and in reminding us of this, he is surely right.

He also points out that there are things that computers cannot ever know about human experience. There are some things we know only by virtue of having bodies, being children, and being treated as human beings by other human beings. And there are things one knows that one may not be able to communicate at all and that one can only understand by having certain experiences as a human being.

These objections, Weizenbaum writes, touch not only on certain technical limitations of computers, but on what it means to be a human being and what it means to be a computer. He examines the arguments of those who find that all human experience must be expressible by computer programs and finds them wanting. His analysis is perceptive (he understands the workers in this field) and probably correct. Interestingly enough, he recognizes that perhaps the best way to communicate even this idea may be in terms of a mechanistic metaphor. Thus when he talks of the contributions of the unconscious he points to the right hemisphere of the human brain (in those of us who are right-handed) where those contributions appear to "come from." It is a bit odd to see him using mechanistic metaphors against the mechanists.

He concedes that computers can make judicial decisions (and) psychiatric judgements but argues that they ought not to make them because they lack the proper background for this role. Even if computers gave the same results as a person in some (or even most) cases, we have no guarantee that they will do so in all. "What emerges as the most elementary insight," he writes, "is that, since we do not have any way of making computers wise, we ought not now to give computers tasks that demand wisdom."

Weizenbaum points to the dangers of what he calls "incomprehensible programs" whose operations are not fully understood. Such programs arise because single programmers do not correctly anticipate the behavior of their programs and because programmers do not program singly.

Incomprehensible programs, and most large programs are incomprehensible in Weizenbaum's sense, cannot be relied on to do what we may think they do and their controlled use can, therefore, be dangerous. This is particularly true in those cases where the decisions made by the program have to be acted on before we have the chance to validate their soundness.

There are, then, at least two reasons not to rely wholly on programs to make human decisions. One is that we know things that programs cannot know and the second is that there are some things that, in a sense, only the program can know and we cannot.

Weizenbaum is unsparing in pointing out that the triumphs of artificial intelligence, to date, are largely "triumphs of technique" and that even his own programs, such as ELIZA, fall in this category. He notes how few results of artificial intelligence research have found their way into industry. Those few have (like DENDRAL and MACSYMA) come from areas in which a precisely and well understood theory had already been developed. But most programs in AI are not theory-based in the same way. This is why they lead to programs whose behavior it is not necessarily possible to understand.

Our tendency to place too much faith in computer programs is not limited to the artificial intelligentsia. It is found frequently among the military and among those in government. There is a strong desire in bureaucracies of all kinds to harness the inexorable information crunching power of the computer to the bureaucracy's aims and to rely on computers because of their "objectivity." Such a reliance, Weizenbaum finds "mindless" in at least two senses of this word.

Weizenbaum observes that it is a curious fact that the use of a word like "ethics" in conversation about science makes us feel uncomfortable. There are, he suggests, at least two reasons for this, neither of which is totally misplaced. One reason is that we fear that what is about to be said may apply to us and the second is that the conversation will be philosophical in the worst sense, which is to say that it will be vague and tedious. Weizenbaum's critique does apply to most of us in computer science and, therefore, reading him may make you feel uncomfortable. But what Weizenbaum says is, by and large at least, not vague or tedious. It is written in a language, and from a point of view, that a computer scientist can understand.

If I have an important objection to what Weizenbaum is doing in this book (which is not to say that I don't also have some unimportant objections) it is that he is not always doing what he claims to be doing. He argues that his aim is not to convert others to his views but to get people to think about what they themselves feel. But toward the end of his book, Weizenbaum gives in to a natural inclination to try to tell us what to think. He prefaces these suggestions with strong warnings against reading them in this way and says that he is only expressing his own views. But these warnings remind me of a bit of the warnings on cigarette advertisements. They are there, but the rest of the message tells you to ignore them. Thus Weizenbaum says that he thinks that work on speech understanding programs is dangerous and that one should not work on such programs. His arguments seem to me to be pretty weak. He also tells us not to couple animal brains with computers because that would be obscene. I may be coarse, and insensitive, but the obscenity eludes me.

Weizenbaum qualifies his injunctions against such research by saying that such judgements "have no force except on myself." He is right, but why does he give us arguments to convince us to agree with them? "I have learned," he writes, "that people are constantly asking one another what they must do, whereas the only really important question is what they must be." But then he tells us what we must do. I wish he hadn't.

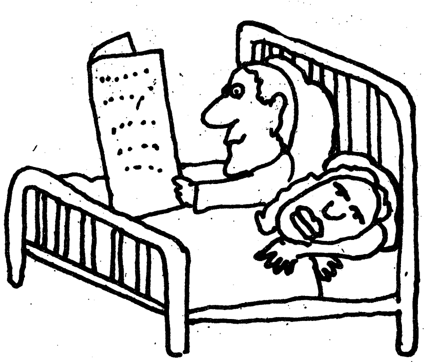
But such injunctions are a minor part of the book and they are easily skimmed. On the whole, Weizenbaum's analysis is very perceptive and very well done. The first three chapters of this book could, I think, be profitably read by students who want to "understand computers." People interested in the impact of the computer on society could profit from reading Weizenbaum's analysis of why the computer is potentially so dangerous, why we are so prone to think of ourselves as being nothing more than flesh and blood computers and why, as Thoreau put it more than a hundred years ago, "men have become the tools of their tools."

But in fact, this book is really primarily addressed to computer scientists and only computer scientists can, I think, really understand it. Weizenbaum admits this. He writes: "If this book is to be seen as advocating anything, then let it be a call to a simple kind of courage. And because this book is, after all, about computers, let that call be heard mainly by teachers of computer science. I want them to have heard me affirm that the computer is a powerful new metaphor for helping us to understand many aspects of the world, but that it enslaves the mind that has no other metaphors and few other resources to call on. The world is many things and no single framework is large enough to contain them all, neither that of man's science nor that of his poetry, neither that of calculating reason nor that of pure intuition."

If you are not familiar with computers, there will be a lot in this book that you may not understand. Be that as it may, if you have at least written a program or two, I think you will find a lot in this book worth thinking, and talking, about. In any case, it is the first book on this subject that I have recommended to my friends. Read it.

Peter Kugel
Boston College, MA

CREATIVE COMPUTING Reviews



Computers and Creativity by Carole Spearin McCauley. 160 pp. $7.50. Praeger Publishers, Inc., 111 Fourth Ave., New York, N.Y. 10003. 1974.

Computers and Creativity asks whether the computer does or does not have artistic ability. It talks about what can be done with a computer other than calculating or processing data in "uncreative ways." The author spends some time trying to explain creativity and also the ways in which some computers learn.

The style of this book did not particularly appeal to me and some details seemed rather tedious. The content, however, was excellent: the kinds of computer applications that this book describes deserve recognition as a growing field in computer science.

Computers and Creativity does not really give enough details about the programs it describes to tell the average programmer how to go about doing likewise. But it would be useful for the person who wants a general introduction to the use of computers as an artistic tool. It deals, in a cursory way, with such questions as "What is creativity?", "What is a computer?", "How does one talk to computers?", "Can a computer sing?", "Can a computer create a picture, a movie, or an advertisement?", "Can a computer write?", and the required chapter in books like this: "Gasp! What will the computer do next?". The book contains a useful, if partial, 8-page list of references.

Peter Rubin
Chestnut Hill, MA

Computer Lib / Dream Machines by Theodor H. Nelson, 128 pp. paper (oversize 11 x 14), $7.00. Hugo's Book Service, 1974. Revised edition 1976.

This is a very unusual book. Comments from university computer people I talked with have ranged from "the best book I've ever seen," which may be true, to "a book of gossip," which is certainly true. It resembles a *Whole Earth Catalog* in its general layout and teensy-weensy type. It is intended to give the total novice, as well as the more knowledgeable person, an idea of why computers do what they do as well as how they do it. Also, the matching T-shirts are now out and the movie is supposed to be on its way.

The front half of the book, *Computer Lib*, tells you in no particular order about where computer work is being done, who is doing it, how computer hardware works, how different types of computer hardware and software differ, how computer languages work, how computers are bought and sold, how some other computer-like things work, and what computers do and don't do, and several other things. All this is embedded in a mish-mosh of pictures, jokes, comments and other interesting goodies. The book lends itself to being dipped into for the interesting parts rather than to being read straight through. Three computer languages are introduced in a fairly good way: two familiar ones (Basic and APL) and one obscure one (Trac). These introductions are meant to give the reader an idea of what programming is like, not to give details, so if you were intending to get this book to learn Basic, don't.

The back half, *Dream Machines*, tells about the author's special interest, graphics. He goes into that the same way he attacked computers in general in the first half, and since this is a smaller field it gets more detailed. It also gets more complex. Since he has definite opinions on graphics, much more than on computers in general, the going can get kind of tough as he gives gruesome details about his favorite things.

He wraps up both halves of the book (in the middle, naturally) with his ideas of what computers, graphics, and people should be doing together, which are interesting in themselves.

Overall, the book has its strong and weak points. The author is no technical computer whiz, which is an advantage since he doesn't assume you are one either. On the other hand, the book has a wealth of trivial errors—the one comment attributed to me is something I never said. Nevertheless, this book is worth getting because you certainly won't find anything like it anywhere else. It's unique.

John Levine
New Haven, CT

*****

The Compleat Computer by Dennie L. Van Tassel, 216 pp. paper, $5.95. Science Research Associates, 1976.

The cover proclaims: "*The Compleat Computer* being a compendium of: Tales of the amazing & marvelous — Poetry — Informative news items — Articles for edification and enjoyment — Cartoons — Plus many other illustrations — with a special section of Splendiferous Science Fiction Art in full color." This in 7 different type faces and two colors, introducing, in my mind, one of the most diverse, interesting and mind-expanding computer books ever to hit print.

I boarded the American flight from Newark to Dallas in the early evening expecting to sleep most of the way, having been up much of the night before with a pregnant cat and cranky son. Most books put me to sleep and I made the mistake of assuming that this one would too. It didn't—Alistair MacLean, Isaac Asimov and John MacDonald move over. Dennie Van Tassel has asembled a fantastic bunch of articles and stories about the computer that would keep Rip Van Winkle alert and wide-eyed.

Setting the stage are a series of articles by what I call "popular" writers titled, by Dennie, "In the Beginning." He then follows with 14 hardware and 7 software pieces. I use the terms hardware and software loosely because Dennie has articles by Michael ("Terminal Man") Crichton, Ray ("The Martian Cronicles") Bradbury, Arthur C. ("2001") Clarke, and believe it or not, Art ("Ha!") Buchwald. C'mon now Buchwald, what do you know about that data processing stuff?

Stewart Brand drops in to talk about "Counter Computers" in the section, "The Present and Potential." Stewart, you'll recall, loyal *Creative* readers, does *CoEvolution Quarterly* and *Whole Earth Catalog* and *Epilogue* and that stuff.

Sections follow on "Applications," "The Impact," and "Governmental Uses" (ugh—not to the writers, but to Uncle—IRS, HEW, CIA, NCIC, GTH—Sam).

"Controls, or Maybe Lack of Controls" comes next with all the (usual) privacy stuff and more including an article "Man Bites Ford" from *Consumer Reports* (watch it, Dennie, you may have permission, but *CR* doesn't permit reprints even when they've given permission—they're suing us now—see you in jail).

Rounding out the book are some keen (40's), super (50's), fantastic (60's), bad (70's) articles on "The Future" (80's). (With one exception—but that's up to you to find).

My review may be on the light side but this is not a book to be taken lightly. It is a book assembled with great care, and a book that can (and should) well serve as the backbone of a computer literacy or computers in society course. Indeed, it should be required reading of every high school and college graduate today. After all, like it or not, the computer is now our constant companion, slave and nemesis, and one really ought to know about this fantastic force/animal/machine/intelligence — (select one or more).

Get a copy. Today!

David H. Ahl
Morristown, NJ

*****

The Elementary Functions: An Algorithmic Approach. G. Albert Higgins, Jr. 335 pp. Prentice-Hall, Inc. Englewood Cliffs, New Jersey 07632.

This book uses the idea of a computer program, and of the algorithm that it carries out, to develop the student's understanding of the idea of a function in general and of many specific elementary functions. It assumes that the reader has some ability to program computers (preferably, but not necessarily, in BASIC), but the reader will need few mathematical ideas beyond those found in elementary algebra and a smattering of elementary set theory. The idea of a function is developed in terms of algorithms for evaluating functions and doing other things to them.

Many of the ideas used are what one might call "geometric" or visually oriented. Some ideas from calculus are introduced including the idea of the area under a curve and of the slope of a curve at a point. But they are introduced in algorithmic clothes (e.g. by using the Trapezoidal Rule to actually compute areas).

The book contains many problems that use the results developed. Numerical answers and graphical interpretations are often given. By providing both algorithmic and geometrical interpretations of ideas that are usually portrayed only algebraically, the author manages to enrich a subject that may often appear rather dry. The considerable amount of historical material presented doesn't hurt in this respect either. But this is a fairly abstract treatment and it is not recommended for use in classes in which students are likely to ask "What is this good for?"

This book is suitable for a post-algebra, post-computing but pre-calculus course in either high school or college. It provides, as the author claims, a reasonable foundation for further work, not only in calculus but also in statistics and-or probability. It is well written.

John Cordeiro
Chestnut Hill, MA.

*****

CREATIVE COMPUTING
Feature Review

Herbert Dreyfus reviews two books by Marvin Minsky, Seymour Papert, and Patrick Winston.

Artificial Intelligence, Marvin Minsky and Seymour Papert, 61 pp., $2.00, Condon Lectures, Oregon State System of Higher Education, Eugene, Oregon, 97403, 1974.[1]

The Psychology of Computer Vision, Ed. Patrick Henry Winston, 280 pp., $19.50, McGraw-Hill, 1221 Avenue of the Americas, New York, N. Y., 10020, 1975.

These two books reveal that an important change has taken place at the MIT AI laboratory during the past five years. In previous works, e.g. *Semantic Information Processing* (1968)[2] Minsky and his co-workers sharply distinguished themselves from workers in Cognitive Simulation who presented their programs as psychological theories, insisting that the MIT programs were "an attempt to build intelligent machines without any prejudice toward making the system...humanoid."[3] Now the prefact to *Artificial Intelligence* states, "the primary use of computers for research into the nature of intelligence is that of simulation,"[4] and Minsky and Papert attempt to argue for the role of symbolic representations in intelligent behavior by a constant polemic against behaviorism and gestalt psychology. Likewise Winston claims, in support of his collection of papers on computer vision, that "Making machines see is an important way to understand how we animals see."[5]

"Computer science has brought a flood of...ideas, well defined and experimentally implemented, for thinking about thinking..."

Underlying this change one can detect the effect of ten years of growing success in the manipulation of *ad hoc* micro-worlds, accompanied by a decade of failure to produce a single system which even begins to approach the adaptability to changing contexts shown by a dog, a cat, or a six months old child. Instead of concluding from this frustrating situation that the machine techniques which work in context-free, game-like, micro-worlds may in no way resemble human and animal intelligence, the AI workers have taken the less embarrassing tack of suggesting that even if they cannot succeed in building intelligent systems, the techniques so successful in micro-world analysis can be justified as a valuable contribution to psychology.

Such a pitch, however, since it involves a stronger claim than the old slogan "that as long as the machine was intelligent it did not matter at all whether it performed in a humanoid way," runs the obvious risk of refutation by empirical evidence. The risk is especially great at this moment when recent work in Cognitive Psychology on the role of rotation of images in pattern recognition (Shepard)[6] and the discovery of the use of stereotypical images in categorization (Rosch)[7], has led many cognitive psychologists to reevaluate the explanatory power of a formal model of perception and cognition.

Compelled by the formal nature of any computer model, however, Minsky and Papert take an unquestioning stand in favor of abstract, symbolic representations, and against concrete, physical, perceptual processes. Thus making a virtue of necessity, they revive the intellectualist position of Kant's *Critique of Pure Reason,* according to which perception is indistinguishable from rule governed thought.

> The Gestaltists look for simple and fundamental principles about how perception is organized, and then attempt to show how symbolic reasoning can be seen as following the same principles, while we construct a complex theory of how knowledge is applied to solve intellectual problems and then attempt to show how the symbolic description that is what one "sees" is constructed according to similar such processes.[8]

But this attempt to invert the prima facae priority of perception to thinking gets Minsky and Papert into the same sort of trouble that eventually led Kant, in the *Critique of Judgement,* to give up this view. Before one can begin to select primatives in terms of which to analyze a scene, the scene must be segregated into local units and salient features. Minsky recognizes this as the gestaltists' argument for the priority of the figure-ground distinction but, on the basis of Guzman's success in the analysis of scenes involving rectilinear objects, he retorts that:

> In complex scenes, the features belonging to different objects have to be correctly segregated to be meaningful; but solving this problem—which is equivalent to the traditional Gestalt "figure-ground" problem—presupposes solutions for so many visual problems that the possibility and perhaps even the desirability of a separate recognition technique falls into question[9].

This, however, presupposes that the top-down technique of looking for edges, which works in segmenting rectilinear objects, can somehow be generalized to curved surfaces. In absence of any such techniques, the question remains how to account for the organization of the primative wholes which form the basis of higher-order recognition processes.

Recently, extension of early gestalt work on the perception of similarity of simple perceptual figures,—arising in part in response to "the frustrating efforts to teach pattern recognition to [computers]"[10]—has revealed sophisticated distinctions between figure and ground, matter and form, essential and accidental aspects, norms and distortions, etc. which are already apparent at the perceptual level even when no recognizable objects are present. Careful, empirical studies of perceptual similarity by Erich Goldmeier have demonstrated a "kind of relation between stimulus variation and phenomenal variation [which] has never been envisioned in psychological theory."[11] He has been led to conclude that these perceptual functions cannot be accounted for in terms of the rule-like relations of formal features of the stimuli, except perhaps on the neurological level, where the importance of Pragnanz or singularity suggests physical phenomena such as "regions of resonance."[12]

Minsky is aware that there are theorists who claim that the organization of perception can only be explained in terms of physical processes such as resonance and holograms, but he rejects this view with the remark that:

> The output of a quantitative mechanism, be it numerical, statistical, analog, or physical (nonsymbolic), is too structureless and uninformative to permit further analysis.[13]

But this thrice begs the question. First since it is not obvious that perception is thinking, or even that all thinking is analysis, it is not obvious that the stable patterns of perceptual organization need provide the sort of features required in higher order computation. Secondly, even if higher order objects *are* recognized in terms of features, we have just seen that concrete perceptual organization, far from being unstructured, provides the necessary structure for higher operations. Thirdly, it cuts no ice against a neurological (nonsymbolic) view that it does not permit further analysis, if this means it cannot be explained in terms of a computer program. What the gestaltists precisely question is whether perception is the sort of phenomenon amenable to formal, symbolic analysis.

Of course, it is still possible that the gestaltists went too far in trying to assimilate thought to the same sort of concrete, holistic, brain processes they found necessary to account for perception. Thus, even though the exponents of symbolic representation have no account of perceptual processes, they might be right about the mechanism of everyday thinking and learning. Such a formal model of everyday learning and recognition is proposed by Winston in his paper, "Learning Structural Description from Examples."[14]

Given a set of positive and negative instances, Winston's self proclaimed "classic" program for learning the structural description of an arch uses a small pre-selected and pre-programmed descriptive repertoire to construct a formal description of the class of arches.

But is this a plausible theory of learning? Winston ingeniously concludes that it is:

> Although this may seem like a very special kind of learning, I think the implications are far ranging, because I believe that learning by examples, learning by being told, learning by imitation, learning by reinforcement and other forms are much like one another.
>
> In the literature of learning there is frequently an unstated assumption that these various forms are fundamentally different. But I think the classical boundaries between the various kinds of learning will disappear once superficially different kinds of learning are understood in terms of processes that construct and manipulate descriptions.[15]

But, of course, this program only works if the "student" is saved the trouble of doing what Peirce called abduction, by being "told" a set of context free features and relations—in this case a list of possible spacial relationships of blocks such as contact, support and alignment—from which to build up the description. These features are just the sort of prominences formed in perception by repeated experience. Minsky and Papert in their account of this program don't seem to notice that without this pre-programmed "training" it would make no sense to say that "to eliminate objects which seem atypical...[the] program lists all relationships exhibited by more than half of the candidates in the set."[16] Without perceptual saliences all the objects share an indefinitely large number of relationships.

Is this then perhaps at least a plausible theory of categorization? Once it has been given what Winston disarmingly calls a "good description"[17] and carefully chosen examples, the program indeed concludes that an arch is a structure in which a prismatic body is supported by two upright blocks that do not touch one another. But even Winston admits that having two supports and a top does not begin to capture even the geometrical structure of arches, many of which are curved. So Winston proposes to "generalize the machine's description attributes to acts and properties required by those acts"[18], adding some *ad hoc* predicate like "something to walk through"[19].

But it is not at all clear how the above predicate which refers to implicit knowledge of the bodily skill of "walking through" is to be formalized. Indeed, Winston himself provides a *reductio ad absurdum* of this facile appeal to formal predicates:

> To a human, an arch may be something to walk through, as well as an appropriate alignment of bricks. And certainly, a flat rock serves as a table to a hungry person, although far removed from the image the word *table* usually calls to mind.
>
> But the machine does not yet know anything of walking or eating, so the programs discussed here handle only some of the physical aspects of these human notions. There is no inherent obstacle forbidding the machine to enjoy functional understanding. It is a matter of generalizing the machine's descriptive ability to acts and properties required by those acts. Then chains of pointers can link TABLE to FOOD as well as to the physical image of a table, and the machine will be perfectly happy to draw up its chair to a flat rock with the human, given that there is something on that table which it wishes to eat.[20]

One can't help feeling, when Winston ends his praise of Minsky's "first step" with the challenge: "Much remains to be done," that this is just a tactful way of saying: "Nothing has been accomplished."

Further work on recognition of arches, tables, etc. must, it seems, either wait until we have captured in an abstract symbolic description all that human beings implicitly know simply by having a body, or else until computers no longer have to be told what it is to walk and eat, because they have human bodies and appetites themselves!

In the meantime Winston's proposal cannot be considered a contribution to a theory of learning and recognition until he solves the following fundamental problems:

(1) The program can only learn even a simplified geometrical concept like arch if the programmer, using his everyday understanding, makes explicit and pre-selects a small set of relevant features to "tell" the program. There is no sign how programs could acquire these features.

(2) To distinguish accidental from essential features the program pre-weights its primitives. Once we see how arches function in our everyday activities, there is no reason to suppose that there is *any* set of necessary and sufficient conditions for defining our everyday notion of an arch, and in any case Winston gives us no idea how the program could assign these weights.

(3) The prominent characteristics shared by some everyday arches are "helping to support something while leaving an important open space under it," or "being the sort of thing one can walk under and through at the same time." How does Winston propose to convert such characteristics into the definable, context-free features required by a formal representation?

Despite these seemingly insurmountable obstacles Winston boasts that "there will be no contentment with [concept learning] machines that only do as well as humans."[21] In fact, there has been little progress in machine learning, induction, or concept formation. Indeed, Minsky and Papert admit that "we are still far from knowing how to design a powerful yet subtle and sensitive inductive learning program."[22] What is surprising is that they add "but the schemata developed in Winston's work should take us a substantial part of the way."[23] The lack of progress in the seven years since Winston's work was published, plus the total dependence of the program on a human programmer to provide the primitives from which it can produce its rigid, restricted, and largely irrelevant descriptions, makes it hard to understand in what way the program is a substantial first step.

Moreover, if Winston claims to "shed some light on" the question: "How do we recognize examples of various concepts?"[24] his theory must, like any psychological theory, be subject to empirical test. It so happens that contrary to Winston's claims, recent evidence collected and analyzed by Eleanor Rosch on just this subject has tended to establish that recognition of basic objects such as chairs and tables does not depend on learning which features define the concept, but on seeing them as more or less distant from an imagined paradigm:

> Many experiments have shown that categories appear to be coded in the mind neither by means of lists of each individual member of the category, nor by means of a list of formal criteria necessary and sufficient for category membership, but, rather, in terms of a prototype of a typical category member. The most cognitive economical code for a category is, in fact, a *concrete image* of an average category member.[25]

This research suggests that we had better look to Minsky's more recent proposal for using frames, or prototypes, to represent everyday knowledge for a contribution to the psychology of categorization.[26] But this ambitious proposal which Winston, graciously returning Minsky's compliment, considers "the ancestor of a wave of progress in AI,"[27] begs every fundamental question raised by Rosch's research. Indeed, a passage from Minsky's influential paper can be used to pinpoint many of the unsolved problems in the field:

> There are many forms of chairs, for example, and one should choose carefully the chair-description frames that are to be the major capitols of chair-land. These are used for rapid matching and assigning priorities to the various differences. The lower priority *features* of the *cluster* center then serve... as properties of the chair *types*...
>
> [There is no argument why we should expect to find elementary context free *features* characterizing a chair *type,* nor any suggestion as to what these features might be. They certainly cannot be legs, back, seat, etc. since these aspects of chairs are not context-free features defined apart from chairs, which then clustered in a chair representation.]
>
> Difference could be functional as well as geometric. Thus, after rejecting a first try at "chair" one might try the functional idea of "something one can sit on" to explain an unconventional form.
>
> [A function so defined is not abstractable from human embodied know-how and cultural practices. If it is treated as an additional symbolic description along with physical features, function cannot even distinguish conventional chair shapes from toilets, thrones, and seats.]
>
> Of course, that analysis would fail to capture toy chairs, or chairs of such ornamental delicacy that their actual use would be unthinkable. These would be better handled by the method of excuses, in which one would bypass the usual geometrical or functional explanation in favor of responding to *contexts* involving *art* or *play*.
>
> [This is what is required alright, but by what elementary features are these contexts to be recognized? There is no reason at all to suppose that one can avoid the difficulty of formally representing a chair by abstractly representing even more holistic, concrete, culturally determined, and loosely organized human practices such as art and play.][28]

This passage, and other such observations as "trading normally occurs in a social context of law, trust, and convention. Unless we also represent these other facts, most trade transactions will be almost meaningless,"[29] show that Minsky has understood the lesson of my book, *What Computers Can't Do,* which argued that intelligent behavior requires as background the totality of practices which make up the human way of being in the world. But Minsky seems oblivious to the hand waving character of his proposal that frames will enable workers in AI to represent all this background in explicit descriptions, as if the programmers could make explicit the totality of activities which they have picked up by training without recourse to explanations or descriptions, and which pervades their life as water encompasses the life of a fish. In the light of the fundamental unavailability of this tacit know-how, one can't help feeling, when Winston ends his praise of Minsky's "first step" with the challenge: "Much remains to be done,"[30] that this is just a tactful way of saying: "Nothing has been accomplished."

One might retrench once more, however, and claim that, although common sense categorization of chairs and tables is too concrete and tied in with human practices to be amenable to formal representation, one might still produce a formal model of pure thought. In that case, science would seem to be an ideal subject for computer simulation, since as a detached theoretical enterprise it deals with context-free attributes, whose law-like relations can, in principle, be grasped by any sufficiently powerful intellect, whether human, Martian, digital, or divine.

Yet, according to philosophers and historians of science, even scientific research requires concrete paradigms for its success. Just as everyday problem solving and more developed forms of technology take place in a practical context which makes possible *insight* into which aspects of objects are *significant* for the task at hand, so *all* appeal to attributes whether practical or theoretical requires *abduction* to exclude from consideration all but a limited number of the possibly *relevant* factors. In science this job is done by an implicitly agreed upon paradigm of successful scientific practice which leads the scientist to notice only a pre-selected sub-set of the possibly relevant factors. Otherwise, the scientist is as hopelessly lost as a Martian or computer. As Thomas Kuhn notes: "In the absence of a paradigm or some candidate for paradigm, all the facts that could possibly pertain to the development of a given science are likely to seem equally relevant."[31]

Minsky in his frames article claims that: "the frame idea...is in the tradition of...the 'paradigms' of Kuhn."[32] It is thus instructive to see how a theory of formal representation such as Minsky's misses the point of Kuhn's analysis. After quoting Kuhn's description of a "paradigm-induced gestalt switch,"[33] Minsky interprets as follows:

> According to Kuhn's model of scientific evolution "normal" science proceeds by using established *descriptive schemes.* Major changes result from new "paradigms," new ways of describing things that lead to new methods and techniques... Whenever our customary viewpoints do not work well, whenever we fail to find effective frame systems in memory, we must construct new ones that bring out the right *features.*[34]

But what Minsky leaves out is precisely Kuhn's claim that a paradigm is not an abstract descriptive scheme in terms of formal features, but a set of shared concrete practices, Indeed, a commonly accepted example of good work, in order to perform its function of providing continuity of agreement, cannot and must not be "rationalized," i.e. made explicit and abstracted from accepted examples of successful science. As Kuhn puts it:

> Scientists can agree that a Newton, Lavoisier, Maxwell, or Einstein has produced an apparently permanent solution to a group of outstanding problems and still disagree,...about the particular abstract characteristics which make those solutions permanent. They can, that is, agree in their *identification* of a paradigm without agreeing on, or even attempting to produce, a full *interpretation* or *rationalization* of it. Lack of standard interpretation or of an agreed reduction to rules will not prevent a paradigm from guiding research. ...Indeed the existence of a paradigm need not even imply that any full set of rules exists.[35]

The point is that even in the area of abstract thought, it is important for the development of science that the underlying practices *not* be fixed in abstract symbolic structures, for the rigidity of an explicit descriptive scheme would eliminate the necessary adaptability to new situations. That is why, as Kuhn puts it:

> Paradigms may be prior to, more binding, and more complete than any set of rules for research that could be unequivocally abstracted from them.[36]

Thus, although it is the job of scientists to find abstractible, formal explanations, and the subject matter of science consists of such formal accounts, the *work* of scientists themselves does not seem to be amenable to this sort of explanation. Indeed, if each scientist had internalized a complete formal description of the defining features of his discipline, this scheme would, as Minsky remarks, require explicit "redefining of 'normal'"[37] for each modification of scientific practice. But, according to Kuhn, this is precisely what does not occur.

What can we conclude, then, concerning the contribution of MIT AI research to the science of psychology? No one can deny Minsky and Papert's claim that "Computer science has brought a flood of... ideas, well defined and experimentally implemented, for thinking about thinking..."[38]. But all of these ideas can be boiled down to ways of constructing and manipulating symbolic descriptions, and, as we have seen, the notion that human cognition can be explained in terms of formal representations does not seem at all obvious in the face of actual research on human perception, everyday concept formation, and abstract scientific thought. Still, Minsky and Papert show a commendable new modesty. They only claim that:

> Just as astronomy succeeded astrology, following Kepler's discovery of planetary regularities, the discoveries of these many principles in empirical explorations of intellectual processes in machines *should* lead to a science, eventually.[39]

Happily, "should" has replaced "will" in their predictions. But their research actually suggests an even more modest hope: ideas derived from computer programming may lead to a science, eventually, probably at the neurological level, if psychologists only learn to profit from AI's mistakes.

Hubert L. Dreyfus
University of California
Berkeley, CAL.

FOOTNOTES

1. Potential purchasers should be warned that these Condon Lectures (Copyright 1974) were given in 1971 and are simply a reprinting (minus chapter 6 on Winograd) of MIT MAC Memo 252 (Jan. 1, 1972). Apparently someone thought that three years after the original presentation not a word needed to be changed to bring this discussion of the state of the art in AI up to date.
2. *Semantic Information Processing,* Marvin Minsky, ed., MIT Press, Cambridge, Mass. 1968.
3. Ibid., p. 7.
4. *Artificial Intelligence,* p. 6.
. *The Psychology of Computer Vision,* p. 2
6. R.N. Shepard, and B. Metzler, "Mental Rotation of Three-Dimensional Objects," *Science,* 1971, pp. 701-703. Minsky recognizes in his frames article that "Many psychologists feel that the experiments of Shepard on matching rotated objects indicate that humans perform continuous operations upon picture-like images." (p. 273), but he dismisses this view in a few sentences. For a more detailed attempt to save formal representations in the face of the latest findings concerning images, see Zenon Pylyshyn's forthcoming paper, "Imagery and Artificial Intelligence," *Minnesota Studies in the Philosophy of Science,* Vol. IX.
7. Eleanor Rosch, "Human Categorization," in N. Warren (ed.) *Advances in Cross-Cultural Psychology* (Vol. 1), London, Academic Press, in press.
8. *Artificial Intelligence,* p. 34.
9. "A Framework for Representing Knowledge," *The Psychology of Computer Vision,* p. 215.
10. Erich Goldmeier, *Similarity in Visually Perceived Forms,* International Universities Press, New York, 1972, p. 1.
11. Ibid., p. 118.
12. Ibid., p. 128.
13. "A Framework for Representing Knowledge," p. 275.
14. *The Psychology of Computer Vision,* Chapter 5.
15. Ibid., p. 185.
16. *Artificial Intelligence,* p. 54.
17. Op. cit., p. 158.
18. Ibid., p. 194.
19. Ibid., p. 193.
20. Ibid., pp. 193-194.
21. Ibid., p. 160.
22. *Artificial Intelligence,* p. 56.
23. Ibid., p. 56.
24. Op. cit., p. 157.
25. Op. cit., preprint, p. 41.
26. "A Framework for Representing Knowledge."
27. *The Psychology of Computer Vision,* p. 16.
28. Op. cit., p. 255. My italics and square bracketed comments.
29. Ibid., p. 240.
30. Op. cit., p. 16.
31. Thomas Kuhn, *The Structqre of Scientific Revolutions,* University of Chicago Press, Chicago, Ill. 1962, p. 15.
32. Op. cit., p. 213.
33. Ibid., p. 260.
34. Ibid., p. 261. My italics.
35. Kuhn, p. 44.
36. Ibid., p. 46.
37. "A Framework for Representing Knowledge," p. 261.
38. *Artificial Intelligence,* p. 25.
39. Ibid., My italics.

Recursive Programming Techniques, W. H. Burge. Addison-Wesley Publ. Co., 1975. 272 pp., $15.75

This is a graduate-level book on programming which will appeal primarily to Computer Science majors. The presentation is taut, rigorous, and thorough, and requires a high level of motivation from the reader. As in most of the books in this (The Systems Programming) Series, there are no exercises, so it is more of a reference text than teaching text, but it is excellent in either role.

Some excerpts from the table of contents may give the flavor of the book:

1. Basic Notions and Notations
   1.3 Variables and Lambda Expressions
   1.4 Data Structures
   1.10 Recursive Functions
2. Program Structure
   2.2 Reverse Polish Programs
   2.8 Compiling Expressions
   2.10 Labels and GO TO Statements
3. Data Structures
   3.5 List Structures
   3.6 Trees and Forests
   3.10 Sequences, Coroutines, and Streams
4. Parsing
   4.3 Context-Free Languages
   4.6 Left-Corner Bottom-Up Parsing
5. Sorting
   5.3 Binary Search Trees
   5.5 Quicksort
   5.9 Tape Sorting

L. D. Yarbrough
Lexington, MA

*****

Intelligent Life In The Universe, by I. S. Shklovskii and Carl Sagan, 509 pp. $3.25. Dell Publishing Co., New York, 1966.

The Cosmic Connection: An Extraterrestrial Perspective, by Carl Sagan, 274 pp. $1.75. Dell Publishing Co., New York, 1973.

These two books are roughly parallel in structure; both cover essentially the same ground. *Intelligent Life,* a translated and annotated version of Shklovskii's Russian edition of *Universe, Life, Mind,* contains more detail and information but is somewhat awkward reading because of its annotated style. *The Cosmic Connection* is much more readable, with less detail and a clearer flow of ideas. Of the two books, I would recommend *Connection* because it is exactly this grand flow of ideas which is important.

Connection is divided into three parts: "Cosmic Perspectives", "The Solar System", and "Beyond The Solar System." Sagan is concerned with who we are, what we are and where we are going. The book is as broad as that. He touches on every aspect of life which should be of concern to intelligent beings, from the evolution of our own civilization, to a modest plan for searching for extraterrestrial life, to some really wild scenarios of possible futures.

Reading this book should be done slowly, on a quiet evening. After you have finished reading, go stand under the night sky and let all those numbers with their incredible magnitudes and implications roll around your head. Think BIG. Think on a civilization lasting for millions of years; of waiting a thousand years for a reply to a message; of what you would say in such a message that would be of value to another world. Think on Sagan's observation: "We are at an epochal, transitional moment in the history of life on Earth. There is no other time as risky, but no other time as promising for the future of life on our planet."

John Lees
Rolla, MO

Artificial Intelligence. Earl B. Hunt. 468 pp. $29.00 Academic Press, Inc., 111 Fifth Avenue, New York, N.Y. 10003 (1975).

This book fulfills a great need for a detailed description of the "state of the art" in the field of research commonly called Artificial Intelligence. There have been previously published research monographs that concentrated on specific points of view and described, to a great extent,the author's own researches - and occasionally their relation to the work of others. We also have the excellent introduction by Phillip Jackson ("Introduction to Artificial Intelligence", Petrocelli Books, New York, 1974) which provides an excellent overview but does not go into much detail.

In the first section of his book, Hunt devotes a chapter to giving a general overview of the field and a chapter to the theory of computation and formal grammars. The next three sections deal with Pattern Recognition (5 chapters), Theorem Proving and Problem Solving, in which the author also includes game playing (5 chapters) and comprehension (3 chapters). The author's treatment has been exhaustive and detailed. The reader is not only informed that methods exist. He or she is taught the methods, shown how they are applied and what basic assumptions make them applicable. Of course, at our present level of understanding the latter is not always possible—but the author has tried to be as thorough in his analysis as he could.

For the reader who wonders why a chapter on Computation Theory is included, the author includes a statement at the end of Chapter 2: "These results... tell us that there are a number of interesting problems that can not be solved by an algorithm and many that can not be solved by a simple one." The student of A.I. is warned, by this, to lower his or her sights in the interest of practicality. The author does not refer to the new results on complexity of computations—which point out that many algorithms give rise to computation times which grow very rapidly with problem size and have to be used cautiously.

It is somewhat disappointing that the section on Pattern Recognition, while containing detailed analyses of both the techniques he calls "sequential" and "grammatical," does not point out that they represent a trend towards greater and greater expressibility in the language of pattern description. Nor does he point out that the Evans program, mentioned by him in Sec. 7.2, is a step ahead of other syntactic description methods in that relations like "larger than" are as easy to use in that language as the mere "attached to" that most syntactic methods are constrained to.

This section contains, in addition to discussions of the syntactic and logical methods of description, discussions of those methods that involve statistical and algebraic techniques. A detailed discussion of Minsky and Papert's work on linearly separable functions has, laudably, been included. This is good. It is however somewhat disappointing to see that some of the work on "growing" languages (like Sherman and Ernst's work on learning concepts in terms of other concepts) has not been mentioned nor has the author pointed out the close relationship that exists between a good pattern description language and the "comprehension languages" discussed in Chapter 14.

The recent work on "algorithms for finding a minimal path to a single goal node" which has become a basis for a new theory of heuristics, is discussed in detail in Chapter 10. This yields the impression that the only theory of heuristics that is possible is numerical in nature. However, the later work of Ernst, following his GPS book with Newell, establishes a viable alternative approach.

These objections to Hunt's book are basically 'nit-picking.' Such objections are possible (or even needed) only because Dr. Hunt's book is likely to set a trend for the field for some time to come, so that in the absence of such nit-picks, some worthwhile recent approaches might get forgotten.

Hunt can be both commended and criticised for not introducing his own biases into the discussion of the overall structure of the field. He observes that "Problem solving by computers.... requires us to think simultaneously in terms of graph theory and formal logic with an occasional use of statistical reasoning... some knowledge in a great many fields may be called for." This situation will continue to exist and one can only hope that a proper understanding of the interaction of these various fields within the field of Artificial Intelligence will be possible. Many people have conflicting views on the form this understanding will take. The author has referred to few but not to all. Given the current state of the art, perhaps it is all the better that the author's analysis has focussed on the algorithms and not in their relationship with each other.

This book will be of great value to new workers in the field. It gives a solid introduction to the techniques and conventional wisdom of the field. But it bothers me that Hunt criticises techniques developed by authors who obtained them as a side product while developing formal theories of problem solving. In Hunt's opinion, "in many specific A.I. projects the informal approach ... would have been as satisfactory as the often forbidding formalisms ..." This reviewer has by and large to agree. However, he would like to note that these highly formal techniques have yielded algorithms of some power and generality about which the general reader has the right to be told. Luckily, Hunt does make references to papers describing such work, and, through these references, to methods motivated by these formal points of view. The harm of these omissions therefore is not irreparable.

R. Banerji
Temple University, PA.

*****

On Machine Intelligence - Donald Michie, 199 pages, Edinburgh University Press, 22 George Square, Edinburgh. 1974. (Published in U.S. by Halstead Press, $6.95.)

As a discipline, Artificial Intelligence is not yet thirty years of age, and serious experimental and engineering work is even more recent. There have been few persons whose work on the field span this same time, yet Donald Michie stands out as one of the half dozen or so figures who symbolize schools of thought on A.I. This book is a collection of fifteen essays, arranged chronologically, which were published elsewhere from 1961 to 1974 and as such they very well portray some of the central themes of Michie's work during this period.

The early works give much attention to game playing which shifts to a more general concern for theories of learning and finally to robots (Integrated Cognitive Systems) as ideal tests for much of the working theory. It is clear from these pieces that Michie is mainly concerned with implementing theoretical developments and consequently the articles have more of an "engineering" rather than "science" bent.

In any collection of this length there is bound to be some redundancy yet on the whole the book is enjoyable and quite non-technical. "Trial and Error" and "Machines and the Theory of Intelligence" are especially good expositions, easily suitable to advanced high school students. I would think the person unfamiliar with A.I. work would gain a very good survey of the kinds of issues with which the Edinburgh group has been concerned. The essays are not technical and do not depend on any knowledge of computing. My only regret is that, as popularly written articles, they often serve only to whet the appetite.

John J. Neuhauser
Chestnut Hill, MA

*****

The Sun Never Sets On IBM: The Culture And Folklore Of IBM World Trade, by Nancy Foy, William Morrow & Co., Inc., 218 pp., $7.95, 1975.

Everyone knows that IBM is big, but few outsiders realize that IBM is more than just a gigantic multinational corporation. IBM is a world in itself, with its own way of doing things, its own language and customs, its own system of ethics and a surprising amount of control over how it is seen by the non-IBM world. Nancy Foy gives a glimpse into the unique entity which is IBM World Trade; the organization with the motto, "World Peace Through World Trade."

The overall impression one gains from the book is that IBM is a good company which has become almost too successful. The company has remained largely non-unionized by the method of giving its employees outstanding working conditions and job security, along with membership in the IBM extended family. IBM expects much from its employees, but it also takes care of its own. Most ex-IBMers remain quite loyal to the company.

IBM is not, however, completely independent of the outside world, it must sell products, and the outside world is changing. IBM is a growth oriented company with revenues in the billions of dollars per year and it is fast running out of room and ways to grow. Everywhere it turns, IBM faces possible anti-trust action. IBM can not even lower the prices of its own products (which in many cases it could easily afford to do) appreciably without wiping out its competitors. IBM now faces the challenge of changing its internal structure to cope with a less growth oriented world.

Nancy Foy's book is one of the few sources of information on the internal history of a company which touches everyone's lives, for IBM rarely makes public information of its own accord. *The Sun Never Sets* is also worth reading for the anecdotes and tall tales alone; the men who all dress the same, IBM's operation in World War II Europe ("some of IBM's German profits went to the Resistance"), the fantastic amounts of money involved. All in all it is a very interesting book, although one is left with the nagging feeling that a lot still remains untold.

John Lees
Rolla, MO

*****

Applications of Computer Systems, Richard A. Bassler and Edward O. Joslin (Eds.), 164 pp., $3.95, College Readings, Inc., P.O. Box 2323, Arlington, VA 22202.

A compilation of published articles on computer use in many fields, this book is intended as a text or (more likely) supplementary text to be used in a course for students already familiar with the fundamentals of computer operation to familiarize them with the types of potential applications identified in a survey of employers.

The very broad spectrum of applications represented (with excessively brief introductions to units by the compilers of the text) does serve to make readers aware of the often expressed, seldom exhibited universality of computer capabilities. In this regard, the book could prove useful to students planning to seek employment in the field.

Most of the articles are on specific applications, few lend themselves to adaptation or transfer, and the accompanying bibliography follows the same pattern—any generalization is left to the reader or his instructor. The book is probably useful in a course preparing students for supervisory roles in computer-associated fields or by employees in such positions.

John L. Randall
Kensington, MD

*****

The Assault on Privacy: Computers, Data Banks, and Dossiers, by Arthur R. Miller, University of Michigan Press, 1971.

While it's true that Mars is the next stop in our conquest of space, this is NOT the "space age". It is the Age of Information. I have not christened this current era nor has Professor Miller but if you read *Assault on Privacy,* you certainly will not forget it. It is not so much the information as the control of it. This detailed, and incredibly well-documented, account will give you second thoughts when you fill out forms and will send you into your own memory banks trying to recall how much information you have already, quite off-handedly, given out about yourself in the past. In fact, it probably will make your hair stand up, incense you and make you want to do something about this threatening area of American life. This is when the real problem will strike home—"what" and "how". It is to say the least overwhelming and that in itself enhances the "consciousness-raising" impact of this book. Professor Miller, after chronicling situation after offensive situation, offers the beginnings of a solution. His direction is a legal one and the one he can speak for most adequately, being a professor of law. His arguments and observations are, at times, complex and difficult to follow but the final result is rewarding if you stick with it. At first, I had a negative reaction as the book progressed but then I chalked it up to my own Americanism. By that I mean the kind of wild-west mentality that possesses each of us in a problematic situation—"there must be some kind of quick and decisive AND terminal way to handle this". Perhaps we think that no problem is worth too much thought and effort, after all, we are blessed here in the United States with invincibility. On the contrary, I feel as does Arthur Miller, that we can no longer face complex problems with simplistic approaches and that we must develop the sophistication and the follow-through to beat the "bugs" in our way of life. However, I myself want to go beyond just bureaucratic solutions and also develop a personal philosophy to maintain what is most personal in our way of life while at the same time employing the best of what technology has to offer, whatever that is. We have yet to define this or to set up the proper controls. So I suggest you read this book and that you not make it your last on the subject.

Bill Griffith
Boston College

*****

The Moon Is A Harsh Mistress, by Robert A. Heinlein, 302 pp. Berkley Medallion Books, New York, 1966

Robert Heinlein is of course one of the greatest science fiction writers of all time. He has in fact written a not too improbable future history of our portion of the Galaxy. *The Moon Is A Harsh Mistress* is not strictly part of Heinlein's Future History series, but it is great reading and it contains a very interesting character: Mike (short for Mycroft), a truly intelligent computer.

The story is about a revolution (The bang, bang you're dead kind. Heinlein is a rather bloodthirsty author. He would call it being realistic.) in which the penal colony on Earth's Moon tries to assert its independence and break away from Earth. The revolution is led by a cabal of three "Loonies" and Mike—a "High-Optional, Logical, Multi-Evaluating Supervisor, Mark IV, Mod L" HOLMES FOUR computer which "woke up" one day after its complexity had reached a sufficiently high level.

As Heinlein paints its personality, Mike is a child prodigy with a prankish sense of humor; no morals, no sense of right and wrong, and loyal only to the technician who recognized its sentience and took the time to talk with it as a sentient being instead of as a machine. Mike becomes the *de facto* leader of the revolution; tricking the Warden, controlling the phone network, calculating strategies, operating the Moon's only possible weapon against Earth and actually giving orders itself.

Mike is such a really lovable fellow, telling jokes, pulling pranks, writing poetry even, and is so obviously on the side of the good guys, that you tend to push to the back of your mind the fact that Mike also kills people. Now this is not to say that Mike is some kind of horrible electronic monster; if any revolution is justifiable, theirs is. Mike has simply progressed from making, for instance, inventory decisions to making life and death decisions, not always in favor of life.

So after you have enjoyed the story, spend a little time thinking about what life will mean to the first computer that wakes up. No small child understands life and death, but what small child has the power of life and death?

John Lees
Rolla, MO

*****

The Listeners, by James E. Gunn, 240 pp., $1.25, Signet Books, 1972.

The Project had been in existence for fifty years. Fifty years of listening among the thin hissing of the Galaxy for a message saying that Earth is not alone. But no message had yet come, and now Congress was balking at funding and the highbrow press was out to bury the Project as an example of fruitless waste. They said that fifty years was long enough; the scientists of the Project were prepared to wait centuries. Meanwhile, the first message had been received and was waiting to be noticed.

James Gunn presents the story of the Project through the lives caught up in it and in the message. He explores the effect which receipt of such a message might have on Earth and the effect on society of waiting more than a lifetime for a reply to be answered. Gunn takes a few technical liberties, the worst of which is assuming that such a message will take so long to decode once it is recognized as a message, but they do not get in the way of a good story.

John Lees
Rolla, MO

Introduction to Data Processing. Martin L. Harris, 326 pp. $3.95. John Wiley and Sons, Inc., New York. 1973.

Harris presents the fundamentals of data processing in a programmed format. Each chapter is divided into numbered sections that present new information. The objectives are identified at the beginning of each chapter and each chapter also contains a review. You may use the review as a pretest and, if satisfied with your performance, you may skip that chapter.

The basic concepts of data processing are presented in an elementary, low-key fashion. The book could truly serve as an "introduction" for the neophyte. Little previous knowledge of data processing or computer application is assumed. "The purpose of this book, then, is to give you some basic understanding of what data processing is, how it is organized, what types of equipment are used, and how a particular data processing system is designed." A cross referencing chart to other data processing texts is provided for the reader who seeks additional information or another perspective to the same topics.

One chapter provides instruction in the BASIC programming language as an illustration of how instructions are written for a computer. Uses for other languages (COBOL, FORTRAN, RPG) are only briefly mentioned.

The text can be recommended to someone who is beginning a study of data processing. It provides an introductory background in a programmed format for self-study. It might also be used as a review text. It is unfortunate, however, that only masculine pronouns (he, his, him) are used throughout the book.

Jane Donnelly Gawronski
San Diego Department of Education
San Diego, CA. 92111

*****

Fun and Games with the Computer. Edwin R. Sage. 351 pages, $5.95. Entelek, 42 Pleasant Street, Newburyport, MA 01950. 1975.

The author proposes to teach computer programming, using the BASIC language, through games. The reader will have mastered the fundamentals of programming and will have learned one computer language, BASIC, if all suggestions for programming and exercises are worked out.

The book is arranged in 7 chapters, each carefully explaining the rules of the game to be programmed, a flowchart of the procedure for playing the game followed by the BASIC program. The latter half of each chapter deals with improving the program, adding 'personality' and checks to ensure the rules are followed by the computer's opponent. As the games become more complicated, ranging from number guessing to Blackjack, so do the programming and language requirements, thus by the end of the book, the computer user has a good feel for the capabilities of the BASIC language.

Although the author requires an interactive computer system, the book does not require any specific system. Exercises in Chapter 1, and reminders throughout the book, direct the reader to inquire into the local systems requirements for running the BASIC language.

The book, as the author suggests, could be used as a supplementary text for a course involving gaming.

The book is recommended for those involved in teaching the BASIC language or anyone wanting to learn the BASIC language. One ends with a small library of interesting games which can be used for demonstration purposes, if the computer system being used does not have such games present.

John R. Jackobs
Coe College
Cedar Rapids, Iowa

(Available from
Creative Computing Library.)

101 BASIC Computer Games, David Ahl (Ed.), 249 pp., $7.50. Digital Equipment Corp., Maynard, MA. 1974. (Available from *Creative Computing* Library. See ad.)

The paperback book, *101 BASIC Computer Games* contains brief descriptions, BASIC listings and runnings of 101 "games". The programs were collected from a great variety of sources and range from simple picture-printing routines to involved simulation games. As intended, *101 BASIC Computer Games* provides instant stimulation and motivation to intermediate level programmers.

This 8½ by 11 inch book is clearly and conversationally written with legible reproductions of program listings. My junior high school students had no difficulty reading the descriptions preceding the game programs. Interspersed throughout the text are many clever cartoon sketches. Other than the use of the word, "varied", it is impossible to characterize the games included. Mathematical games, simulation games, card games, and sports games are all represented. Titles range from the familiar (NIM, POKER, BASEBL, LIFE) to the surprising (ANIMAL, MUGWMP, SPACWR, ZOOP). The level of the programming is uneven and a variety of versions of BASIC is used. My students delighted in "improving the original" and debugging. Often the latter operation consisted of translating the program into one of our dialects of BASIC (H-P or IBM 370). This translation process itself proved to be an interesting and simulating exercise.

101 BASIC Computer Games is a valuable resource book for teachers and students of BASIC. Its utilization will depend on the course objectives and on your teaching strategy. I have found it to be useful as a source of project ideas for students who have already gained some fluency in BASIC. Additional project ideas and exercises are contained in the companion volume described below.

Patrick Corry
Irvington High School
Irvington, New York

*****

Understanding Mathematics and Logic Using BASIC Computer Games, David Ahl, 60 pp, $1.50. Digital Equipment Corp., Maynard, MA. 1974.

This book extends and analyzes many of strategies and ideas involved in the games contained in *101 BASIC Computer Games.* In this extension, the author discusses logical strategies, gives exercises, and suggests projects related to some of the 101 games. There is a brief but illuminating introduction which includes some thoughts about the use of teams and games as teaching tools. The use of mathematical logic in games is discussed with several examples. The exposition is clear and brief and is intelligible to a high school student. The exercises and project suggestions will be especially useful to those teachers who wish to use *101 BASIC Computer Games* as a source for a computer literacy course. All teachers of BASIC programming should find this volume useful as a source of ideas for discussion and individual projects.

Patrick Corry
Irvington High School
Irvington, New York

*****

What To Do After You Hit Return, 157 pp., $6.95. People's Computer Co./Hewlett-Packard, Menlo Park, CA. 1975. (Available from *Creative Computing* Library. See ad.)

This is an interesting and timely book. At all levels it represents a melding of diverse, even polarized elements. At once it is a potpourri of games, an activity dating back to the roots of man, and a catalog of creative ways to use man's most modern and sophisticated tool, the computer. The contents range from old number games like NIM, which originated in ancient China, to STTR1, a simulation adapted from a modern science fiction television series. The computer version of NIM (and many of the other games) is merely an adaptation of an age-old diversion; these games may be enjoyed equally well without the use of a computer. Games such as STTR1, on the other hand, would never have been possible without the speed and computational abilities of modern computers. (In fact, STTR1 not only requires a computer to be played; it includes a computer as an integral part of the game.)

The publishing of this book represents a combined effort. The computer games and simulations were contributed by a variety of people, from mathematicians to businessmen, from students to professors, and from computer phreaks to homemakers. Indeed, the book was made possible through the joint cooperation of the People's Computer Company, a store-front, non-profit, educational group that publishes a funky newspaper, and Hewlett-Packard Corporation, a more traditional company that manufactures time-sharing computers, among other things. This, in itself, is significant.

In short, *What To Do After You Hit Return* is destined to become one of *those* books . . . It is conspicuous—one of those books that is too big to fit on the shelf, so you find it lying about on a table; it is eclectic—one of those new, soft-cover newsprint catalogs that is crammed to the margins with interesting tidbits and graphics; it is a curiosity—one of those books you feel compelled to pick up, just to see what is inside; and most important, it is an educational resource—one of those books that will help you find, obtain, or "get into" new materials for the enrichment of learning.

Bob Kahn
Lawrence Hall of Science
University of California, Berkeley

*****

Game Playing With Computers, by Donald D. Spencer, $12.95. Spartan Books, New York, 1968.

This unusual book, completely devoted to computerized game playing, introduces more than 70 games, puzzles and mathematical recreations that may be programmed for a digital computer.

Over 25 complete programs are presented, most of which have never before appeared in computer books. Each of the programs includes a description of the game, flowchart, a program written in FORTRAN, BASIC, a description of how the program works, and output produced by the program, In addition to an in-depth analysis of over twenty different types of magic squares and how they may be generated by a computer, the book features complete programs for prime numbers, Sieve of Eratosthenes, pick-a-number, blackjack, binary card games, the counterfeit coin game and 15 other puzzles.

To stimulate the interest of students and beginning programmers and to challenge the ingenuity of more experienced senior analysts, senior programmers and mathematicians, more than 50 games are presented for computer solution. These include such popular pastimes as Keno, Roulette, Go, Chess, Nim, Pantomino and Tic-Tac-Toe.

*****

Games, Tricks and Puzzles for a Hand Calculator, Wallace Dudd, 100 pp, $2.95. Dymax, Menlo Park, CA, 1974. (Available from *Creative Computing* Library. See ad.)

Any young person who has access to an electronic calculator of any description will find something in this book to challenge and amuse him or her. A child just learning the elements of arithmetic will find games and tricks at the appropriate level. At the same time, there is at least one game that is sufficiently difficult to analyze so that a trained adult mind will have difficulty in finding a winning strategy. (The reviewer admits to not having solved it. Our solution appears in a separate article in this magazine.)

There is a rapidly growing literature on the hand calculator, but I find this book outstanding because of its attractive presentation and wealth of ideas. For example, there is a good chapter on isolating malfunctions, which might have been subtitled "How to Become the Sherlock Holmes of the Calculator Age." All in all, well written and a lot of fun.

L.D. Yarbrough
Lexington, Mass.

*****

Games Calculators Play. Thomas J. Seymour, 32 pp, $2.00. Seymour Publications (P.O. Box 1788, Rockford, IL 61110), 1975.

I don't know who first turned his calculator around with 0.7734 on it and noticed it now said "HELLO", but immediately a new pastime was born: calculator word games. Unfortunately the vocabulary is limited to I,E,H,S, L,B,O,G, and D depending a bit upon the calculator and your imagination.

This little book is a collection of 57 simple but clever calculator problems similar to the following: "If a man invests $211,843 in the stock market (ENTER 211843) and his stock prices fall 26 percent (ENTER x 26), what does he have?" (Answer: BIG LOSS) Also included is an appendix of some 250 words possible to form on a calculator. Lots of 5379919.

David H. Ahl
Morristown, NJ

*****

Mathematical Carnival, Martin Gardner, $8.95. Alfred Knopf, New York, 1975.

So everyone has a pocket calculator. You casually remark at a party that you don't need one of these toys because you can multiply two nine-digit numbers in your head. Ha, ha. Someone challenges you. "Get pencil and paper" you say, "and I'll give you the one nine-digit number and you give me the other." You write 142,857,143, your friend writes, say 123,456,789. Without hesitation you take the paper and write 17,636,684,160,493,827. He takes it off to a corner and after 5 or 10 minutes is back looking at you with more awe than before. Incidentally, most small pocket calculators can't handle this problem—too many digits.

Gardner explains this trick along with scores of others such as naming the day of the week for any specified date. He also describes the work of Dr. Fliess, a Berlin physician who was talking about biorhythms (male 23 days, female 28 days, he believed) way back in the 1800's. Included too are a variety of games including the particularly intriguing "Sprouts."

This book has something for everyone, beginner and enthusiast alike. You'll enjoy it!

David H. Ahl
Morristown, NJ

*****

The Computerized Society. James Martin and Adrian R. D. Norman, 560 pp, $12.50. Prentice Hall, New York, 1970.

The Computerized Society is a lengthy book of 29 chapters, divided into three sections which discuss what have become the traditional topics of computer and society. Attention is given to the rapid growth of computing in the last 30 years, present application areas, the social and philosophical consequences of the new technology, and the controls which may be necessary to insure the proper use of computers.

The book ranges from well-documented chapters on the growth and uses of computing, to the authors' rather nebulous attempts at formulating policies which would lead to a Utopia with the new technology. Too much attention is given to categorizing mankind and then assigning jobs based on the often questionable results of IQ tests.

One problem with books of this type is that they become dated almost before they get into print. For example, the quote opening the chapter on Law Enforcement: "When little old ladies have to wear tennis shoes so they can outleg the criminals on the city streets, there's something wrong with the way we're doing things" is by Spiro T. Agnew. Certainly a better choice could have been made.

The book does give a thorough treatment of all the topics important to computers and society, and, in general, is useful background material for the professional in computing or the layman concerned about the implications of the so-called "thinking machines". Unfortunately, given the book's shortcomings, it cannot be recommended for general reading.

David L. Feinstein
University of Wisconsin
River Falls, Wisc. 54022

*****

Simulation Games in Learning, Boocock, S.S. and Schild, E.O. (Eds.). Sage Publications, Beverly Hills, CA. 1968.

In 1968, when *Simulation Games in Learning* was published, simulation games—the combination of the ancient technique of gaming with the relatively recent technique of simulation—were viewed as an educational innovation. When published, the purpose of the book was to present a progress report on this new technology, i.e., present a "valid picture" of what is known today about simulation games in learning and of current work in the area. The volume actually grew out of two issues of the *American Behavioral Scientist* edited by James S. Coleman.

In 1975, the book has very little new information to offer the educator interested in instructional computing. Most of the research findings have been reported or summarized in more recent publications. (See, for example, *Simulation and Gaming in Social Science* by Inbar and Stoll, 1972, or *Simulation Games for the Classroom* by Mark Heyman, 1975). In addition, only one of the seventeen articles deals with computer-based simulation games. In that article, Richard Wing describes a small-scale, controlled experiment designed to examine the applicability and learning effectiveness of two simulation games—the Sumerian Game and the Sierra Leone Game—with sixth grade students. The results of the experiment led the author to conclude that "... computer-based games can be used in practice even with sixth graders; they do teach as well as conventional classroom methods; and they seem considerably more effective than conventional methods, when the time investment of the student is taken into consideration."

Individuals that have used or run some version of the Sumerian Game (SUMER) may find this single article interesting, since it gives some information on how the simulation game is intended to be used.

Dan Klassen
Lauderdale, Minn.

*****

Math, Writing & Games in the Open Classroom by Herbert R. Kohl, 252 pp. $2.45. Vintage Books by Random House, New York ,1974.

This book is two small, loosely related works bound in a single jacket. The first part "Approaches to Writing" is partly a rehash of topics in the author's earlier book *36 Children,* and is based upon his experiences teaching writing and self discovery to children in a Harlem school, as well as conducting seminars for elementary school teachers.

Part II, "Games and Math", deals with simple, inexpensive games for young children which were created by both adults and children and which help make learning about science, geography, music and mathematics more fun. Simple versions and modifications of games such as chess, checkers, nim, Go and Wari are illustrated and discussed. Since the book appears to have been written for the elementary school teacher who has not yet discovered educational games, most of the games are rather tame stuff for the game buff, especially the computer game enthusiast. However, there is something here for the expert, because Mr. Kohl categorizes and discusses games according to *themes, playing boards, pieces, decision devices, goals* and (most important) *how children learn to play games.* He also illustrates the educational value of games by showing how skills and attitudes can be developed through game playing, how games can be made a part of all areas of the curriculum, and what sources teachers can draw upon for ideas and resources for games. The distinctive approach to educational games in this book is that children should be encouraged to create their own games, to modify the rules of standard games to suit their own purposes, and to play games because they want to, not because it is required by the teacher.

The author discusses the similarities between the way theories are constructed in natural and social sciences and the nature of creating, exploring and modifying games. When creating games, children construct and explore theories, perform experiments, make appropriate modifications, and examine the consistency and applications of their theories. Games are shown to be good ways for children to learn how to work together and to make decisions.

While most readers may find a few new ideas for educational games in this book, the author's major purpose seems to be that of changing the reader's view of education and educational games and to present some of his own opinions about teaching and learning. This book is nontechnical and informal, and the general reader may find Mr. Kohl's approach to games and other teaching strategies in the "open classroom" interesting and informative. However, there is no mention of computer-based games.

Frederick H. Bell
University of Pittsburgh

*****

Getting the Most out of Your Electronic Calculator, William L. Hunter. $7.95 Hardbound, $4.95 Paperback. TAB Books, Blueridge Summit, PA, April 1974.

So Uncle Fred has given you a handy-dandy pocket calculator for Christmas—or perhaps you are contemplating getting one for yourself or a friend. What now? Is it really worth the money, and just what is it good for, except as this season's "in" toy?

This book is an attempt to answer some questions about what the little boxes are for and how to make some intelligent use of them. In so doing, it points up the amazing variety of hand calculators on the market and their differences and similarities.

The strongest points about the book are, first, it is about the only such book now on the market, and second, it has some interesting chapters on applications, especially on Income Tax preparation. Of particular interest are photos and manufacturer's blurbs on 27 different models. However, this portion of the book is already obsolete because of the dramatic drop in prices and the introduction of new models in the year or more since the book was compiled.

My evaluation of the book is largely negative. It is obsolete, as was mentioned above; although it is intended "to serve as a modern course in general and commercial mathematics", it does not seem well suited for its avowed purpose (it is filled with examples, but there are no exercises by which a student could measure his skills); it is far better designed as a reference than as a text book.

Finally, the examples are worked out in terms of a non-existent "typical" machine so that each algorithm must be translated into the framework of the user's particular capabilities. On the other hand, if you are buffaloed by your own calculator and feel you are not getting your money's worth, you ought to check this book out. It may be just what you need to get you started using it effectively.

L.D. Yarbrough
Lexington, Mass.

*****

Problems for Computer Solution, Fred Gruenberger and George Jaffray, 401 pp, $7.25. John Wiley & Sons, New York. 1965. (Available from *Creative Computing* Library. See ad.)

A superb selection of 92 problem situations from business and science as well as mathematics. Each problem is well described and specific exercises are identified.

Numerous settings are presented on prime number applications, decimal representations & probability related problems using random numbers. Many business examples are described including compound interest, dividends, sorting & statistical procedures (queueing, curve fitting, quality control).

Although finished programs are not presented, the problems and suggestions are clear and thorough.

Hank Kepner
Milwaukee, WI

*****

Primer in Computer Utilization. Richard F. Curtis and Maynard L. Erickson, 233 pp., $6.95. General Learning Press, Morristown, NJ, 1974.

Fortran IV is introduced by the authors with a non-mathematical flair. An attempt is made to develop in the reader a basic level of programming skill by stressing the logic of programming, giving experience in writing simple programs, and mastering each level of learning before progressing on to the next level.

The reader is led through numerous problem solutions aided by diagrams, sample program printouts, and graphic illustrations.

A social science, psychology, pre-medical, or other non-mathematics student may find this primer a refreshing first exposure to the realm of computing at the undergraduate level.

Gary D. Schafer
Lauderdale, Minn. 55113

*****

Computer Algorithms and Flowcharting. Gerald A. Silver and Joan B. Silver, 170 pp., $5.95. Gregg/McGraw-Hill, New York.

The authors lead the reader through the steps necessary to logically analyze a problem and formulate it in a fashion which is readily computer-programmable.

The topics covered can be loosely divided into four units. The first is devoted to problem analysis and algorithm development. The second unit involves flowchart preparation, including what is sometimes called macro and micro flowcharts, and a very detailed presentation of specialized flowchart symbols. Programming techniques such as conditional and unconditional branches, loops, counters, and arrays are treated in depth in a third unit. A final unit on applied programming logic involves presentation of real-life business situations and their solutions using the techniques previously discussed.

Each section in the book contains a set of exercises. The early exercises emphasize the language of computer programming. Later exercises reinforce key terms and provide practice in developing computer algorithms for solutions of problems. The text contains an abundance of clearly presented flowcharts that clarify the programming techniques and solutions to problems. Two nice points about the book are that no previous programming skill is needed, and that the book can be used with a variety of languages such as Fortran, Cobol, and Basic.

This well-written book would appeal to the business department rather than the mathematics department in a school. I would recommend this book for use in a data processing course with access to a computer, or as a reference book to be used for its description of flowcharting and problem analysis in the business field.

Bruce W. DeYoung
Oakland, N.J.

*****

Mathematics, A Human Endeavor. Harold R. Jacobs. 529 pp. $8.50. W. H. Freeman and Co., 660 Market Street, San Francisco, CA. 94104.

Mathematics in the Modern World. Readings from the Scientific American with introductions by Morris Kline. 409 pp. $7.00. W. H. Freeman and Co.

Although one doesn't have to know much mathematics to deal with computers, it doesn't hurt to know some. Here are two very attractive books (W. H. Freeman certainly knows how to put out a good looking book) that introduce various mathematical subjects to readers or students who may have missed them (or slept through them) on the way through school.

Jacobs subtitles his book "A textbook for those who think they don't like the subject" and although I happen to like the subject, I think it is a good book for those who don't. But it's good for people who like math too. His chapter titles suggest what his book covers: 1. The mathematical way of thinking, 2. Number sequences, 3. Functions and their graphs, 4. Large numbers and logarithms, 5. Regular Polygons, 6. Mathematical curves, 7. Some methods of counting, 8. The mathematics of chance, 9. An introduction to statistics, 10. Some topics in topology. Jacobs style and approach are charming. He keeps asking students and readers to think instead of just read. His book is full of good writing (His introduction to logarithms is a gem.) It is full of cartoons (Peanuts and others), advertisements, photographs and good diagrams. There is very little here that bears directly on computers but lots of things that people who deal with computers could use. This book is suitable not only as a high school or college text but also to read just for the fun of it. Highly recommended.

"Mathematics in the Modern World" is a book of readings from the Scientific American. Here are articles by Halmos, Kline, Davis, Euler, Kac, Quine, Dirac, Einsein, Ulam, Kemeny and other biggies. Among the articles directly relevant to the computer field are articles on "Mathematical Machines" (Davis), "Computers" (Ulam), "Computer Logic and Memory" (Evans), "The Uses of Computers in Science" (Oettinger), "Systems Analysis and Programming" (Strachey), "Cybernetics" (Wiener), "Man Viewed as a Machine" (Kemeny) and others.

Readers familiar with the Scientific American will know what to expect. Here are well edited and well illustrated articles that no normal person can quite use to learn about a field but which will get somebody started and will be appreciated by experts for saying a lot in a little space, and saying it both well and correctly.

The reader who plans to buy this book should note that there are two other books of readings ("Information" and "Computers and Computation: Readings from Scientific American") that also collect articles from the Scientific American about computers whose articles tend to overlap the ones in this book. Also Freeman sells reprints of articles from the Scientific American individually. Still this collection is very attractive and reasonably priced.

Peter Kugel
Boston, Mass.

Creative Computing Feature Review . . .

by Sema Marks

Learning Alternatives In U.S. Education: Where Student And Computer Meet. Beverly Hunter, Carol S. Kastner, Martin L. Rubin, and Robert J. Seidel. Educational Technology Publications, Englewood Cliffs, New Jersey 07632. 1975. xvi + 398 pp., $14.95.

Every few years another major study appears on the future of computers in education. There was the Rosser Report of the National Academy of Sciences in 1966;[1] the Pierce Report of the President's Science Advisory Committee in 1967;[2] the Oettinger and Marks report of the Harvard Program on Technology and Society in 1969;[3] the Tickton report of the Commission on Instructional Technology in 1970;[4] the Levien report of a study for the Carnegie Commission on Higher Education and the Rand Corporation in 1972;[5] *The Fourth Revolution*, a report and recommendations by the Carnegie Commission on Higher Education, also in 1972;[6] and now a report by Beverly Hunter, Carol S. Kastner, Martin L. Rubin and Robert J. Seidel in 1975.

The latest in this series, published as *Learning Alternatives in U.S. Education: Where Student and Computer Meet*, is based on a project performed at the Human Resources Research Organization (HumRRO) under a grant from the National Science Foundation. The purpose of the project was "to study development and dissemination of computer-based learning materials in the U.S. and to identify approaches for achieving beneficial, nationwide use of computers in education."

In the decade since the Rosser Report little has changed in the classroom, but a great deal has changed outside of it. The use of computers has grown substantially and people's attitudes towards them have changed accordingly. We have moved from what Levien calls the "parochial era," during which computer usage was expensive and justified for only a narrow class of numerical and clerical tasks, to what he calls "a universal era," in which computers are economically accessible for a wide class of new and previously infeasible applications.[7]

Before too long computers will be accessible to everyone. Today over 13 million people in the U.S. alone own pocket calculators, perhaps the first sign of free and easy access to computer power for all.[8] Millions more are coming into *direct* contact with computers through point-of-sale terminals located in stores, supermarkets, airline reservation counters and offtrack betting parlors; credit cards; cash machines; cash registers; digital watches; toys and games; electronic mail; and an ever-increasing number of terminals located in schools and colleges, libraries, museums, and store-front and community computing centers. It is even possible today to build your own computer with 4K memory, cassette operating system, alpha-numeric keyboard, and extended Basic for less than $500. By 1985, predicts, F.G. "Buck" Rodgers, Corporate Vice President for Marketing at IBM, one out of every 100 homes will have a computer terminal. Some people think that he is a stodgy pessimist!

It is the *accessibility* to computer power, brought about by greatly reduced costs and high demand, that is the key to its future use. Major changes accompany the use of any new medium as it passes from the hands of the few to the hands of the many, and the history of technology shows that it is next to impossible to predict when a medium is in the hands of the few how it will be used in the hands of the many. Yet the authors of *Learning Alternatives* content themselves with describing the present and simply extrapolating from today to tomorrow. They paint the picture within the frame but fail to consider the changing framework in which it's all happening.

But first for the good news. The authors do a superlative job of telling us where we are. The book is an invaluable source of information about what computer-based learning materials are available today, where to find them, and what you can do with them. The set of references (all twenty-three pages of them) and sources of information listed in the appendices alone are worth the price of the book, and the analyses offered by the authors in the first four chapters are clear and insightful.

The first two chapters, beautifully written by Hunter and Seidel, point out the diversity of purposes and activities which can be served by technology, and the interdependence of social goals, values, educational reform and technology. They provide many useful distinctions such as the difference between an educational innovation designed to be used within the current educational structure and one intended to reform it.

In the third chapter there are many examples of computer-based learning materials, arranged by discipline, an excellent guide for the person who wants to get started and needs to know where to begin. The Decision Guide offered in Appendix 4 will help him avoid the mistakes of others by answering some well formulated questions about the target situation into which the innovation will be introduced, the characteristics and purposes of the innovation, and the costs and support necessary to maintain it.

Chapter 4 presents a series of interesting case studies illustrating various approaches to the development of computer-based learning materials. Although the reader is left to draw his own conclusions the authors caution that "Whether you place your bets on creative individuals, discipline experts, systematic methodology, student or teacher involvement, powerful technology, multi-disciplinary teams, magnitude of funding, or a combination of these, *there are no guaranteed outcomes in quality or acceptance of the end product.* [emphasis added]"

Now for the bad news. In spite of this cautionary note and the statement that "it seems neither useful nor possible to seek principles or optimum strategies for development," the authors nevertheless conclude that the major problem in the development of quality materials justifying widespread adoption has been the "*lack of a coherent plan.*" They therefore advocate that a "coordinated national program" be established with "strategy coherence" based on one of five alternative models. All of the models involve massive infusions of Federal funds, curriculum by committee, and are extentions or combinations of existing approaches that certainly have neither proved themselves nor given any indication that "bigger is better."

Are there no other alternatives? Is there no place else to look for *authors* and distribution mechanisms than within the current structures which have proved so unsuccessful to date?

Perhaps looking at the past uses of computers in the classroom is not the best way to think about future uses of the computer in education. Perhaps a new viewpoint is necessary.

Let us consider for a moment some of the changes in the production of materials which accompanied another powerful technology as it moved from the hands of the few to the hands of the many, and recall some of the more notable changes that occurred when typography released the books from the hands of the scribe and the monastery into the hands of the public at large.[9] Until Gutenberg, there were no authors writing for a public. The public of Dante and Chaucer was necessarily a small group who listened to the poets reciting their verses. Reading publics, in our sense of the term, did not exist. The situation then was not unlike that of the composer before phonograph, radio and LP—the audience for new works was small and select.

The impact of having computer terminals located in homes, libraries, and places of public access, will not simply be to move the classroom to a new location but to change the audience and the conditions of use. This will happen regardless of anything that educators do or don't do. The use of pocket calculators by school children today is a case in point. It happened, and would have done so, with or without the approval of the National Council of Teachers of Mathematics.

What can we do then to assure the nationwide, beneficial use of computers in education? Let me offer a few suggestions of my own.

First, we must act now to gain the experience needed to make intelligent and considered decisions about computer use in education. When computer power becomes as freely and easily accessible as electric power is today, the question must not be "what can we afford to do," but "what do we want to do."

Investments in providing accessibility to computing power in our schools should not be viewed as an investment in machines, but rather as an investment in the experience to be gained by students, faculty and administrators. Long lead times are necessary to prepare for its use.

Second, we should take the opportunity to rethink everything that we do in education, remembering that the vast range of uses to which print has been put in education were not at all apparent on the day that Gutenberg set his first line of type. I would guess that we haven't even begun to scratch the surface in finding new and imaginative roles for the computer. We must look ahead rather than behind; we are at the start of an adventure, not the end.

Third, we must not consider the computer in isolation, but rather view it together with the other communications technologies which are shaping our lives—radio, broadcast television, cable television, home videodisc systems, telephone, satellite carriers and the rest.

And fourth, let us remember that computers alone will not bring about change. They need the efforts of imaginative people who can try out and demonstrate their ideas in viable settings, which are large enough to accurately portray the critical interactions which often fail to appear in small settings and whose absence creates misleading results.

The technology that will allow us to do whatever we want to do will soon be at hand. We must be prepared with the visions, the insights, and the experience to know what we want to do. *Learning Alternatives* serves a well-defined need. It tells us how to begin.

FOOTNOTES

1. National Academy of Sciences — National Research Council, *Digital Computer Needs in Universities and Colleges*, Washington, D.C., 1966.
2. President's Science Advisory Committee, *Computers in Higher Education*, The White House, Washington, D.C., 1967.
3. Oettinger, Anthony G. with Sema Marks, *Run, Computer, Run: The Mythology of Educational Innovation*, Harvard University Press, Cambridge, Mass., 1969.
4. Commission on Instructional Technology, *To Improve Learning: A Report to the President and the Congress of the United States*, U.S. Government Printing Office, 1970.
5. Levien, Roger E. *et. al. The Emerging Technology: Instructional Uses of the Computer in Higher Education*, McGraw-Hill Book Company, New York, 1972.
6. *The Fourth Revolution: Instructional Technology in Higher Education*, McGraw-Hill Book Company, 1972.
7. Levien, 1972, p. 3.
8. For those who consider the pocket calculator a toy, let me point out that many of the 13 million calculators now in use are equivalent in power to what John von Neumann had available to him in 1945 for the entire Manhattan Project.
9. The ideas here are stated by Marshall McLuhan with respect to the increased availability of 8mm film in the schools and appear in *8mm Sound Film and Education*, Louis Forsdale (ed.), Bureau of Publications, Teachers College, Columbia University, New York, 1962. Many of the comments on the value of accessibility as a key factor in the usefulness of an educational product were first expressed by Louis Forsdale in this volume.

*****

Edcentric: A Journal of Educational Change, published by The Center for Educational Reform, Inc., P.O. Box 10085, Eugene, Oregon, 97401, six-issue (one-year) subscription $6.00 for individuals, $10.00 for institutions.

The name says it very succinctly: *Edcentric* is a journal of educational change and criticism, relevant and up-to-date because it is published and written by people in the schools, for people in the schools. The topics covered in the magazine range from, well, anything to everything of interest to people who believe that schools should treat people as human beings. The magazine is aimed at teachers and parents and concerned persons who are interested in "working to change education and to make changes through education" and sharing their experiences with others.

Edcentric is rather unusual in that it is put together by an editorial collective, the members of which are in a constant state of flux. As a consequence, the magazine comes out somewhat irregularly (it does come out—37 issues so far, the latest on Public Alternative Schools) but it has a certain life and spirit that marks *Edcentric* as a vital force for change and understanding.

John Lees
Rolla, MO

Security, Accuracy, and Privacy in Computer Systems. James Martin, Prentice-Hall, Englewood Cliffs, N.J. 1973, $19.95

The savvy professional who wants considerable, sound technical detail computer security would do well to invest in James Martin's *Security, Accuracy, and Privacy in Computer Systems*. This book can function either as a reasonably complete course in computer security, or as a reference book on specific areas. The 600-page length of the book makes reading it more than a casual task, but the writing is well done and never gets in the way of the information.

Martin divides his tome into five well-balanced parts, each containing information applicable to a range of system types, from batch processing to teleprocessing in real-time. These major sections are:

I. Definition of the overall problem
II. Design of the computer system
III. Design of physical security
IV. Design of administrative controls
V. Design of the legal and social environment

The book also contains extensive appendices giving guidelines for the construction of storage vaults, a checklist which relates security vulnerabilities to specific sections of the book, and copies of legislation pertaining to privacy. These are not currently relevant because the Privacy Act of 1974 (H.R. 93-579) was passed after this book was issued, and because Congress has some firm proposals for legislation to apply to the private protection in computer systems.

James Martin is well-known for his expertise and prolific writing in the area of data communications which makes his advice and analysis particularly relevant to today's many remotely accessed computer systems. Too often in the past, data system protection has meant fire and flood protection. These traditional security concerns, as well as those relating to personnel management, locks, and print-out disposal are given their due. But concerns for data scrambling, communication line protection, operating system "glitches," and other areas which are more intimately involved in the technology of modern data processing systems are emphasized.

Deanna J. Dragunas
Wetumpka, AL

*****

Scelbi's Galaxy Game by Robert Findley, 1976, 172 pp. paper, $14.95. Scelbi Computer Consulting, 1322 Rear Boston Post Road, Milford, CT 06460.

Scelbi's First Book of Computer Games by Nat Wadsworth and Robert Findley, 1976, 122 pp. paper, $14.95. Scelbi Computer Consulting.

If you own a small 8008 or 8080 based system, you'll be interested in these two new computer game books by Scelbi. The games in these books aren't in Basic or any other high-level language, but instead are in assembly language and machine code. Extensive documentation is included.

The object of Galaxy is to travel throughout the galaxy in search of alien ships and destroy them in a limited number of stardates. You have two types of weapons: torpedoes, and a "phasor" (fired by the Starship Enterprise we assume). The game is pretty bony—no floating point math, very short messages, and some things which will bother Star Trek—excuse me, Galaxy purists, but remember, it all fits in 4K!

The other book, *Scelbi's First Book*, contains Space Capture, Hexpawn, and Hangman. In Space Capture, played on an 8 by 8 grid, you attempt to prevent an enemy spacecraft from moving by destroying all the sectors around him with your "phasor." Hexpawn appears to be very much like the Hexpawn game described by R.R. Wier in *Byte*, while Hangman is modeled on the popular word game.

Although the messages contained in these games are brief and a little too cute, (example: !#0# DARN! YOU HAVE ME CAPTURED!!) you can change them easily. Since each program is thoroughly explained, it might be interesting to add some of your own features if you have enough memory.

Steve North
Newfoundland, NJ

Sorting and Sort Systems, Harold Lorin. Addison-Wesley, 1975. 460 pp., $16.95.

This book is for experienced programmers who are interested in developing or using a sort. It would also be useful to non-programmers who need to understand the behavior of Sorting Systems, since it is one of the most readable texts on sorting in the literature with the details of programming left to the appendices. The specific sorting methods are covered broadly and carefully while recognizing that the ultimate performance of algorithms depends on the computer in use. Thus the reader comes away with a good intuitive feel for sorting and some confidence that, if there were not a good sort available in his subroutine library, he could knock off a respectable one in a few hours.

I feel that sorting should be one of the first things taught fledgling programmers, even in science-oriented schools or departments. Besides its practical value, it is an interesting mathematical subject. (No one yet knows how to make a SHELLSORT behave optimally. And who would expect that the fastest known procedure for reliably putting order into a sequence of values involves the generation of *random* numbers?)

The only criticism of the present book is that there are no exercises. In spite of that, the book ought to be available to any teacher of programming, if only because it is stimulating and pleasurable reading.

Not the least attractive feature of the book is the inclusion of all the ACM sorting algorithms (in ALGOL) as well as several related PL/I programs.

L. D. Yarbrough
Lexington, MA

*****

Mechanics. Herbert D. Peckham, Student Lab Book, 32pp, $1. Teacher's Advisor, 40pp, $1. Hewlett-Packard Co., Cupertino, Calif., 1972.

These booklets, in the HP Computer Curriculum Series, are intended to help meet the need for computer-oriented problems in physics by providing students an opportunity to use the computer as a problem solving tool within a particular subject matter area. Specifically, this unit is intended as an "enrichment" experience in the field of mechanics.

There are seven topics: rates, displacements, Newton's second law, half step method, the harmonic oscillator, more complicated forces, and orbital motion. In each section, there is a little preparatory explanation, an initial supplied program, and several exercises where the student is asked to modify the given program to extend or generalize the results. Each section also provides at least one advanced programming exercise where the student is asked to construct a separate but related program from scratch. The Teacher's Advisor provides program listings and sample runs. With only minor adjustments, these programs should be adaptable to different computers.

The author assumes that the student (1) has had algebra and some trigonometry, (2) that he already knows how to write a simple computer program in BASIC, and (3) that he will have access to a computer for at least two hours per week.

Perhaps the most thorny problem in teaching introductory physics is that the student does not have the requisite tools of calculus at his disposal. This booklet serves as an example of how the computer can help overcome the difficulty. Although the words "calculus," "derivative," and "integration" are never used, nearly all of the exercises here involve calculus by means of difference approximations. The student who uses this material will accordingly have a much better feeling for calculus when he or she studies it analytically.

The computer enriches the study in other ways. Using just one program, the student can perform many different experiments by making slight modifications to the basic program, and make generalizations on the basis of the results.

The exercises are sequenced so that the student applies what he or she has learned in the previous problem to solving the next one. This feature provides a strong thread of continuity and makes the treatment highly attractive. On the other hand, because it does build on previous results and is cumulative, one cannot omit any section and pick up a later section.

Although the exercises, in a "watch me and then imitate, modify, and extend" format begin slowly, they quickly become challenging in terms of programming skills required. The first two sections on rates and displacements involve concepts and problems that are simple and straightforward. In section 3, on Newton's second law, the programming begins to get a little more involved. Perhaps the author discovered this, for he has added an appendix where he looks at the section 3 initial program in more detail. There is a flow chart and a line-by-line description of what is happening in the program. The reviewer agrees that this would be a good time to nail down a few key programming skills by looking at a fairly central program. The time would be well spent since subsequent exercises are somewhat more involved. An average high school physics class in my area would find these later exercises moderately difficult. Indeed, the author assumes that "students taking "introductory physics (in college?) will be quite capable as a group."

This unit could be used with a uniformly good high school physics class, but as the instructor, I would want to provide some additional instruction on the concepts as well as an occasional review of programming principles and techniques. It would also be useful as a directed individual study project for a bright high school senior with some programming experience and an available terminal.

With regard to the pamphlet's printed format, it would be helpful if at least the statements of the exercises could be reprinted in the Teacher's Advisor so that the teacher could relate the solution commentary to the problems, and not have to refer back and forth between manuals.

Daniel S. Yates
Glen Allen, VA

*****

User's Guide to Computer Crime, Its Commission, Detection and Prevention. Stephen W. Leibholz and Louis D. Wilson, Chilton Book Company, Radnor, Pa. 1974. $9.95

This book comes complete with disclaimer that all the cases of theft, fraud, sabotage, and espionage conducted via computer as described in the book have actually been comitted *and discovered.* However, the implication to the casual reader is clear: There must be more clever people who have not yet been discovered!

Were it not for the fact that the technology is mercifully somewhat sketchy and general in places, the *User's Guide to Computer Crime* could be labelled as hazardous to the health of computers. As it stands, it could be a generalized cookbook for how to make money, or at least mischief, by manipulating the computer.

Those who are already deeply involved in computer security will find nothing new here. But those who had labored under the mistaken notion that because the computer is complex, it is also safe, can profit from it. The incidents of unauthorized computer manipulation make some of our flights of fancy seem positively earthbound. The book attempts to move the reader who may be responsible for a data processing system from the edge of immediate panic to a more positive predisposition to do something, perhaps using some of the valuable suggestions contained in the latter portions of the book.

The book has four major divisions: basic problems and case histories of various "crimes," specific measures and general operating principles to achieve reasonable protection from incidents, methods of auditing and detecting unusual activity along with some legal discussion of computer use and abuse, and recommendations for user programs of action in accounting and legislative realms. The appendices are as revelatory as the book itself: A checklist for ensuring data processing operations security, a reprint of the Fair Credit Reporting Act, and even the reprint of the first of 105 counts in the Equity Funding indictment, one of the largest computer-assisted heists in history (so far), and the Stanford Research Institute survey of programmer ethics. In the latter, of the 55 managers and programmers surveyed, 13 thought it was "okay" to attempt to log onto a time-sharing service for which they were not authorized users. 10 had actually done it. Only 17 thought it was illegal.

Not only is the book worthwhile for anyone who has had cause to manage, use or mistrust computers, but it is written in an easy, readable style.

Deanna J. Dragunas
Wetumpka, AL

Electronic Computers. S. H. Hollingdale and G. C. Tootill. Penguin Books (A Pelican Original), Middlesex, England. Published 1965 and revised in 1975. 378 pages, illustrated, indexed, 4½ x 7, £1 ($3.95 in the United States and Canada), paperback.

The publisher's notes on the back cover describe this book as being intended for "the general reader." "[The authors] have taken particular care with the specialist jargon of their subject, explaining each term as it occurs."

A "general reader" had better equip himself or herself with a solid background in mathematics and electronics if that general reader wants to wade through some parts of the book. A certain technical patois in a work of this sort is probably unavoidable, but it is irresponsible if not actually misleading of the publisher to represent the book as a general work when it is in fact no such thing.

For example, it is hardly helpful to offer, as the authors do in a footnote on page 107, a twenty-five word definition of a derivative. A reader who doesn't already know what a derivative is will not understand either the definition or the second order differential equations sprinkled across the page in the solution of a multi-storied building stress analysis problem. And no reader who doesn't know *something* about electronics is likely to understand the wiring diagram on the next page of the analog computer which is supposed to work out the solution.

The authors are both mathematicians who have been involved in the British computer industry almost from its beginnings after World War II. The writing is clear, although a bit leaden in the inimitable style of British technical writing. The book's emphasis is on the British computer industry and British computer users, with little said of what is happening on the other shore of the Atlantic. The bias is somewhat strange in light of the almost total American domination of the field.

There are twelve chapters. The first two present a detailed history of computing, beginning with the abacus. It has always been a mystery to me why the decimal notation was not used in ancient times by the same peoples who used the abacus so skillfully. Nothing seems to be more natural than the transition from the positional decimal notation of the abacus to a similar system for writing the same numbers on paper or papyrus or cuneiform. Yet the ancient Greeks, who were first rate mathematical heavyweights, never made the connection and used a cumbersome alphabet-based system for transcribing numbers. Hollingdale and Tootill offer the explanation that since arithmetic was always done on an abacus and never on "scratch papyrus," the awkwardness of the method used to write down numbers was never really a problem. I find this explanation clever but not quite satisfying.

In fact the Babylonians did use a positional notation with a radix of sixty but without a zero, so 1, 60, 3600 and 1/60 were all written the same way and the correct figure had to be deduced from the context. The elusive zero wasn't invented until many centuries later by the Hindus, even though it was right there on the abacus all the time.

Of course we can always ask questions of the type — If the Greeks were really so smart, why didn't they invent the cheeseburger?" In retrospect many brilliant ideas seem to be so simple that it is difficult to understand why no one thought of them before. Hindsight is one of the most exact of the sciences.

In 1946 the American Army in Japan staged a competition between a Japanese version of the abacus, known as a soroban and the most modern electric (not electronic — the word wasn't even in the dictionary then) calculator. To quote *Stars and Stripes*, "the abacus victory was complete." So ended an era.

The emphasis in the book is on hardware. There are numerous diagrams of the "insides" of various units, including a differential analyzer! Not much space is devoted to minicomputers, and the little there is on the subject of microminiaturization seems to have been added as an afterthought when the book was last revised in 1975. This last comment is not offered as a criticism, since no one in his right mind could have foreseen the Intel 8080, but the sparse treatment does date the book.

In fact the 1975 revision appears not to have been especially extensive. In several instances the authors mention the technique of punching out intermediate results on cards for use in later steps of the calculation, as though tapes and disks had not yet been invented. Nevertheless, the workings of tape units are explained in chapter seven and almost the whole of chapter twelve is devoted to a very detailed description of disk drives.

Although there is a chapter on software, it is mostly devoted to a description of programming languages, especially ALGOL. There is too little said about operating systems or time sharing, and nothing at all of interpreters or of structured programming.

The effect of computers on society is not treated at all, except in an aside in which the authors mention "effective ways of using [computers] — for good or ill."

To sum up, the book is clearly written and informative, although hardly as up to date or inclusive as it might be. It is also too technical for the general reader, but it can be useful to the reader with a good background in computers who wants to fill in some gaps in his or her knowledge, particularly about early hardware.

A final point — if one British pound is worth* about two American dollars, why does this book cost £1 in Britain and $3.95 in the United States?

Alex Ragen
Jerusalem, Israel

* *When this review was written, matters changed rather seriously soon after.*

*****

Configurations (Game). Harold L. Dorwart. Wiff 'N' Proof Learning Games Association, 1490 South Boulevard, Ann Arbor, MI 48104. $6.75.

After playing the game *Configurations* a student should have: 1) The realization of the existence of finite projective geometries, 2) An aroused interest in finite projective geometries, 3) A better understanding of some number patterns.

Configurations contains a series of geometric puzzles, as games of solitaire and discovery, based on finite projective geometries. The geometries used are the Fano 7_3, the Mobius-Kantor 8_3, the Pappus 9_3, the Desargues 10_3 and others. Each of the games is played on a game board with small plastic numerals provided in the game kit. By placing the numerals on the boards according to the rules found in the instruction manual, the player is led to very interesting mathematical discoveries.

Configurations would be a worthwhile addition to any Math Lab or Math Resource Center to be used by students working on independent projects or by those who are interested in a fun way to acquire new knowledge.

Peter B. Danos
W. Redding, Conn.

*****

Software Tools, Kernighan, B.W. and Plauger, P.J., Addison-Wesley Publ. Co., 1976, $8.95.

To teachers of programming: Stop what you are now doing; get yourself a copy of this book and study it. Then give some thought to what you have been teaching and what you should be teaching about programming. You will find in the book a lot of good ideas and a wealth of well-written, useful examples that you can use to explain to your students how to write clear, effective, reliable, understandable programs. If you get caught up by the ideas in the book you can send $25 to Ms. Linda Banks at Addison-Wesley and she will send you a 9-track 800 BPI magnetic tape containing all the programs in this book (over 200 in all) for you and your students to use and study.

To students of programming: If your teacher is too slow in responding to the above, get your own copy of this book and study it. Shortly you will be programming rings around your classmates and your teacher, and you will be preparing yourself to take your place among a new wave of professional computer scientists who know, for a change, what they are doing.

To professional programmers: Get on the ball; the new wave is coming.

To the authors and publishers of software tools: I am adding your book to a short list of things I wish I had done, along with the pocket calculator, the cassette tape recorder, and a few other goodies. This is the best programming book published in the last 3 years, at least. Thank you.

L. D. Yarbrough
Lexington, MA

TTL Cookbook by Donald E. Lancaster. $10.75 Howard W. Sams, Indianapolis, Indiana, 46268. 1974.

People who use computers often get sufficiently involved that a knowledge of hardware becomes useful, whether to understand better how the computer "does it" internally, to maintain a computer system, or to add that special gadget as an input or output device. One needs several sorts of information resources, ranging from catalogs of logic devices (integrated circuits), hardware catalogs, explanations of hardware functions, etc. One can compile an impressive array of logic sheets and catalogs and still not have any guide to putting it all together.

The *TTL Cookbook* does a good job of combining a range of factual information about the logic devices, the folklore and accepted good practice of their use, and a range of techniques and suggestions for projects. It is based on the premise that the TTL logic line has reached its maturity; a wide range of inexpensive functional parts is available and can be usefully combined and applied by persons without much training in electronics. The point is made forcefully that the building-block approach has removed the requirement for many of the standard logic design techniques that were essential a few years ago.

The cookbook analogy implies that the reader can sample a recipe for a particular function (gating, timing, storage, counting, display) or find more general information on the proper preparation of logic systems using TTL devices. This background comes in the first chapter and assumes some acquaintance with the tools of electronic servicing and design such as oscilloscopes, power supplies, decoupling, resonance, inductance, etc. Although necessary before one undertakes a serious design, the detail can be skimmed by the novice if he promises to return later. The second chapter is a catalog of 77 frequently used members of the TTL family, with part numbers, functional names, pin designation diagrams, functional descriptions, logic delays and current requirements. The descriptions also contain cautionary notes about features that might cause problems for the naive user, which will help avoid some of the common problems that sometimes haunt new designs.

A discussion of logic as applied to the TTL line follows in Chapter 3. Then come chapters devoted to gates and timer circuits, flip-flops, counters, shift registers and rate multipliers. Each section discusses several applications ranging from the realization of logic functions to the design of fun projects. Recurring examples include displays using light emitting diodes, timers and music generators. Most examples are relatively complete, although toward the end of the book several are presented in a relatively bare sketch of the technique. The discussion of the television time display and the TV typewriter are left at the high level block diagram stage. These and many other sketches of systems use of the applications detailed in previous chapters are found in the last chapter, entitled "Getting It All Together," designed to challenge the reader to do some design thinking himself. A list of simpler project suggestions, still requiring synthesis, is given at the end and might serve as stimuli for science-fair type activities.

The TTL parts described are readily available, for the most part, and are often available on the surplus market at tremendous reductions in cost. There are a few exceptions which may be aggravated by the particular distributers one uses. The use of a timing device designated as the 555 was new to me; searching for it was aggravated by the fact that manufacturers are mentioned inconsistently. In this case, try Signetics or Intersil NE555 or SE555, about $1.75. Some other types given are not necessarily manufactured by all of the major semiconductor houses, and certainly the cost of some of the devices used will exceed the implied low price unless one has ready access to the surplus market. Some indication of relative pricing would be helpful in the description of parts; the "as low as 30¢ per package" can rise to $10 for some of the circuits. Returning to the analogy implied by the title, a knowledge of basic prices, substitutions and willingness to shop around for good buys, may be as necessary in the logic lab as it is in the kitchen these days.

Robert S. McLean
Toronto, Ontario

Queries 'N Theories (Game), by Layman E. Allen, Peter Kugel and Joan Ross, $9.75. Wff 'N Proof, 1490 South Blvd., Ann Arbor, MI 48104.

QUERIES 'N THEORIES is a game (or series of games) which can be used for several purposes. It incorporates ideas from modern theories of linguistics. The play simulates some important features of the scientific process. The authors state that it is "designed to develop a basic and uniquely human skill: asking good questions." The instruction manual discusses briefly the relationship of the game to both linguistics and the scientific method.

The game is designed for three or more players; there is a variation for two players described in the manual.

One player, the "Native", formulates a "Language" which consists of a set of rules for producing ordered strings of colored chips (Sentences). The remaining players, called "Querist-Theorists" or "QT's", attempt to understand the Language by asking whether specific strings of chips are Sentences of the Language (these yes-or-no answer questions are the "Queries" of the title). When a QT thinks he has a correct theory of the Language based on the Native's responses to the Queries, he so states and becomes the Linguist. Querying then stops and the remaining QT's affiliate with the Linguist if they think they have enough information to understand the Language; they affiliate with the Native if they believe there is not yet enough information available. The Linguist and his affiliates are each tested by asking them to predict whether or not strings constructed by the Native and his affiliates are Sentences in the Language. Payoffs are based on the complexity level of the Language and the number of Queries asked before testing begins. The Linguist wins if he gets all the Native's test questions correct; the fewer the number of Queries that were asked initially, the more points he gets from the Native. The Native wins if the Linguist does not predict all the test questions correctly; the more Queries, the greater the number of points. Side payoffs are also made involving the Native-affiliates and the Linguist-affiliates. The scoring system is complex; charts for assisting the calculations are included in the game.

This game is a very good simulation of the scientific process and students who work at it should be able to improve their "strong inference" abilities (see Platt, J. R., *Science, 146,* 347 (1964)). In my experience, however, the game is not easily learned, and in an initial 2 to 3 hour session, most students are only able to grasp the basic rudiments of the game. The instruction manual is itself over 50 pages long; about half of it is devoted to presenting the basic ideas and sample games in a programmed learning format. For greatest learning value, several playing sessions should be scheduled. As players gain experience, the complexity of the Language can be increased. There are also mechanisms for asking compound ("Strong") Queries. These features provide additional richness, so that players' interest and, therefore, learning continue as they play more games. I have worked mostly with a computerized version (unpublished) in which the computer plays the role of the Native. With it, student interest is somewhat unevenly distributed: Most students are either mildly interested or totally disinterested in the game; a smaller number do get hooked, sometimes but not always developing into good players. The game does require thinking; people who don't like to work at thinking won't enjoy it.

I would recommend the game to teachers who are interested in helping students develop their analytical abilities and to anyone who enjoys problem solving. It should be a useful supplement to any introductory course in the sciences at the high school or college level that includes a discussion of the scientific process. I have not used it in a linguistic context and thus cannot comment on its usefulness in this area.

Richard A. Cellarius
The Evergreen State College
Olympia, WA 98505

CREATIVE COMPUTING Reviews

SPACE:1999

By Ruth Glick
Columbia, Maryland

As a substitute for *Star Trek* reruns, this season's non-network science fiction series called *Space 1999* just doesn't make the grade

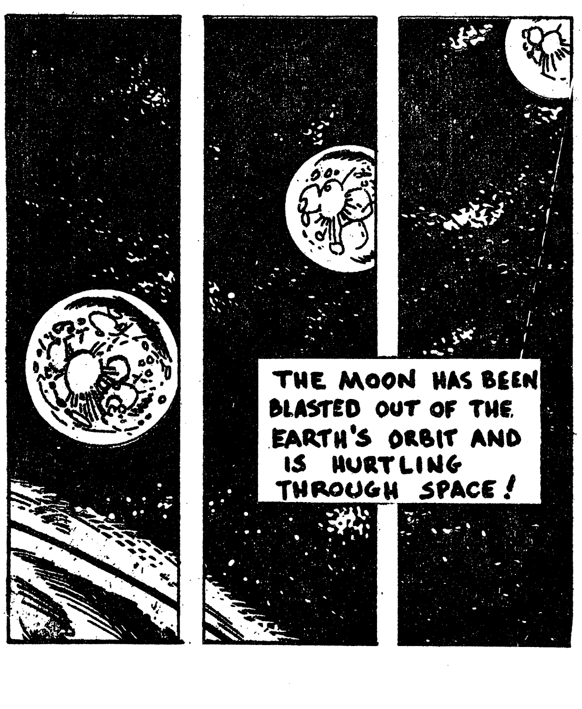
How come? Hasn't it got spectacular special effects like the moon blasting out of orbit, plenty of action, and Moon Base Alpha uniforms designed by Rudi Gernreich?

True. But these are only superficialities—the external trappings of T.V. science fiction. Unfortunately, *Space 1999* also has infantile plots, boring characters, ridiculous motivation, poor story construction, gadgets for their own sake and a whole wax museum full of unexplained and unbelievable monsters.

Comparing the new show directly to a successful series like *Star Trek* is a good way to illustrate the problems. Let's look at a memorable *Star Trek* monster—the horta—for example. It wasn't just a mindless beast menacing a group of miners. During the show, its intelligence was established. And later it was given believable motivation—that of a mother defending her young.

On the other hand, *Space 1999's* second episode treated the viewer to an old-fashioned horror show featuring an oversized hydra that sucked in human victims and spit out their charred remains. But no attempt was made to explain what the monster was, where it came from or why. Similar objections can be raised to the other monsters *Space 1999* parades on the T.V. screen almost every week. Each is simply a *deus ex machina* brought in to foster a particular story line.

Lively characterization is another missing element in *Space 1999*. One of *Star Trek's* biggest appeals was the crew of the star ship Enterprise. Spock, Captain Kirk, Dr. McCoy, Uhura and the rest still have a passionate following today because of their well-defined, complicated personalities.



The staff of Moon Base Alpha also comes in an international assortment of colors. But there's no one the viewer really wants to root for. The female lead, played by Barbara Bain, is colorless and wooden. And the other cardboard characters, with the exception of the base's Commander Koenig, are like interchangeable pawns moved about by the plot.

Koenig himself is another problem. Unlike *Star Trek's* Captain Kirk, who was allowed some human fallibility, he's never wrong. And his dependable infallibility makes it possible for the viewer to figure out very quickly where the plot of any given *Space 1999* episode is leading.

Of course, being able to unravel the plot quickly is a definite handicap, since the stories on *Space 1999* usually drag—with long slow scenes that barely advance the action. Take the space expedition at the end of which that over-sized hydra was lurking. For 10 minutes there was no action. To make the sequence even slower, it was told as a flashback, unaccountably narrated by Barbara Bain, who wasn't even present during the expedition.

Repetition of plot devices is another deficiency. One week Alpha is being menaced by a blinking blue light. The next week it's flashing a green light. And, in every episode so far the script dredges up the danger of radiation leakage.

It's possible to point out a lot more problems—especially with the show's hardware. First, there are those ridiculous little gadgets that the base personnel use to open doors. You'd think that man would have invented something better than a glorified key by the year 1999. Imagine if one dropped off someone's belt and he were trapped in a room for days before being missed.

Or what about the use of computers in *Space 1999?* Maybe they don't have the technology to match *Star Trek's* "library computer." But they should be able to do better than a machine that prints out its hard copy on what looks like adding machine tape.

And then there are the hand weapons—which are almost identical to present-day staple guns—except that they shoot energy rays instead of staples.

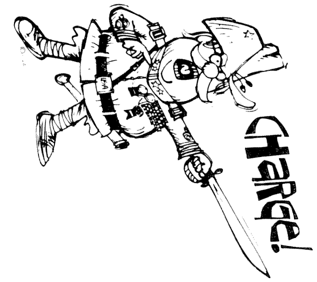
It's obvious that far less thought went into designing believable equipment for Moon Base Alpha than for the star ship Enterprise.

However, *Space 1999* does have it all over *Star Trek* in the special effects department. Their moon landings, space-suited lunar surface sequences, and atomic explosions are unbelievably good for a T.V. production.

But they don't have the scientific accuracy to match the effects. Isaac Asimov took about 1500 words pointing out these problems in a *New York Times* review of the show more than a month ago. His biggest quarrels were with the number of planets Moon Base Alpha has encountered in such a short time and with the show's inaccurate references to a "dark side of the moon."

With a highly successful model like *Star Trek* to crib from, the producers of *Space 1999* should have come up with some satisfying science fiction drama. Too bad that what they have is a bunch of spectacular special effects in search of a decent story line and a few interesting characters.

Books and Merchandise



Subscriptions to CREATIVE COMPUTING

creative computing

Books, Merchandise & Subscriptions

Quan	Cat	Descriptions	Price

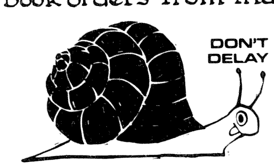
shipping charge $1. USA $2. Foreign ________

NJ residents add 5% sales tax ________

TOTAL ________

Name ________
Address ________
City ________ State ________ Zip ________

☐ Cash, check or M.O. enclosed
☐ Visa/BankAmericard ☐ Master Charge
Card no. ________ Exp. ________
☐ Please bill me ($1.00 billing fee will be added)
Book orders from individuals must be prepaid.



For faster service call
toll free 800 631-8112
In NJ call (201) 540-0445

Allow 8 weeks delivery on all books and subscriptions

creative computing

Books, Merchandise & Subscriptions

Quan	Cat	Descriptions	Price

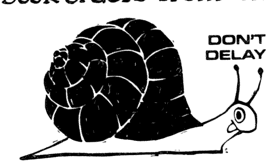
shipping charge $1. USA $2. Foreign ________

NJ residents add 5% sales tax ________

TOTAL ________

Name ________
Address ________
City ________ State ________ Zip ________

☐ Cash, check or M.O. enclosed
☐ Visa/BankAmericard ☐ Master Charge
Card no. ________ Exp. ________
☐ Please bill me ($1.00 billing fee will be added)
Book orders from individuals must be prepaid.



For faster service call
toll free 800 631-8112
In NJ call (201) 540-0445

Allow 8 weeks delivery on all books and subscriptions

creative computing

Subscriptions

☐ New ☐ Renewal

TYPE OF SUBSCRIPTION		U.S.	Foreign Surface	Foreign Air
Institutional	1-yr.	☐ $15.	☐ $20.	☐ $28.
	3-yr.	☐ 40.	☐ 52.	☐ 68.
Individual	1-yr.	☐ 8.	☐ 12.	☐ 20.
	2-yr.	☐ 15.	☐ 23.	☐ 39.
	3-yr.	☐ 21.	☐ 33.	☐ 57.
	Lifetime	☐ 300.	☐ 400.	☐ 600.

YOUR NAME AND ADDRESS:
Name ________
Address ________
City ________ State ________ Zip ________

PAYMENT INFORMATION
☐ Cash, check or M.O. enclosed
☐ Visa/BankAmericard ☐ Master Charge
Card no. ________ Exp. ________
☐ Please bill me ($1.00 billing fee will be added)
Book orders from individuals must be prepaid.

For first issue allow 8 weeks

Thank You

creative computing

Place
Stamp
Here

creative computing
P.O. Box 789-M
Morristown, New Jersey 07960

Attn: Kay

Place
Stamp
Here

creative computing
P.O. Box 789-M
Morristown, New Jersey 07960

Attn: Kay

Place
Stamp
Here

creative computing
P.O. Box 789-M
Morristown, New Jersey 07960

Attn: Kay

$8.95

THE BEST OF creative computing

VOL. 2 EDITED BY DAVID AHL

ISBN 0-916688-03