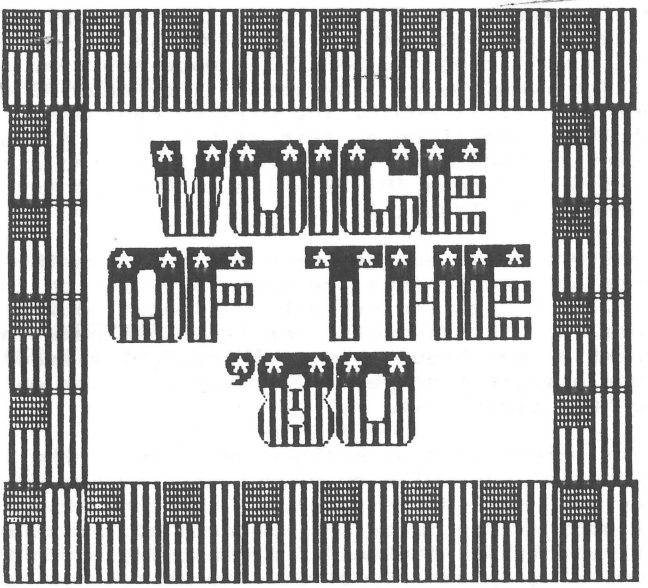
NEXT MEETING JULY 2, 1985

7:30 PM DETAILS ON PAGE 19



JULY 1985

VOLUME 6 NUMBER 2

HOW WOULD YOU DO IN A COMPUTER COMPETITION?

By Robert A. Fowkes

The recently concluded Third Annual Tri-County (Westchester, Rockland and Putnam,NY) Computer Competition, of which I am the chairman, contained a variety of BASIC problems on three different levels. The students who represented finalists from 35 separate school districts competed in an elementary division (Grades 4-6) <easy>, a middle school division (Grades 7 and 8) <medium> and a senior division (Grades 9-12) <hard>.

The competition was structured so that each contestant had a total of two hours to complete the five problems on their choice of computer (a TRS-80 Model III, an Apple II+, an Apple IIe, a Commodore 64, or an IBM PC.) The students were limited to utilizing the BASIC language resident within the machine (no versions of Disk Basic). For the TRS entrants this meant Level II BASIC. The primary reason for this was to make the competition as fair as possible. (Imagine a contestant trying to solve a problem involving a sort having access to CMD"O" on a TRS-80 while another contestant floundered in Applesoft!).

The questions, when solved, were judged on whether or not they were able to perform the task asked for on the computer. Each question was either right (1 point) or wrong (0 points) and the students were told that once the judging of a question was begun they would not get a second chance. Therefore a maximum score of 5 points would be obtainable in the two hour time limit.

With this information as background I would like to offer the readers of the VOICE an opportunity to see how they stack up against today's students. For the next five months we will publish one question from each of the three competition classifications. The following month the solutions will appear along with the next set of questions. Remember, you are not to call any disk routines in solving the problems.(Use the equivalent of Level II BASIC on whatever machine you work with.) Also keep in mind that the entire set of five questions per division was presented and successfully completed by contestants well within the two hour time frame.

Good luck and happy programming!

Question #1: <easy>

A positive integer greater than 1 is said to be "prime" if its only divisors are the number 1 and itself. For example, the integer 7 is prime. So are 5, 11 and 23. The number 15 is not prime because 3 and 5 are even divisors of 15, in addition to the numbers 1 and 15.

Write a program to determine if a number entered as input is a prime number. Structure your program so that it behaves in a similar fashion to the sample runs below. Also, proper spacing in the output must be observed. For example: "THE NUMBER21IS NOT A PRIME" would be marked wrong.

NOTE: The number 1 is NOT defined as a prime number and your program should allow for this fact.

SAMPLE RUNS

```
RUN

ENTER A NUMBER? 21
THE NUMBER 21 IS NOT A PRIME.

RUN

ENTER A NUMBER? 13
THE NUMBER 13 IS A PRIME.

RUN

ENTER A NUMBER? 1
THE NUMBER 1 IS NOT A PRIME.

RUN

ENTER A NUMBER? 571
THE NUMBER 571 IS A PRIME.
```

Question #1: <medium>

Write a program that accepts a three digit number and then outputs the sum of the digits. The program should only accept three digit numbers. If an attempt is made to input something other than a three digit integer the error message "THAT IS NOT A 3 DIGIT INTEGER" should appear and then ask for the number again. NOTE: No other error message should appear or the question will be judged as being incorrect.

SAMPLE RUNS

```
RUN

ENTER A 3 DIGIT INTEGER? 456
THE SUM OF THE DIGITS IS 15

RUN

ENTER A 3 DIGIT INTEGER? 3456
THAT IS NOT A 3 DIGIT INTEGER
ENTER A 3 DIGIT INTEGER? 12
THAT IS NOT A 3 DIGIT INTEGER
ENTER A 3 DIGIT INTEGER? 123
```

```
THE SUM OF THE DIGITS IS 6

RUN

ENTER A 3 DIGIT INTEGER? 1.23
THAT IS NOT A 3 DIGIT INTEGER
ENTER A 3 DIGIT INTEGER? 777
THE SUM OF THE DIGITS IS 21

RUN

ENTER A 3 DIGIT INTEGER? ØØ7
THE SUM OF THE DIGITS IS 7

RUN
ENTER A 3 DIGIT INTEGER? AAA
THAT IS NOT A 3 DIGIT INTEGER
ENTER A 3 DIGIT INTEGER? 9ØA
THAT IS NOT A 3 DIGIT INTEGER
ENTER A 3 DIGIT INTEGER? 999
THE SUM OF THE DIGITS IS 27
```

Question #1: <hard>

A palindrome is a word or an expression that reads the same backwards or forwards. Some examples of palindromes are:

(1) NOON
(2) ABLE WAS I - ERE I SAW ELBA
(3) NORMA IS AS SELFLESS AS I AM RON

Notice that spaces and punctuation do NOT count when determining whether or not a phrase is a valid palindrome, only the letters themselves (see example #3).

Write a program to determine if an input string is a valid palindrome. Ask for the phrase, and then output the phrase and a message as to whether or not it is a palindrome as in the sample runs below.

NOTES: (1) For purposes of this question NO phrases will be input that include commas. Therefore your program does not have to allow for this.

(2) You may assume that the judge will only input UPPERCASE letters.

(3) You must observe proper spacing in your output. For example: "OTTOIS A PALINDROME" would be not allowed.

SAMPLE RUNS

```
RUN

PLEASE INPUT A PHRASE? WAS IT A BAR OR A BAT I SAW
WAS IT A BAR OR A BAT I SAW IS A PALINDROME.
```

RUN

```
PLEASE INPUT A PHRASE? DENNIS AND EDNA SIGNED
DENNIS AND EDNA SIGNED IS NOT A PALINDROME.
```

The Mailbag

Date: Sat May 11, 1985 10:54 am EST
From: David R. McGlumphy / MCI ID: 181-7759

TO: * Alan B. Abrahamson / MCI ID: 195-8470
Subject: newsletter article

Dave McGlumphy 02/03/85
4429 Paula Ln.
Chattanooga, Tn. 37415

WILL "REACH OUT AMERICA" SAVE ME $$$?

One of AT&T's new marketing strategies is called "REACH OUT AMERICA". The idea is that your long distance bill is determined by the amount of time that you talk rather than the distance or frequency of your calls. To that end, you'll be billed a flat rate of ten dollars ($10.00) for the first hour of service. Anything over that hour is prorated at eight dollars and seventy-five cents ($8.75) per hour which means that if you talk for a total of an hour and ten minutes, you'll be charged ten dollars for the first hour and 8.75 times 10 divided by 60 (8.75 * 10 / 60) for the remaining ten minutes - totalling eleven dollars and forty-six cents ($11.46). It doesn't matter whether you make one long call or a whole bunch of short calls. Normally, you're billed more for the first minute (or three minutes?) of a call than for the rest of the time, but not under this plan. That's a plus for the plan.

There's a bonus plan too. The regular plan covers only night and weekend time slots. Since I don't like to call people after eleven o'clock at night, the plan doesn't really fit me very well, but the bonus plan does. It gives a fifteen percent discount for calls made during the evening hours for only a dollar and a half more per month than the regular plan.

That's all well and fine, but how do I know whether or not I'll save money by using the service? My program tells me. It's based on the following information graciously supplied to me by Mr. S. R. Willcoxon, Executive Vice President of AT&T.

"The following formula can be used to determine the economies of Reach Out America against your current long distance bill. Keep in mind that simple changes in your existing calls (shift a call from day to evening; evening to weekend; etc.) or increasing the numbers of calls you make can mean a big difference in this evaluation.

STEP #1: Add up the TOTAL of all charges EXCLUDING evening (E/EM), night (N/NM) and weekend direct-dialed interstate calls.

STEP #2: Add up the total number of MINUTES for all INTERSTATE direct-dialed calls made during the night/weekend period (N/NM).

STEP #3: Add up only the total AMOUNT charged for all INTERSTATE direct-dialed calls made during the evening period (E/EM).

	Basic	Bonus
TOTAL LONG DISTANCE BILL	(A)	(A)
ENTER TOTAL CHARGES FROM STEP #1	(B)	(B)
FLAT RATE: Initial Hr. Rate per mth	$10.00 (C)	$11.50 (C)
TOTAL MINUTES (FROM STEP #2) - 60 x8.75 / 60	(D)	(D)
TOTAL AMOUNT OF EVENING CALLS FROM STEP #3 Basic - Enter Total Bonus - Total x.85	(E)	(E)
TOTAL COST WITH REACH OUT AMERICA (ADD ITEMS B THROUGH E)	(F)	(F)
AMOUNT SAVED - (A - F)		

And so to the program. I deliberately coded it using elementary BASIC instructions so that it can be modified easily for use on other computers. For example, I didn't use the PRINT USING function. That makes for a strange looking output for some negative decimal values because of the vagaries of expressing some negative fractions in binary. So if you see -.100002, understand that it means minus ten cents. I didn't even use the ELSE function of the IF statement because some versions of BASIC don't recognize ELSE. I used primarily numeric input to avoid uppercase / lowercase problems and because

numeric-only input is easier for people who are accustomed to using numeric keypads. You can delete any or all REMarks without affecting the operation of the program, but don't delete a whole line just because it has a REMark on it.

Here's a brief idea of how the program flows. The first screen is an introduction. The second screen asks how much the total long distance bill is. The competitor whose figures I tested put the total of the calls on the same page as the calls, but they put a discount figure for more than an hour's calls and the actual amount I had to pay on another page. The third series of screens is for entering the type, duration, and charge for calls. There are essentially three types of calls to deal with: daytime, evening, and nighttime. The program does not really need to know the duration of the daytime calls, but it is easier to type in the minutes for all categories than to try to remember to skip the minutes for daytime calls. The fourth screen tells how much money, if any, the Reach Out America plan or the Bonus plan can save you as compared to the alternative long distance company. The fourth screen also gives you the opportunity to use the fifth screen to correct or add entries that you've already typed in.

So would I save by using AT&T's plan(s)? Yes and no. Last month, my family talked for 4 day-minutes, 105 evening-minutes, and 85 night-minutes. The regular plan would have cost me ten cents more than I had to pay to the competiton. Under the bonus plan, however, I would have saved one dollar and eighty-nine cents. We're not talking big bucks here, are we! What about competition ads that claim to save 10, 20, up to 40 per cent? Not this way, they don't. I'll have to check another month's figures to get a more representative idea, but if the pattern holds, AT&T might have a better thing going here than first met my eye.

A warning - if you make less than sixty minutes of long distance interstate calls in a month, you will take a comparatively whopping loss because you will be paying for service that you did not even use. That could eat up many months' worth of small savings.

More food for thought. Even if I "lose" ten cents under the regular plan, I think it's worth it to switch back to Ma Bell. There have been more than a few times when my calls didn't go through correctly when I was using the competition's system. I'm not saying whether or not it was their equipment, but I will say that I didn't have the problems using Ma's hardware. Even if there's a facility to get credit for bad calls, I'd rather avoid the problem altogether.

The parting shot: there's a move underway to get you to declare who your preferred long distance company is so that when you dial 1 for long distance, the local phone company will automatically route your call to the long distance company you selected. Some of the long distance companies are sending out cards for you to fill out to pick

them as your preferred long distance company. Why send in the first one you get until after you have checked out your potential savings and service? I sent my card in too soon because the competition was offering one hour of free service as an inducement and because I didn't know how to figure my costs, but after checking out my potential savings for another month or two, I may very well switch back even though it will cost me five dollars. You get to declare your preference for free, but it costs you five dollars each time you change your mind.

REACHOUT/BAS

```
1 REM THIS IS PROBABLY NOT WORTH TYPING IN.  IF YOU CAN GET
IT
2 REM FROM YOUR CLUB'S EDITOR ON DISKETTE, OK, BUT IF YOU'D
3 REM HAVE TO TYPE IT IN, YOU'D PROBABLY BE BETTER OFF
4 REM JUST USING THE FORMULA SHOWN IN THE ARTICLE.
5 REM delete REMarks 1-5.
10 GOTO 30
20 CLS : PRINT "save 'reachout/bas'" : SAVE "reachout/bas" :
RUN
30 DIM TC$(100), MI(100), CH(100)
40 CLS : PRINT "02/03/85  Dave McGlumphy  4429 Paula Ln
Chattanooga, Tn. 37415  (615) 875-4261."
50 PRINT
60 PRINT "WILL 'REACH OUT AMERICA' SAVE ME $$$ OR NOT?" :
PRINT
70 PRINT "This program is designed ONLY for the INTRASTATE
(as opposed to"
80 PRINT "interstate) 'REACH OUT AMERICA' telephone savings
plan."
90 PRINT "Don't enter local charges or long distance charges
that are" : PRINT "within your state." : PRINT
100 PRINT "Enter all dollar and cents amounts with a decimal
point."
110 PRINT : PRINT "Do not enter discounts, but do include
them in your total bill."
120 PRINT : PRINT "Press <ENTER> or <RETURN>, ok";
130 INPUT A$
140 CLS : PRINT "How much was your total long distance
bill";
150 TB = 0 : INPUT TB
160 IF TB < .01 THEN 140 : REM You must have paid something!
170 GOSUB 800 : REM Headings
180 IF CS > SB THEN 250 : REM Changing total bill
190 PRINT "Type of charge";
200 TC$ = "" : INPUT TC$
210 IF TC$ < "0" OR TC$ > "3" THEN GOSUB 800 : PRINT "Enter
a single digit from 0 to 3." : GOTO 190
220 IF TC$ = "0" THEN 370 : REM Exit input routine
230 PRINT "How many minutes for this call";
240 MI = 0 : INPUT MI
250 PRINT "How much was it";
260 CH = 0 : INPUT CH
270 IF CH = 0 THEN PRINT "Enter non-zero amounts." : GOTO
250
280 IF CS < 1 THEN 310
```

```
290 IF CS > SB THEN TB = CH
300 IF CS > 0 THEN TC$(CS) = TC$ : MI(CS) = MI : CH(CS) = CH
: GOTO 370 : REM To recalculate totals
310 SB = SB + 1 : IF SB > 100 THEN CLS : PRINT"The program
is limited to 100 entries unless you change the dimension
statements." : END
320 TC$(SB) = TC$ : MI(SB) = MI : CH(SB) = CH
330 GOTO 170 : REM for more data
340 REM
350 REM End of input routine
360 REM
370 CLS : REM Audit & allow corrections if audit fails.
380 GC = 0 : TC = 0 : TM = 0 : TE = 0
390 FOR J = 1 TO SB
400 IF TC$(J) = "1" THEN TC = TC + CH(J) : GOTO 430
410 IF TC$(J) = "2" THEN TM = TM + MI(J) : GOTO 430
420 IF TC$(J) = "3" THEN TE = TE + CH(J)
430 GC = GC + CH(J) : REM Grand Charge total
440 NEXT J
450 CLS : REM Print the results
460 PRINT "                          ", "BASIC", "BONUS"
470 PRINT "TOTAL LONG DISTANCE BILL", TB, TB
480 PRINT "TOTAL CHARGES               (D)", TC, TC
490 PRINT "FLAT RATE (INIT HR/MO)   ", "10.00", "11.50"
500 CC = (TM - 60) * 8.75 / 60 : REM Calculated Charges
510 CI = INT(CC * 100 + .5) : CC = CI / 100
520 IF CC < 0 THEN CC = 0 : REM CC applies only after the
first hour
530 PRINT "TOTAL MINUTES CHARGE       (N)", CC, CC
540 TS = TE * .85 : REM TS is for the BONUS plan
550 TI = INT(TS * 100 + .5) : TS = TI / 100
560 PRINT "CHARGES FOR EVENING CALLS (E)", TE, TS
570 A1 = TC + 10 + CC + TE : A2 = TC + 11.5 + CC + TS
580 PRINT "REACH OUT AMERICA COST    ", A1, A2
590 PRINT "AMOUNT SAVED (OR LOST IF -)  ", TB - A1, TB - A2
600 PRINT
610 IF ABS(TB - GC) > .05 THEN PRINT "The total you entered
doesn't equal the calculated total." : PRINT TB; "-"; GC;
"="; TB - GC; : PRINT "  Probably a discount." : PRINT
620 PRINT "Type 0 to end or 1 to see your data, ok";
630 INPUT A$ : IF A$ = "0" THEN END
640 IF A$ <> "1" THEN 620
650 REM End of results routine
660 REM
670 CLS : REM routine to display entries & allow correction
680 PRINT "You say that your total bill is";TB
690 PRINT "The total of your entries is"; GC
700 PRINT "To change one of these detail entries, type its
number which is"
710 PRINT "just before the colon. To add an entry, type 0."
720 PRINT "Otherwise, just press <ENTER> or <RETURN>."
730 FOR J = 1 TO SB : PRINT J; ":"; TC$(J); MI(J); CH(J), :
NEXT J
740 PRINT : PRINT SB + 1; ":"; TB; "(Reported total)"
750 CS = 0 : INPUT CS : IF CS > SB THEN CLS : GOTO 250 : REM
Total
760 GOTO 170 : REM To add or change detail data
770 REM
```

```
720 PRINT "Otherwise, just press <ENTER> or <RETURN>."
730 FOR J = 1 TO SB : PRINT J; ":"; TC$(J); MI(J); CH(J), : NEXT J
740 PRINT : PRINT SB + 1; ":"; TB; "(Reported total)"
750 CS = 0 : INPUT CS : IF CS > SB THEN CLS : GOTO 250 : REM Total
760 GOTO 170 : REM To add or change detail data
770 REM
780 REM End of display/allow-change routine.
790 REM
800 CLS : REM Headings
810 PRINT "                    TYPES OF CHARGES" : PRINT
820 PRINT "0 = Finished entering data."
830 PRINT "1 = (D)  all charges EXCLUDING evening, night and weekend"
840 PRINT "    direct-dialed INTERSTATE calls."
850 PRINT "2 = (N)  INTERSTATE direct-dialed calls made during the night"
860 PRINT "    and weekend period."
870 PRINT "3 = (E)  INTERSTATE direct-dialed calls made during the evening"
880 PRINT "    period." : PRINT
890 RETURN : REM From headings routine
900 REM
910 REM Variable list
920 REM A$  general input variable
930 REM A1  reach out cost
940 REM A2  reach out bonus plan cost
950 REM CC  Calculated Charges for night
960 REM CH  detail entry CHarges
970 REM CI  Integer of CC to round to 2 decimal places
980 REM CH  CHarge amount
990 REM CS  Change Subscript
1000 REM GC  Grand total of entered Charges (TC+TE+TM)
1010 REM J   For/next loop counter
1020 REM MI  MInutes
1030 REM SB  SuBscript
1040 REM TB  Total long distance Bill
1050 REM TC  Total daily Charges
1060 REM TC$ Type of Charge
1070 REM TE  Total Evening charges
1080 REM TI  integer of TS to round to 2 decimal places
1090 REM TM  Total night Minute charges
1100 REM TS  same as TE but for bonus plan
```

Password program for NEWDOS 80 V. 2
Key scan locations for the Model 100
By George Saladino

This month I am starting off with a short program that you can type in to put password protection on your NEWDOS V. 2 disk. This can be used on the Model 1 or Model 3/4. You can attrib passwords to all your files or use the system command

to set "AA=Y" or "AB=Y". See your manual for this.

However, you might want to try my program. Once you made the password by running my program, a person not knowing your password can not get around putting in the password if the disk is in drive 0 and is booted up.

I will explain the program below to show you what the lines do. I feel that this will give you an insight into some more advanced programing.

Lines 80-100 poke in my machine language program to create the password of three letters. (This program creates passwords of three letters only.) The program does not care if you type in lower or upper case when you enter your password. The program up to this point creates a password program that has the password of "ABC" in it.

Next lines 150-320 get your input of three letters and poke them into the program that was poked into memory by the lines mentioned above. Line 240 I create an upper case letter if you typed in a lower case letter or the other way around if you typed in a upper case letter. Line 250 shows you the letters you typed in. Please remember them, so you know what your password is. If you forget the password, you will have to put a different NEWDOS disk in drive 0 and boot it up. Then replace the disk with the forgotten password in drive 0. Run again my program and create a new password.

Lines 330-370 create the "CMD" password file on the disk. Line 330 writes what is in memory to disk. Line 340 makes your file "PASS/CMD" invisible. Line 350 makes it an "AUTO" program on boot up. Line 360 sets the "SYSTEM" system on NEWDOS80 to not accept the enter key to override the auto command. Line 370 boots the disk and allows you to try out the password you created.

Lines 380-500 is the data for the machine language password program. The rest of my program are just to read it into memory and create it.

Now call up a directory. You will not see the "PASS/CMD" program in the directory unless you type in "DIR I". The program is below.

PASSWORD/BAS

```
10 REM PASSWORD GENERATOR - USE IN NEWDOS V2 ONLY
20 REM MODEL 1 OR MODEL 3/4.
30 REM RUN PROGRAM IN DRIVE 0 ONLY!
40 REM GEORGE SALADINO
50 CLS
60 CLEAR 100
70 PRINT"W O R K I N G"
80 FOR X=&H929C TO &H9332
90 READ A:POKE X,A
100 NEXT
```

```
110 CLS
120 PRINT"Put in 3 Letters for your password as prompted."
130 PRINT "Do not use numbers or symbols!"
140 PRINT
150 FOR L=1 TO 3
160 PRINT"Put in # ";L;"Password letter";
170 PRINT
180 A$(L)=INKEY$
190 IF A$(L)="" THEN 180
200 A(L)=ASC(A$(L))
210 IF A(L)<65 THEN 160
220 IF A(L)>122 THEN 160
230 IF A(L)>90 AND A(L)<97 THEN 160
240 IF ASC(A$(L))>90 THEN B(L)=ASC(A$(L))-32 ELSE
B(L)=ASC(A$(L))+32
250 PRINT CHR$(A(L));" ";CHR$(B(L))
260 NEXT
270 POKE &H9300,A(1)
280 POKE &H9305,B(1)
290 POKE &H9313,A(2)
300 POKE &H9318,B(2)
310 POKE &H9323,A(3)
320 POKE &H9328,B(3)
330 CMD"DUMP PASS/CMD 929CH,9332H,929CH"
340 CMD"ATTRIB PASS/CMD INV"
350 CMD"AUTO PASS"
360 CMD"SYSTEM 0,BD=N"
370 CMD"BOOT"
380 DATA 243,205,201,001,243,033,194,146,017,000,060,001
390 DATA 031,000,237,176,195,251,146,000,000,000,032,032
400 DATA 032,032,032,032,032,032,032,032,032,032,032,032
410 DATA 032,032,069,110,116,101,114,032,121,111,117,114
420 DATA 032,112,097,115,115,119,111,114,100,032,116,111
430 DATA 032,099,111,110,116,105,110,117,101,032,032,032
440 DATA 032,032,032,032,032,032,032,032,032,032,032,032
450 DATA 032,032,032,032,032,032,032,032,032,032,032,205
460 DATA 043,000,000,254,065,202,015,147,254,097,202,015
470 DATA 147,195,251,146,000,000,000,205,043,000,254,066
480 DATA 202,031,147,254,098,202,031,147,195,015,147,205
490 DATA 043,000,254,067,202,047,147,254,099,202,047,147
500 DATA 195,031,147,195,045,064,000,000,000,000,000,000
```

Now a chart of the locations that the Model 100 looks at to see if the keys are pressed. If you peek the locations to the left of this chart, you can see the values returned. Using a statement such as "A=PEEK(65425)" you can see what values are returned for the keys Z, X and so on. You can use these locations to have the Control, Code, GRPH work for you in your program. For example if you put in "A=PEEK(65433)" and you put in an "IF" statement as follows, "IF A=32 then xxx" where "xxx" is another line number or some action such as "PRINT", you can find out when the "CAPS LOCK" is on! The chart is below:

Values	1	2	4	8	16	32	64	128
Mem Loc								

65425	Z	X	C	V	B	N	M	L
65426	A	S	D	F	G	H	J	K
65427	Q	W	E	R	T	Y	U	I
65428	O	P	BR	:;	"'	,<	.>	?/
65429	1!	2@	3#	4$	5%	6^	7&	8*
65430	9(	0)	-_	+=	LA	RA	UA	DA
65431	SPA	DEL	TAB	ESC	PAS	LAB	PRT	ENT
65432	F1	F2	F3	F4	F5	F6	F7	F8
65433	SHI	CTR	GRP	COD	NUM	CAP		BRK

BR = Brackets
PAS = Paste
SHI = Shift
GRP = Grph
CAP = Caps Lock
LA = Left Arrow
DA = Down Arrow
SPA = Space
ENT = Enter
CTR = Ctrl
COD = Code
BRK = Break
RA = Right Arrow
UA = Up Arrow

Lastly, a few words on the IBM PC and the Tandy 1000. There are the annoying keys of CAPS LOCK, NUM LOCK, and SCROLL LOCK that can on the PC be on with out you knowing it. On the Tandy 1000 there are lights on these keys. Well anyway, below is a short program that allows you to see what the values of these keys are when pressed. Just like the 100 above.

```
10 CLS
20 DEF SEG = &H40
30 I=PEEK(&H17)
40 IF I=0 THEN 30
50 PRINT I;
60 GOTO 30
```

There is a simple program. You will find the following values of the keys at segment &H40, location &H17 of memory.

Right shift = 1
Left shift = 2
CTRL = 4
ALT = 8
Scroll lock = 16
Num lock = 32
Caps lock = 64
INS = 128

Now that you know this you can always have a program where as above, DEF SEG to &H40 and then PEEK at &H17 and with a if statement, use these keys in your program. Also, if you want the NUM LOCK on in your program, DEF SEG to &H40 and then POKE &H17,32. NUM LOCK is on!

Try this also, DEF SEG to &H40 and then poke &H17,1. You will find that your number keys will always be in the shift position! No here is the handy thing about all this! To make sure that the keys are in there boot up positions in your programs, you can do this. 10 DEF SEG=&H40. 20 POKE &H17,0. You will find that the keys are reset to the boot up position!

THE MSDOS CHRONICLES

By Robert A. Fowkes

I recently had the dubious distinction of being named the MSDOS librarian for FCUG. The investiture was a simple one; 4 boxes of disks were deposited on my doorstep wrapped in the electronic equivalent of swaddling clothes -- the fliptop box. The simple directive "get these programs in order" accompanied the parcel and thus our MSDOS LIBRARY was born (or at least conceived). Although far from being an immaculate conception things are at least beginning to take shape and we are about ready to start making available a fairly large number of disks chock full of MSDOS public domain goodies.

The primary source of these programs was a now-defunct PC Bulletin Board. FCUG has obtained the rights to distribute these programs to our members who use the MSDOS operating system on their Tandy (or compatible) machines -- currently the Models 1000, 1200 and 2000. MSDOS has gone through several generations and what started as a single-sided 8 sector/track environment (MSDOS 1.0), and then changed to a double-sided 8 sectors/track format (MSDOS 1.1) is currently a double-sided 9 sector/track environment (MSDOS 2.x and 3.x....). The disks in our library collection are of the double sided, 8 Sectors per track variety. It is my guess that the programs were originally on single sided disks (MSDOS 1.0) and then were consolidated when MSDOS 1.1 became available. This is reinforced by the fact that most of the disks contain TWO directory documents (for example DOSUTIL1.DOC and DOSUTIL2.DOC). In addition the disks arrived with a numbering system already in place (each disk has an internal Label on the Directory, such as CLUB01.LBL and this is referred to in the directory descriptive files.)

In order to make the MSDOS library available as quickly as possible it has been decided to retain the previous numbering system. This brings some instant order to the large number of programs available and allows for future expansion. After reviewing the materials the following 26 disks are being placed in the FCUG MSDOS library (with more to follow):

DISKS #01 - #03: DOS UTILITIES
DISKS #10 - #12: BASIC UTILITIES
DISKS #20 - #22: BASIC APPLICATIONS
DISKS #30 - #32: COLOR PROGRAMS
DISK #41: PC-TALK III + Enhancements
DISK #42: PC-WRITE (Public Domain Word Processor)
DISKS #70 - #74: GAMES
DISKS #80 - #81: SPREADSHEET UTILITIES
DISK #82: MUSIC PROGRAMS
DISK #84: EDUCATIONAL PROGRAMS
DISK #86: DBASE PROGRAMS
DISKS #88 - #89: C PROGRAMS

A more complete table of contents for 25 of these disks (taken directly from files on the disks themselves) will be available under seperate cover [*See Editors Note*] so that

you can better determine whether the programs meet any of your MSDOS needs. (Note: The PC-WRITE disk contains its own 90+ page instruction manual which explains all modules on the disk.) Additional disks will be added to the library as more programs are catalogued and examined.

A WORD ABOUT COMPATIBILITY

Even though the majority of these programs will function in a generic MSDOS environment some of the programs have special requirements. Obviously, if a program looks for a color board and your machine does not contain this option there is a good chance that the program will not work. Also, some of the home-grown utilities may address the IBM BIOS (Basic Input/Output System) directly (a programming shortcut). Many times these BIOS calls result in incompatibility with Tandy and other MSDOS (non-IBM) machines. It is up to the individual member to test for compatibility. While many of the programs have been successfully run on a Tandy 1000, a Tandy 1200 and a Compaq there is no guarantee on this score.

DOCUMENTATION

A majority of the MSDOS files are accompanied by extensive documentation files that can be printed out for reference. Many of these documents can be printed directly from MSDOS and others are ASCII files that are best treated through a word-processor. (One or two documents are in Wordstar format, but some of the utilities available can convert these document files to other formats.)

A FINAL WORD (for now)

One of the most impressive things about the MSDOS community is the vast number of GOOD "Freeware" programs available. (Many of the documents discuss the entire "Freeware" concept in greater detail for those of you who are interested.) The origin of our MSDOS library was a source that was current in 1983. Several of the better known programs have undergone extensive upgrading over the years. It is virtually impossible to keep track of all the upgrades available. The best source of information concerning recent program enhancements,etc. would be one of the many IBM bulletin boards in the area or one of the IBM SIGS (Special Interest Groups) on Compuserve, etc. The primary focus of our MSDOS library will be to introduce you to the great variety of programs available in this format. Once exposed you will find it easy to keep abreast in your areas of interest.

Editors Note: Due to space limitations in the "Voice" the editorial staff has decided to publish the MS-DOS Library listing under seperate cover. This listing currently, runs about 30 standard pages. The staff will reduce this to a manageable size through reduction processing and have 25-30 copies available at the next meeting. Anyone unable to attend the meeting can aquire this listing by sending a self addressed 9 X 12 envelope with $ .37 postage, and your request to:

Herman Parks, 36 Creeping Hemlock, Norwalk CT 06851

BATCH FILES

by Bill Chilcote
Reprinted from Greater Cleveland PC User's Group

Batch files allow the use of replaceable parameters. This means that a special symbol, the percent sign (%), can be introduced into the batch file to act as a variable or "dummy" parameter. What "%" will actually represent is defined when the batch file is called up. For example, to copy a file or program from one drive to another, change its name and erase the original file, create a batch file called CHANGE.BAT with the commands and drive designations as follows:

```
COPY A:1%.OLD B:%2.NEW
ERASE A:%1.OLD
```

The numbers following the "%" stand for different variables to be used. In this case, the old filename is the first replaceable parameter and the new filename is the second. Up to nine separate variables can be used (%1-%9). %0 always represents the batch filename used. The variables will automatically be replaced by putting the desired filenames after the batch filename CHANGE separated with spaces. For example:

```
CHANGE OLD NEW
```

The command above will invoke the batch file with OLD and NEW substituted as follows:

```
COPY A:OLD.OLD B:NEW.NEW
ERASE A:OLD.OLD
```

For more versatility, the drive designation can be left out of the batch file and included as part of the filename in the following way:

```
COPY %1.OLD %2.NEW
ERASE %1.OLD

CHANGE A:OLD B:NEW
```

This allows you to choose any disk drive to work with. To actually use this example, don't forget to have a file called OLD.OLD on your disk.

One of the enhancements of DOS 2.0/2.1 over DOS 1.1 is the creation of logical capability. Some of the batch commands designed for this purpose are SHIFT, GOTO, IF, IF NOT and the FOR... IN... DO... statement.

The SHIFT command works with the replaceable parameters discussed above. When a SHIFT is performed, the parameter %1 takes on the value that used to be %2, and %2 becomes the %3 value, etc, etc. To demonstrate this an artificial batch

file called ALLCOPY.BAT will copy three files in the following manner:

```
ALLCOPY OLD NEW JOEBOB
COPY A:%1 B:
SHIFT
COPY A:%1 B:
SHIFT
COPY A:%1 B:
```

This is a little tedious and will only copy three files. use another batch command, GOTO, to set up a loop using both SHIFT and GOTO to copy forever. First tell GOTO where to go. This is done with a "label". A label can be anything - a word of eight letters or less - preceded by a colon to identify it. A label must be placed on a separate line by itself but can be placed in the batch file before or after the GOTO command. The label affords yet another chance to place a comment in a batch file. The comment is placed on the same line but after the label. The label and the comment are never displayed during the operation of the batch file. For example, modify ALLCOPY.BAT in the following way:

```
ALLCOPY OLD NEW JOEBOB
:CHERRY
COPY A:%1 B:
SHIFT
GOTO CHERRY
```

This unfortunately creates an infinite loop and the computer will start complaining that it can't find any files after the first three have been copied. To stop the loop when we run out of files, use the IF command to test for conditional logic. A true statement in the IF command will enact the command which follows it. The conditional statement to be tested followed by the command to be enacted are placed on the same line as the IF keyword. The IF command test for three conditions:

1. EXIST filename. This tests for the presence of a file on the default drive. If a file exists on the default drive, then the statement will be true.

2. ERRORLEVEL number. This tests for the presence of an error code number. If a program has reported trouble by signaling an error code to DOS, then this statement will be true (assuming you have picked the right error code number).

3. A string comparison which tests to see if two character strings are identical. This takes the form of IF string1==string2 (note the double equal signs). Both strings have to match exactly for the statement to be true.

If the condition in the IF statement is found to be true, then the command following it is executed. If it is not true, the batch file will skip to the next line. IF NOT works the opposite way. If the condition in the IF statement is false, the command following it is executed. Include these statements in the batch file as follows:

```
ALLCOPY OLD NEW JOEBOB
:CHERRY
COPY A:%1 B:
SHIFT
IF EXIST %1 GOTO CHERRY
```

The batch file above will copy along until it comes to the nonexistant file after JOEBOB. Since there is no file meeting this specification on the default disk, the condition is false and the batch file proceeds with the next line.

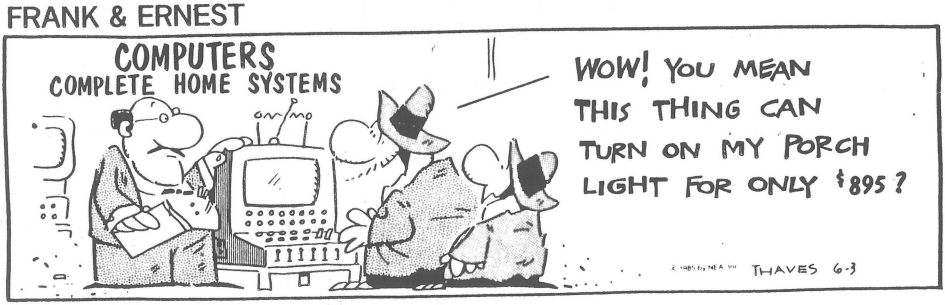
There is another batch command which can be used to perform repeated functions: the FOR... IN... DO... statement (the FOR statement for short). The FOR statement uses a variable with two percent signs (%%) followed by a character string. After the IN part of the statement, parentheses are used to enclose the group of filenames that are to be used by the command that follows the DO part of the statement. Perhaps an example will explain this better. To copy a group of files again try:

```
FOR %%FILES IN (OLD,NEW,JOEBOB) DO COPY A:%%FILES B:
```

FOR defines the variables, in this case "%%FILES", and DO defines the command to be performed. This one statement will copy each of the files in parentheses from the A: drive to the B: drive. A powerful statement! It should be noted that FOR statements cannot be nested. This means you can only put one of them on a line.

The real magic of batch files comes from fitting them to your specific needs and having the computer perform a series of commands from the simple push of a button.

FRANK & ERNEST



The President's Page

DATE: TUESDAY, JULY 2,1985
TIME: TIME: 7:30 PM
PLACE: NEW CANAAN COUNTRY SCHOOL

1. POWER GRAB AT FCUG!

The President has disappeared! Where he has gone I do not know. But, now, I AM IN CHARGE HERE (at least until Robert returns in the fall). So, I shall grasp the reins, don the mantle, mount the throne, surreptitiosly check my zipper, and hold forth at the July meeting with the fine program that our new program chairman, John Krause, has put together.

2. HOT PROGRAM FOR NEXT MEETING!

J.K. is taking no chances with his premier offerings. He has enlisted those two crowd-pleasers, Roger Giler and Alan Abrahamson, to speak.

Roger's set-up at home consists of a Model III and an IBM PC. These are arranged so that he can swivel 90 degrees to go from one keyboard to the other. Frequently, both machines are on and talking to each other. Roger is going to tell us how he feels about the capabilities of the two systems - and in the process introduce us to the MS-DOS operating system. This will be in keeping with our new policy of paying more attention to MS-DOS as the coming standard of compatible systems. Chances are, even if you are not using it now, your day will come, so listen carefully.

As any of you who have quizzed him about a blown disk (or a "farkled" disk as Lucien Greif calls it) has learned, Alan is a whiz at bringing back files, directories, and even whole disks to life. So harken to the wizard as he shares his secrets with us.

3. NEW "VOICE" FORMAT

Have you noticed that the copy of the "Voice" you are holding is somewhat changed in appearance from previous issues? The change has been made to offset a slow but steady shrinkage in our monetary balance. The new format reduces the paper used by 55%, the postage by 43%, the image size by 23%, and the actual editorial content by only 13%. So we feel that its a good deal.

See you Tuesday night --- Herman

FAIRFIELD COUNTY COMPUTER
USERS GROUP, INC.
c/o Alan Abrahamson
10 Richlee Road
Norwalk, CT 06851

FIRST CLASS

TO:

VOICE OF THE '80
Index for July

SUBJECT	Pg	SUBJECT	Pg
COMPUTER COMPETITION ..	2	Question # 1 <easy> ...	2
Question # 1 <medium> .	3	Question # 1 <hard> ...	4
The MAILBAG	5	Will REACHOUT Save Me $	5
REACHOUT/BAS	8	Password Program ND80V2	10
Key Scan Locations M100	10	PASSWORD/BAS	11
Model 100 key Locations	13	THE MSDOS CHRONICLES ..	14
BATCH FILES (MSDOS) ...	16	The President's Page ..	19

The Voice of the '80 is a publication of the Fairfield County Computer Users Group, Inc. Permission to reprint is granted for non-commercial and non-profit users. Credit is appreciated. Submissions should be sent care of the editor at the address above:

PRESIDENT: Robert Jackson VICE PRES: Herman Parks
SECRETARY: Patricia Brinson TREASURER: Aaron Bisberg
EDITOR: Alan Abrahamson TBBS SYSTEM: 203/846-6728
Ask Alan HOTLINE: Evenings (6PM-10PM) 203/866-7883
This months cover art by: Alan Abrahamson

This newsletter printed by Technical Reproductions Inc.
326-A Main Avenue, Norwalk CT 853-9100