TANDY ELECTRONICS

May 1981

TRS-80™ Microcomputer NEWS

P.O. Box 229, Rydalmere, NSW. 2116.

THE MICROCOMPUTER NEWSLETTER FOR TRS-80 OWNERS VOLUME 2 Issue 1

INTRODUCING:

Up-to-the-minute news from where it all began

Fort Worth Scene

For all you Model II owners who have, by now, been totally confused by 1.2a and 2.0a, many thanks for your patience. I would like to tell all the TRS-80 owners about what happened as it illustrates just how crazy the computer biz is. I'm sure most of you know that the microprocessor has a lot of logic in it to interpret the commands it receives. You may not know that a lot of other chips also have imbedded logic or microcode and are also processors of a type. For example, in the Model II, the keyboard, video, serial ports and the disk controller are all run by their own intelligent processor IC's which are big chips called LSI devices. These are very complex devices, some much more complex and expensive than the Z80A microprocessor.

As with all complex IC's, there is always the chance that some of the code is incorrect and that the IC will not, under some circumstances, do what the manual says it will do. If you start to get the idea that microcomputers are designed by the semiconductor manufacturers, you are not far wrong. Anyway it seems that the disk controller IC in the Model II had a very obscure fault. By very obscure I mean that none of our software ever caused the fault to be found in over a year of Model II production, nor did any other users of the same IC find the fault. Enter Model II Accounts Receivable (269-4504), our first COBOL program. During many hours of testing it always worked here but, once it got to you, all of a sudden the Model II started producing an error message "sector not found". What that means is that the computer can't find the data it wrote out to the disk. Result here, PANIC! To compound the problem our engineers and software people, who worked a lot of overtime on this one, found that the data was on the disk but the controller for some reason could not find it.

After several days we had to conclude that TRSDOS was not at fault, nor was our COBOL, it was the Model II itself and most likely the disk controller IC. At this point our supplier of the controller IC sent a crew down to Fort Worth with orders not to return until the problem was solved. Well they found the problem and announced that, sure, they could remask the chip and make new ones — only it would take about 6 months. They then sat down with our system software people and worked out a software fix for the hardware problem and we released TRSDOS versions 1.2a and 2.0a. The final decision on what to do and how to do it occurred December 24, late in the afternoon and long after we all had promised to be home. Well no one ever said Murphy's law respected Christmas.

For those of you who do not own Model IIs, you might be interested to know that we mailed every owner a new Accounts Receivable disk and made the "a" versions available, free, to every owner.

So if any of you want to go into the microcomputer busines, do it with your eyes open, there are a lot of mine fields out there.

Until next month.

This month's issue of the TRS-80 Microcomputer NEWS signals a change to a new format as well as a new name. The Microcomputer NEWS will be a monthly publication containing programs, hints and fixes to Tandy programs.

This newsletter is an open communication channel between Tandy and you, our customer. Our objective is to keep you informed, whether it is about new products, corrections to our programs or programming suggestions.

We also welcome the receipt of computer programs, hints and any suggestions you may have about the content of this newsletter. In short anything that you think may be of interest to other TRS-80 owners. In order for us to reprint your submission, you must specifically request that your program be considered for inclusion in the newsletter and provide notice that you hold no copyright or any other exclusive rights to the material. This assures that our readers can recopy the material without any legal hassles.

Computer Support

Tandy is firmly committed to its moral responsibility as a microcomputer manufacturer to back up its sales with service, software, consulting personnel, providing instructional facilities and maintaining a parts inventory. To this end, a special department has been established called Computer Support with the above stated objectives.

Progression towards achieving these aims is as follows:

We currently have five service centres in operation (Sydney, Melbourne, Adelaide, Perth, and Brisbane) and have introduced a field service contract for Model II owners to provide on-site repairs and maintenance.

Our range of software is extensive and is continually being added to (both local and U.S. programs) with some exciting new programs in the pipeline.

Consulting personnel are available at all Tandy Computer Centres to answer your questions or refer you to someone who can.

Instruction manuals have been printed for all our equipment as well as books and manuals on BASIC programming. On top of this, introductory and programming courses on our computers are being held on a regular basis, at Computer Centres.

We have a comprehensive parts inventory in our warehouse at Rydalmere and maintain parts inventories at all service centres.

As you can see, since the introduction of the TRS-80 into Australia in March 1978 we have been working towards achieving our objectives and in fact have progressed a long way. What other microcomputer company in Australia can offer all these services?

Machine Language Sort for Model I and Model II

The following machine language program will sort an array of alphanumeric strings. It is an in memory sort, that is the sorted data will remain the same array (A$).

To give you an idea of the speed of the sort routine we compared it to the BASIC sort routines as described in the May/June 1980 Newsletter. Using the same type of random strings, this is how the routines compared (times in seconds)

Number Strings	Best BASIC	Machine Language	
		Model I	Model II
10	2	1	1
50	7	1	1
100	50	1	1
500	978	4	2
1000	2560	9	4

For a 16K Level II:

```
1000   REM LIST FOR 16K (ENTRY = &H7F00)
1010   REM MEMORY SIZE = 32512
1020 DATA 205,127,10,94,35,86,237
1030 DATA 83,19,127,35,94,35,86,237,83
1040 DATA 213,127,33,0,0,34,211,127
1050 DATA 237,91,211,127,203,59,127
1060 DATA 203,58,48,2,203,251,237,83
1070 DATA 211,127,122,179,200,42,19
1080 DATA 127,237,82,34,207,127,33
1090 DATA 0,0,34,205,127,42,205,127,34
1100 DATA 203,127,42,203,127,237,91,211
1110 DATA 127,25,34,209,127,235,33
1120 DATA 0,0,25,25,25,229,237,91,203
1130 DATA 127,33,0,0,25,25,25,237
1140 DATA 75,213,127,9,235,225,9,229
1150 DATA 213,14,0,126,71,26,184,48
1160 DATA 3,14,1,71,175,176,40,25,197
1170 DATA 19,35,78,35,70,197,225
1180 DATA 235,78,35,70,197,225,193,26
1190 DATA 150,56,10,32,39,19,35,16
1200 DATA 246,203,65,32,31,209,225,6
1210 DATA 3,78,235,126,113,235,119
1220 DATA 35,19,16,246,42,211,127,235
1230 DATA 42,203,127,175,237,82,34
1240 DATA 203,127,48,144,24,2,209,225
1250 DATA 42,205,127,17,1,0,175,25
1260 DATA 34,205,127,237,91,207,127,237
1270 DATA 82,218,58,127,195,24,127
1280 N=0
1290 FOR I=1 TO 203
1300       READ A
1310       N=N+A
1320        POKE I+32511,A
1330 NEXT
1340 IF N◄►22393 THEN PRINT" ERROR IN DATA"
```

The following program uses this routine, and demonstrates the speed of the sort:

```
10   CLS
20   CLEAR 10000
30   POKE 16526,0:POKE 16527,127 ' FOR 7F00
40   DEFINT A-Z
50   DIM A$(750),X(2): Z=0
60   GOSUB 1000
69   REM GENERATE A LIST OF STRINGS TO SORT
70   INPUT "NUMBER OF ITEMS ◄ 750";N
```

(continued on page 7)

Model II Bugs, Errors and Fixes

General Ledger (269-4501)

The batch Total in a posting summary does not equal the document balance.

To correct this, Change line 730 of "Txpost" to read:

```
730   IN#=0: FORQ=0TO NE-$:
      IF AM#(Q) ►0THEN
      IN#=IN#+AM#(Q)
```

You also need to change line 6730 of the "Txentry" program. Change this line to read:

```
6730   IN#=0: FORQ=0TO NE -1:
       IF AM#(Q) ►0 THEN IN#=IN#+
   AM#(Q)
```

Accounts Payable (269-4505)

There have been several customers who have encountered error code 5 in the invoice section of version 1.0 Accounts Payable. The same error is also encountered in End-of-period processing. To correct these problems, make the following changes:

For invoices, change line 350 of the "APINVCE/BAS" program to read:

```
350   N#=VAL(IN$): W#= ABS(N#)* 100:
    W#= INT(W#* 100D0+ .5D0)/
      100D0
      V$="":...
```

The rest of the line is unchanged. Be sure you save a copy of the corrected program.

For End-of-Period, change line 800 in the "APPROC/BAS" program to read:

```
800   W#= ABS(N#)* 100:W= INT(W#*
      100D0+ .5D0)/ 100D0:
      V$="":...
```

The rest of this line is unchanged. Again, do not forget to save a copy.

Patches for TRSDOS 2.0

In the TERMINAL utility, the "G" (Get disk file into RAM) function, when used with a variable-length record file, will insert 1 extra byte before every logical record.

To prevent this from happening, apply the following two patches:

```
PATCH TERMINAL A=3723
      F=FE4628 C=D65620
PATCH TERMINAL A=372A
      F=06004E23AFB9C004C9
      C=474E230DC0E1C3F632
```

False Error Code 7 — Illegal Disk Change

Early reports from Model II owners indicate that some disk drives are more prone to soft read errors than others. Some media may also cause these soft read errors.

If you have a problem of getting too many false Error 7's due to these soft read errors, apply the following patches to TRSDOS:

```
PATCH SYSRES/SYS A=0E9C
      F=7EB7 C=CB7E
PATCH SYSRES/SYS A=0D45
      F=91 C=99
PATCH SYSRES/SYS A=0D94
      F=91 C=99
```

Applying these patches will not affect system performance, but will eliminate the incorrect Error 7's.

If you get a false Error 7 after these patches have been applied, your diskette contains a hard read error and must be re-formatted.

269-4505 Model II Accounts Payable

Some customers have asked about an option which will allow posting of invoices without them having to return to the Main Menu. The purpose of this option is to save time by not having to load those large programs. If these changes are made, you should be aware that a COMPLETE posting of ALL invoices must be done since you will be unable to select invoices as you can in the normal posting method. Make the following changes to the "APINVCE/BAS" program.

Change lines 502, 850, 860 and 1024 as follows :

```
502   PRINT@(23,20),EL$"► ENTER A
      SELECTION : ";:FL=1:GOSUB
      10: IFCF=1 ORCF=2 THEN 850
      ELSE IFCF◄►0 THEN 502
850   CLOSE:OPEN"O",1, VP$....
860   ....:NEXT:CLOSE1
1024   ....;TAB(40);FNRV$("N");
       "EXT";TAB(55);FNRV$("F2")
       " POST INVOICES"
```

Note that in line 850 we removed the first part of the line, and that in 860 and 1024 we changed only the end of the line.

To complete these changes we need to add a new line:

```
870  IFCF=2THEN RUN"APPOST/
     BAS" ELSE RUN"APS/BAS"
```

Save these changes by using:

SAVE"APINVCE/BAS"

With these changes, pressing the F2 key at the "INVOICE MAINTENANCE" menu will take you directly to "INVOICE POSTING". Remember that if you do this, you will have to use the Complete option in posting.

In Model II Accounts Payable, a problem is occurring with indexing of vendors. A zero is placed in the array that holds the record numbers for the vendors. This causes an error code 61 to occur. To stop this from happening, change APS/BAS as follows:

Change lines 1722 and 1726 to read:

```
1722  IFW1►P1 THEN S1= -1 ELSE
      P1=P1-1
1726  IF W2►P2 THEN S2= -1 ELSE
      P2=P2-1
```

If you have had this error, you must re-index the vendor file. The following procedure will reconstruct the index. NOTE do the following to a backup of your Accounts Payable working diskette. Do NOT perform the procedure on your current work diskettes until you are sure the recovery procedure has corrected the problem.

PROCEDURE FOR RE-INDEXING APS (269-4505)

1) From TRSDOS READY, type : BASIC -F:3 ENTER

2) Type in the following program:

```
20000   ZN=VN: ZD=VD
20010 VN=1: VT=0: VD=0
20020   FOR W9=0 TO ZN-1
20030   P(W9,0)=0: P(W9,1)=0
20040   NEXT W9
20050   FOR ZZ=1 TO ZN-1
20060   GET2,ZZ
20070   VV=CVI(V0$)
20080   V$(1)=VF$(1)
20090   GOSUB 1700
20100   VT=VT+1: VN=VN+1
20110   NEXT ZZ
20120   IF ZD=0 THEN 1000
20130   CLOSE 1
20140   OPEN"I",1,VP$
20150   FOR W9=1 TO 23
20160   INPUT#1, XX!
20170   NEXT W9
20180   FOR W9=1 TO VN-1
20190   INPUT#1,X1,X2
20200   IF X1►0 THEN 20280
20210   VL=ABS(X1)
20220   FOR W8=0 TO 1
20230   FOR W7=1 TO VN-1
20240   IF P(W7,W8)=VL THEN P(W7,
        W8)= -VL:W7=VN-1
20250   NEXT W7, W8
20260   VD=VD+1: VT=VT-1
20270   IF VD=ZD THEN W9= VN-1
20280   NEXT W9
20290   GOTO 1000
```

3) Save this program using: SAVE "RECOVER/BAS",A ENTER

4) Type: LOAD"APS/BAS"ENTER

5) Type: MERGE"RECOVER/BAS" ENTER

6) Type: RUN ENTER

7)) Enter the password. At the Main Menu,press BREAK

8) Type: GOTO 20000 ENTER

The recovery process will be completed and the program will return to the Main Menu. Press F1 to exit the program.

269-4502 Inventory Management

Inventory Management, when listing unposted sales transactions in version 1.0, will list 16 items and drop the 17th. To correct this, change line 2540 (Version 1.0 ONLY!) in the "SALES/BAS" program to read:

```
2540   P=P+1: NEXT: IF N+15◄NS
           THEN NN=N+16 ELSE
       N=1
```

It is possible that the number of items carried by Inventory Management will be wrong. This can happen for any of the following reasons:

1) Abnormal exit from the program

2) Index file damaged

3) When RECOVER is run on the 1.1 version, the system comes up with the wrong number of inventory items if there were duplicate stock numbers. The following procedure will correct this problem for versions 1.0 and 1.1:

At TRSDOS READY, type IMS ENTER

At the Main Menu:

Version 1.0 Press BREAK

Version 1.1 Press CTRL P then

BREAK

When the program BREAK's (both versions), type:

NI=n ENTER (where n is the number of items.)

CONT ENTER

Now press the R key. This causes the program to go into the Review/Edit mode. Press the F1 key to return to the Main Menu. Check the number of inventory items assigned.

269-4920 BASIC 1.2 Manual Error

Page 3/12A of the TRSDOS 2.0 BASIC 1.2 reference manual is the page which describes the NAME command.

Change ALL references to:

RENAME OLDNAME TO NEWNAME

to:

NAME OLDNAME AS NEWNAME

Please note that there are two distinct errors here. The first is the use of the connective "TO" instead of "AS". The second was the use of "RENAME" instead of "NAME."

Model I Bugs & Fixes

In Memory Information (269-1508)

In version 3.0, when entering data (cards) and approaching the end of available memory, all data will be lost if another card is entered and the "BYTES FREE" is less than TWICE the card length.

The program will accept the information, but then display the number of bytes free for about 3 seconds, indicate "OUT OF MEMORY" and execute a total restart, as if beginning the program from scratch.

On page 7 of the documentation is a note that states the following:

NOTE: If the BYTES FREE value drops below the CARD LENGTH value you can't add any more records.

Please change this line to read:

Note: If the BYTES FREE value drops below two times the CARD LENGTH value, you can't add any more records; to do so will result in loss of all data (execution of (R)ESTART).

Accounts Payable (269-1554)

During End-of-Period processing (in VERSION 3.0 only), an error code 5 may occur in line 45. To prevent this, change lines 45 and 151 of the "PROCESS" program to read:

```
45   W#=ABS(N#)*100:W#= INT(W#*
     100D0+ .5D0)/ 100D0: V$= "":
     X=W#/ D1#: W#=W#- X* D1#: V$= V$+
     CHR$(X- (N#◄0)* 128): X=W#/ D2#:
     W#=W#- X* D2#: V$=V$+ CHR$(X):
     X=   W#/ D3#: W#=W#- X* D3#: V$=V$+
     CHR$(X)+ CHR$(W#): RETURN
151   PRINT: PRINT CHR$(30);
      "UNEXPECTED ERROR CODE"; ERR/ 2+
      1; "IN LINE"; ERL: GOSUB 65: END
```

Accounts Receivable (269-1555)

If you are using the 3.0 VERSION of Accounts Receivable under the three drive option, you will encounter a BAD FILE MODE error in line 628. To prevent this error from occurring, the following changes need to be made to the "SETUP" program:

Change lines 220 and 880 to read:

```
220  FL=1: GOSUB 280: R$= IN$:
     IF CF◄►0 THEN 220 ELSE IF
     R$◄►"Y" AND R$◄► "N" THEN PRINT
     CHR$(8);: GOTO 220
880  PD=2: PC=500:
     PT=2500: IF Q$="M" THEN ON
     ERROR GOTO 895: KILL PT$: PT$=
     LEFT$(PT$,LEN(PT$)-1)+"2": CLS:
     PRINT@458, "INSERT DATA DISK IN DRIVE 2 and
     PRESS ◄ENTER►": ELSE
     GOTO 890
```

Notice that we removed the end of line 220, and added a statement to the end of line 880. You also need to add the following line:

```
225  IF R$="N" THEN GOSUB 560: IF
     Q$="I" THEN ZX=0: GOTO 80
```

Also in Version 3.0, there is an extra quote mark (") at the end of line 1590 in the "ARS" program. This extra quote can cause an error code 2 to occur duirng posting. If you will eliminate the last quote mark in line 1590, you will eliminate this error.

Business Mailing List 269-1558

The following procedure will allow you to recover and re-index Mailing List data, in versions prior to 3.0 only, which has been lost due to a system failure or an abnormal program exit. Note: This procedure will also "recover" item which were deleted if the space was not re-used by the program.

1) Enter the following program module:

```
4999   'RECOVERY
5000   CLS: N=0: TN=1: F=0
5005 PRINTTAB(20) "** FILE RECOVERY  **"
5010   FOR Q=0 TO CP: V(Q,0)=0:
       V(Q,1)=0: NEXT
5015   PRINT@320,"** RECORD NUMBER : "N
5020   N=N+1: J=N: GOSUB 1840:
       GOSUB 1900: GOSUB 1910
5030   IF E1$◄ CHR$(32) OR
       E0$► CHR$(127) THEN 5100
5040     NS=1: GOSUB 3210: TN=TN+1:
       EL$=E1$
5060   IF N◄CP THEN 5015
5100   GOSUB 1790
5110   PRINT: PRINT "► RECOVERY
       COMPLETE — PRESS ◄ENTER►";
5120   GOSUB 1460: GOTO 210
```

2) Save the module using the following BASIC command:

SAVE "RECOVER/ASC",A ENTER

3) Type: LOAD"MLS" ENTER

4) Type: MERGE"RECOVER/ASC" ENTER

5) Type: RUN ENTER

6) When the Mailing List menu appears, press the BREAK key.

7) When the screen shows "READY", type:GOTO 5000 ENTER

8) The system will examine your data files and retrieve, re-sort and index all the valid data found. When this process is complete, the screen will show: "RECOVERY COMPLETE — PRESS ENTER". Press ENTER

9) The Mailing List Menu will appear. Press the @ key to exit the program immediately.

10) Make BACKUP copies of your "recovered" disks. Examine the data carefully before copying onto your original disks. If the data is not correct, try the operation again.

Tiny Pascal (269-2009)

You should be aware that in mathematical calculations the largest value you can work with is 32767. There are two corrections that need to be made on page 9 of the Tiny Pascal manual:

OUTP(a,x) Outputs the value x to port a. This is the opposite of what the manual says.

SQR(exp) returns the square of exp, not the square root.

FORTRAN — Model I (269-2201) and Model II (269-4701)

The FORTRAN User's Manual (Page 19 for Model I, Page 9 for Model II) indicates that the default logical record length (LRL) for a CALL OPEN statement is 128 bytes. This is incorrect.

If a programmer does not specify a LRL in the CALL to the OPEN subroutine, no error message is generated by the compiler, linker, or run-time package, and the LRL which is produced is unreliable and may vary from run to run. The format of the records written to files opened with no specified LRL is unreliable.

Solution: ALWAYS specify a LRL when using the CALL OPEN statement.

269-1558 Business Mailing List

There have been several reports that listings and labels printed in the Postcode order would be out of sequence after updating any Postcode. The changes listed below should keep this problem from occurring. To correct a diskette on which this has already occurred, you will need to make the following program changes, then delete and re-enter any items which were printed out of order. Make these changes to the "MLS" program:

```
830   PA=N1: GOSUB 2980: PRINT@64,
      "SELECT LINE TO BE UPDATED--►";
1090  GOSUB 2050: P7=640: IFCF◄►0
      THEN 640 ELSE 1130
1130  GOSUB 1870: IFPA◄►N1 THEN 530
      ELSE GOSUB 2190: GOTO 640
```

269-1565 Microfiles

Microfiles contains the following problem:

If a file contains more than 256 records and you delete any record, you can not add records to that file again. The one exception is if the record is the last one in the file.

On a two disk system follow this procedure"

1) Place a system (TRSDOS) diskette which contains BASIC in drive 0. NOTE: The Microfiles diskette does not have BASIC.

2) Load BASIC and enter this program:

```
10  INPUT" READY NEXT DISK";A$
20  OPEN"R",1,"FILES/VIR"
30  FIELD#1, 48 AS X1$, 1 AS F1$,
    148 ASX2$, 1 AS F2$
40  GET 1,15: LSET F1$=CHR$(4)
    :PUT 1,15
50  GET 1,3:LSET F2$="6": PUT 1,3
60  CLOSE: GOTO 10
```

3) After you have verified the program, enter SAVE"MICRO/FIX" to save a copy of this fix program.

4) RUN the program with your Microfiles diskette in drive 1.

5) You can make this modification to all copies of Microfiles by swapping to the next diskette at the point where "READY NEXT DISK" is displayed. To stop the program press READY when the "READY NEXT DISK" message is displayed.

For a one disk system, follow steps 1-3 above. Swap diskettes so that the Microfiles program diskette is in drive 0. Type RUN. Continue with step 5 above until all copies of Microfiles have been corrected.

In addition to the above problem, there is also a problem with the Microfiles index for versions 2.0 and earlier. An error exists in the logic of the FIND command. Even though the "INDEX BY" command puts records in the proper sequence, they are sometimes not found by "FIND". The error is most likely to occur when there are many records with the same value in the index field and a few of these records contain one or more trailing spaces. Due to an error in logic, this will sometimes cause the binary search to "take a wrong turn". When this happens, a "DISPLAY NEXT" usually shows the record(s) being sought.

The solution is to eliminate the trailing spaces. This will not only make "FIND" work properly, but will also save space in your data file. To check a field for trailing spaces, do the following (in Microfiles):

BUILD FORMAT TEST ENTER O (do NOT press ENTER)
DOWN-ARROW, Y, SHIFT-RIGHT-ARROW
(5 Times) then type the name of the
field to be checked, ENTER
SHIFT-LEFT-ARROW twice, L, *, ENTER , Y, ENTER
Then type : D ALL ENTER

This will cause the field to be displayed for all records, each one on a new line. If it is correct, there will be one space before the asterisk. Use the space bar to slow up the listing if necessary. When you see extra spaces, push the ENTER key to stop the listing. Then type: D PREV ENTER if necessary to display the one with the error. When the last record displayed on the screen is the one with the error, type C the name of the field, ENTER

Now use the right-arrow to move the cursor over the first space that follows your data. Then push SHIFT-CLEAR. This will remove the trailing spaces. The cursor will jump back to the beginning of your data if you did it correctly. Finally press ENTER to store the record.

Type: D REST ENTER to review the rest of your records.

NOTE: Even if you had not had this problem, it is worth knowing about and using SHIFT-CLEAR. Typed spaces, even at the end of a field are considered to be data and are stored in the record.

Teacher Aid (269-1713)

A problem occurs if more than nine grades are assigned to a category.

Two lines must be changed in the program.

Change line 240 to read:

```
240  FOR I= 1TO5:G(I)=VAL(MID$(BF$,
     63+(I-1)*7,4)): T(I)=VAL(MID$
     (BF$,66+(I-1)*7,4)): IF T(I)=0
     THEN G(I)=0 ELSE G(I)=G(I)/T(I)
```

Change line 630 to read:

```
630  BF$=RIGHT$(SF$(I),35): G(I)=
     VAL(MID$(BF$,1+(GC-1)*7,4)):
     G(2)=VAL(MID$(BF$,4+(GC-1)*7,4))
     : G(1)=G(1)+SD(I): G(2)=G(2)+1
```

Cassette Portfolio (269-1506)

Cassette Portfolio is designed to be a personal record keeping system that will help you keep track of your portfolio, and how you stand in relation to the money you have invested.

One area of this program which has been questioned by a few customers, is the area of computing returns. Cassette Portfolio computes returns on your ACTUAL PURCHASE PRICE (Effective Yield), not current market price (Current Yield).

If you want or need to compute returns based on current yield, the following change should be made to the program TSUMM:

In line 129, change B#(J) to H#.

269-1604 Model I Versafile

In Model I Versafile there is a subscript out of range error. To keep this error from occurring, make the following changes in the "VERSA" program.

```
110   CLEAR 13000
120   DEFSTR A,B,C,D,E,F,K:
      DEFINT L,X,Y,Z: DIM E(200),
      Y(200), A(25), K(25)
```

269-1705 Model I Advanced Stat. Analysis

In some instances, frequency distributions have not correctly placed items in the correct categories. Change line 10 in the "ASAFD" program to read:

```
10   CLS: DEFINT I-N: DEFSTRZ: N=0:
     J=0: MT=1: A$="######.###":
     DIM A(11)
```

Change line 10 in the histogram program to read:

```
10   CLS: DEFINT I-N: DEFSTRZ: N=0:
     J=0: MT=1: A$=###.#"
```

Make the following changes to the "HISTOGRAM" program to make the range between different frequencies equal.

```
1480   FI=INT (FI+KK/6*10)/10: I=I+1
       L2(I)=INT(FI)+LX: H=L2(I):
       FI=L2(I): NEXT J
1485   L2(1)=0: F2(1)=0: I=1: T=L2(7)/N
       : H=0: FOR J=888TO120 STEP -192
       : PRINT@J,"";: PRINTUSINGA$;H;
1530   FORLY=40TO40-LA(I)/L2(7)*37
       STEP-1: SET(J,LY): NEXT LY,J
```

Some explanation concerning interpretation of histogram is in order.

1) Frequencies will always be whole numbers. That is, you cannot enter half of one data item into the program.

2) When reading the histogram, keep in mind that the purpose is to get an overall picture of the frequencies of the items in the different intervals, and to compare them.

3) Since the precision of the histogram is limited by size and utilization of the video screen, it is a good idea to use the results from the "Frequency Distribution" program together with the histogram for determining the exact relationships.

In analysis of variance with two groups, the counter for group two does not count correctly. To correct this problem, make the following changes to the "Tape Data Files" program: Delete line 1090.

Change the following lines:

```
1060   GOSUB6000: JJ=0: FOR J=1 TO 8:
       JJ=JJ+1: IF Z(J)="@" N(K)=M+JJ-1
       :JL=J:GOSUB 7000:JJ=0:K=K+1
       : IF MT=3 M=M-JL

10090   E=0: KL=1: FOR K+1 TO NT: E=E+1
        : IF ZD (K)="@" THEN KL=KL+1
        : E=E-1: GOTO 10100: ELSE IF
        ZI="Y" LPRINT"ELEMENT #";E,
        "GROUP #";KL,ZD  (K)
10095   PRINT"ELEMENT #";E,"GROUP #";
        KL,ZD(K):GOSUB9900 REM DELETE
        REST OF LINE
```

Accounts Payable (269-1554)

Some customers have destroyed their index files due to improper exit from the program. The following sequence of instructions will reconstruct the index, unless the index has been "zeroed out".

At TRSDOS READY type:

```
BASIC ENTER
```

Answer the questions and type:

```
LOAD"APS" ENTER
```

Add line 257:

```
257 NA=1: GOTO620
```

Type RUN

Enter a new vendor and sort, this should reconstruct your index file.

The Accounts Payable program will allow you to enter an invalid first check number. Your check numbers should range from 0001 to 9999. An error will occur if you enter a check number in the range 32767 to 999999.

To prevent this problem from occurring, make the following changes to the "CHECKS" program. Change line 555 to read:

```
555   GOSUB 590: FL=-4: ....
```

Accounts Receivable (269-1555)

After end of period processing you can increase the number of accounts if you wish. You cannot decrease the account capacity.

If you wish to do this, make the following changes to the "SETUP" program.

Change line 880 to read:

```
880   PD=2: PC=500: PT=2500: IF Q$="M"
      THEN ON ERROR GOTO 895: KILL PT$
      PT$=LEFT$(PT$,LEN(PT$)-1)+"2"
      OPEN"R",3,PT$: CLOSE 3: ON ERROR
      GOTO 600
```

Add line 895:

```
895 IF ERR/2+1=54 THEN RESUME NEXT ELSE ON
ERROR GO TO 0
```

You can now change the number of accounts as you desire.

NOTE:
All references to "less than" and "greater than" symbols (< and > on the TRS-80) have been replaced with the following symbols:

◄ is the same as <

► is the same as >

```
80   FOR I=0 TO N-1
90   A$((I)= STRING$(RND(30),
     RND(26)+64)
140  PRINT I; A$(I),
150  NEXT I
160  INPUT"PRESS ENTER TO START SORT"
     ;ZZ
170  X (0)=N
180  X(1)=VARPTR  (A$(0))
190  Z=USR(VARPTR(X(0)))
200  FOR I=0 TO N-1
210  PRINT I;A$(I),
220  NEXT I
230  INPUT"PRESS ENTER TO REPEAT";ZZ
240  CLS: GOTO 70
1310  RETURN
```

This listing shows the proper format to use with 16K Level II BASIC.

The program can be run in 32K or 48K by 'relocating' it. To relocate the program, change ALL occurrences of 127 (except the first one) in the DATA statements as indicated. There are 23 occurrences which must be changed.

For 32K, change to 191.

For 48K, change to 255.

Memory sizes are:	Level II USR entry points are :
16K — 32512.	16K — POKE 16526,0: POKE 16527,127
32K — 48895.	32K — POKE 16526,0: POKE 16527,191
48K — 65279.	48K — POKE 16526,0: POKE 16527,255

Disk BASIC DEFUSR entry points are:

16K — &H7F00

32K — &HBF00

48K — &HFF00

For 32K, make the following additional changes:

```
1240  POKE I-16641,A
1260 IF N◄► 23865 THEN END
```

If you are using a 48K system make these additional changes:

```
1240  POKE I-257,A
1260  IF N ◄► 25337 THEN END
```

This program should allow you to 'play' with this sort routine. Notice that it only sorts literal (alphanumeric) data items which have been stored in a single dimension array (A$). When the sort is finished, the sorted items are still in the array A$. The size of the array which can be sorted is limited only by the amount of memory you have available. For 32K and above you can easily increase array size to 1500.

In addition to the single dimension array A$(n), array X(2) should be dimensioned, and Z set to zero before the USR routine is called. One further caution is to avoid placing any additional statements between lines 170 and 190 in the demonstration program. If you assign valued after you have assigned X(0) and X(1) and before you use the USR, the sort will not work because the pointers have changed.

Model II Version

For Model II, our setup is somewhat different. Since we do not have a POKE command, we will use the DEBUG facility under TRSDOS to enter the machine language sort. Once you have entered the routine and saved it to disk, everything, including subsequent uses of the machine language routine will be handled from your BASIC program.

The following information is given for a 64K Model II.

Using the DEBUG utility as described in the Model II TRSDOS Manual, we need to modify memory starting at location F000.

```
00  00  00  5E  23  56  ED  53
13  F0  23  5E  23  56  ED  53
D5  F0  21  00  00  22  D3  F0
ED  5B  D3  F0  CB  3B  AF  CB
3A  30  02  CB  FB  ED  53  D3
F0  7A  B3  C8  2A  13  F0  ED
52  22  CF  F0  21  00  00  22
CD  F0  2A  CD  F0  22  CB  F0
2A  CB  F0  ED  5B  D3  F0  19
22  D1  F0  EB  21  00  00  19
19  19  E5  ED  5B  CB  F0  21
00  00  19  19  19  ED  4B  D5
F0  09  EB  E1  09  E5  D5  0E
00  7E  47  1A  B8  30  03  0E
01  47  AF  B0  28  19  C5  13
23  4E  23  46  C5  E1  EB  4E
```

Check all of the entries for accuracy. If you have made a mistake, use the arrow keys to move the cursor to the location of the error, then press the correct key. When all of the entries are correct, press **F2**.

Now type **M** then F080. This will move the display so that you can enter the rest of the program.

Press **F1** and make the following changes to memory.

```
23  46  C5  E1  C1  1A  96  38
0A  20  27  13  23  10  F6  CB
41  20  1F  D1  E1  06  03  4E
EB  7E  71  EB  77  23  13  10
F6  2A  D3  F0  EB  2A  CB  F0
AF  ED  52  22  CB  F0  30  90
18  02  D1  E1  2A  CD  F0  11
01  00  AF  19  22  CD  F0  ED
5B  CF  F0  ED  52  DA  3A  F0
C3  18  F0  00  00  00  00  00
21  00  F0  E5  2A  03  28  E9
```

Ensure all entries are correct and press **F2**.

Our machine language routine is now in memory, and we need to transfer a copy to disk. Press **S** to return to TRSDOS READY, then enter DEBUG OFF.

Now enter the following:

```
DUMP SORT/CIM START=F000, END=F0E0,
     TRA=F0D0, RORT=R
```

This will dump a copy of the routine to disk, under the file name SORT/CIM.

Enter BASIC using the following :

```
BASIC -M:61439
```

This loads BASIC and protects memory above 61439 for the machine language routine. In BASIC you can make the following changes to the Model I BASIC routine to see how the sort works. Delete line 10-70:

Add the following lines:

```
10  CLS
20  CLEAR 23000
30  DEFUSR =&HF0D0
40  DEFINT A-Z
50  DIM A$(1500) ,X (2)
60  Z=0
65  SYSTEM"LOAD SORT/CIM"
70  INPUT NUMBER OF ITEMS ◄ 1500";N
```

Change line 240 to read:

```
240  GOTO 10
```

Model II Mailing List for 80 Column Printers

By making the following changes to your Model II Mailing List (269-4506), you will be able to print labels and listings on an 80 column printer.

The only report which requires a 132 column printer is the file listing report.

NOTE: The line numbers used here are those from version 1.1 and may vary depending on the version that you are using.

Change these lines to read:

```
3100  LPRINT STRING$(80,“=”): LC=LC+1:
      RETURN
3350  IF OP$=“L” THEN 3170 ELSE PRINT
      D1$ TAB  (18)  D0$ TAB (37)  D2$ TAB
       (60) D3$
3360  LPRINT TAB (13) “Categories:”:
      FOR I=0 TO 7: IF(2 ^ IANDCVI(D7$))
      THEN LPRINT I+1;: NEXT ELSE NEXT
3370  LPRINT TAB (60) D4$
3380  LPRINT STRING$(80,“-”): LC=LC+4
      : IF LC►57 THEN LC=0 SYSTEM
      “FORMS T”
```

Add these lines:

```
3372  IF M8◄►0 THEN LPRINT TAB(18)
      “ Remarks:”;D8$
3374  LPRINT TAB(60) D5$ TAB(70) D6$
```

We have spread out the contents of the lines to make them more readable. In the program, most of the spaces between items on the program line are missing. Please note that a significant amount of material was deleted from lines 3350 and 3370. After you have made these changes, save a copy of the program under a file name of “MLS/BAS”

Mailing labels for Model I Accounts Receivable (269-1555)

The following program will allow users of Model I Accounts Receivable to use the customer information to create mailing labels. The mailing labels are available through any Tandy store as catalogue number 269-9305.

```
10  CLS:PRINT“*ARS MAILING
    LABELS*”
20  CLEAR 500
30  DEFINT A-Z
40  DIM P(500,1)
50  INPUT“BY: (0=LAST NAME,
    1=POSTCODE) ”;A
60  INPUT“GIVE ARS PASSWORD”;
    P$
70  PI$=“CUSINDEX.”+P$+“:1”
80  PD$=“CUSDATA.”+P$+“:1”
90  ON ERROR GOTO 1000
100  OPEN“I”,1,PI$
110  INPUT#1,Z,TN,Z,Z,Z,Z,Z,Z
120  N=TN-1
130  INPUT#1,Z,Z,Z,Z,Z,Z,Z,Z
140  FOR X=1 TO N
150  PRINT X;
160  INPUT#1,Y,P(X,0),P(X,1)
170  NEXT X: CLOSE
180  OPEN“R”,2,PD$
190  FOR X=1 TO N
200  J=P(X,A)
210  JR=INT((J-1)/2+1
220  JD=J-2*INT((J-1)/2)-1
230  FIELD 2,JD*127ASD$,11ASF$,
     13ASL$,17ASSA$,15ASST$,
     5ASZ$
240  GET 2,JR
250  LPRINT F$;“ ”;L$
260  LPRINT SA$
270  LPRINT ST$
280  LPRINT Z$
290  LPRINT “ ”
300  LPRINT “ ”
310  NEXT X: CLOSE
320  PRINT“*END OF JOB*”: END
1000  CLOSE: PRINT“ERROR” ;ERR/
      2+1
1010  PRINT“*JOB ABORTED*”: END
```

Budget Management for 80 column printers

We have had several requests for information on printing Model I Budget Management (269-1603) reports on an 80 column printer i.e. Quick Printer I (269-1153). The simplest method is to follow the instructions on page 59 of the manual. These instructions tell you how to send the video reports to the printer.

If you would like to be able to print all the information on your printer, you will need to edit lines 3110-3160, 3530, 3540 and 3900 in the “REPORTS” program.

Make the following changes to lines 3110 and 3120:

Within the quote marks, delete the first 2 blanks, move over 9 spaces, delete 2 blanks, move 26 spaces, delete 1 blank, move 12 spaces, delete 1 blank, move 12 more spaces, delete 12 blanks, move 10 spaces and delete 2 more blanks.

In lines 3130 and 3135 — within the quotes, move 6 spaces, delete 5 blanks, move 30 spaces, delete 5 more blanks.

In line 3140, inside the first and second sets of quotes, delete the first 2 blanks and the last 5 blanks. Change both occurrences of the number 15 to 8.

In line 3160 delete the 3 blanks inside both sets of quotes.

In line 3530, change the TAB(40) to read TAB(26). In the string statement (STRING$), change 25 to 22.

In line 3540 change the first TAB(40) to TAB(26), change TAB(45) to TAB(31).

In line 3900, change the number 110 to 80.

After you have made these changes, be sure and save a copy of the corrected program to either tape or disk as required.