6800/ICCD JOURNAL
INTERNATIONAL COMPUTER CENTER DIRECTOR

ICCD JOURNAL
The ICCD Journal is published four times yearly by the International Computer Center Director, P.O. Drawer 2790, Norman, Oklahoma 73070. The ICCD is a wholly owned division of Academic World Incorporated. Subscription rates are $18/year U.S. and Canada for membership. $21/year for countries outside the U.S. and Canada.

Opinions expressed by the authors are not necessarily thoses of the ICCD. Address all subscription correspondence to:

ICCD
P. O. Drawer 2790
Norman, Oklahoma 73070
Cable: ACAWORLD
405-364-1119

Articles and high quality manuscripts are welcomed from readers of ICCD. ICCD provides a free one-year membership to successful submitters.

Volume 1, No. 3 July, 1979

TABLE OF CONTENTS

A Note from the Publisher ..... *H. Zallen* **2**

Database Disk Oriented System ..... *J. Petty* **3**

XMON 6800 Extended Monitor ..... *W. Gilliland* **11**

Source for Port 2 Print Routine
Using an ACIA ..... *J. Petty & J. Waldvogel* **14**

File Edit Ver. 1.0 ..... *K. Erickson* **15**

COMING IN THE NEXT ISSUE

S. J. Carter Business System
Review of FLEX+++ 2.0
Escon Selectric Typewriter System
Intertube I, II Terminal
TSC Text Processor *Canned Programs*

LIST OF ADVERTISERS

Escon Products Inc. ..... Inside Front Cover
Integral Data Systems, Inc. ..... Back Cover
International Computer Center Director..... 1
On-Line 9
Smoke Signal Broadcasting..... 13
SSI Publications 20
Technical Systems Consultants, Inc..... 10, Inside Back Cover

Going dotty over the quality of your printing?

Any microcomputer can interface with any model IBM SELECTRIC®

Prices*

S-100 . . . . . $496.00	RS-232 . . . . . $549.00
Parallel $525.00	IEEE-488 . . . . . $560.00

*Prices valid in USA only.

Escon Products, Inc.
171 Mayhew Way, Suite 204,
Pleasant Hill, CA 94523
(415) 935-4590

SUBSCRIBE NOW

To the premier 6800 publication:

JOURNAL

INTERNATIONAL COMPUTER CENTER DIRECTOR

The *Journal* will be published quarterly and will feature programs, articles, and evaluations by some of the most prominent 6800 users.

Don't delay. Start or renew your subscription and begin receiving the new expanded *Journal*, and as a bonus, receive a subscription to the *Chip* — a 4 page newsletter published periodically throughout the year.

Name_____

Address_____

City_____

State_____ Zip_____

Country_____

☐ Start with Vol. 1, No. 1

Subscription rate is $18 per year in the U.S. and Canada. $21 per year outside the U.S. and Canada.

Send Remittance in U.S. funds to:

I C C D
P.O. Drawer 2790
Norman, Oklahoma 73070

Cable: **ACAWORLD**
Phone: **405-364-1119**

☐ New ☐ Renewal

A Note From The Publisher

Thanks for your patience in awaiting this issue of the *Journal*—we're sure you will find it well worth the wait. Thanks to some changes and additions you can now expect your *Journal* on a regular quarterly basis.

We would like to take this opportunity to announce some new staff members. Joining the *Journal* as Associate Editors are R. Lynn Smith (Balibago Double Standard, Vol. 1, No. 2) and William T. Gilliland whose XMON Monitor is featured in this issue. Coming on board as Assistant Editor is Jerome Laizure who is responsible for the new format of the *Journal.* The 6800 Advisory Board welcomes F. Emerson Brooks, Jr., D.Eng., (MEMTST—A Memory Program, Vol. 1, No. 1).

We at the *Journal* have re-emphasized our commitment to produce a quality publication regularly. The new format is just the first step. Starting with the next issue— when a suffcient amount of advertising is solicited— the *Journal* will expand to 32 pages.

With your continued support and a good response from potential advertisers we plan to make the *Journal* the premier 6800 publication. So renew your subscription today and pass along a copy of the subscription form to a friend, members of your local user's group, or other professionals at the office.

The *Journal* would also like to acknowledge the debut of the *68 Micro Journal* and wish publisher Don Williams, Sr. the best of luck and an offer to exchange subscriptions.

One last note. The *Journal* is always looking for high quality articles, programs, evaluations, etc. Send the manuscript along with program documentation to the address on the front cover.

Harold Zallen, Ph.D.
Publisher-Editor

Database Disk Oriented System©

James Petty, M.D.
1016 N.W. 41st Street
Oklahoma City, Oklahoma 73118

The 6800 ICCD feels that this Database Management System software is more than worth your entire subscription costs and frustrations you have had waiting for issues. In fact there are goodies in this package that far exceed the $95 Cromenco™ Database Management System lauded by Creative Computing *in a recent issue authored by John Craig. We should tell you at the outset we are giving you a single source license for this software as ICCD members. The software should be critiqued by you and comments sent to the 6800 ICCD. At a later date this will be marketed and sold to non-6800 ICCD members in a fashion similar to Cromenco and others. We sincerely hope you like it and use it productively.*

Editor

The Database Disk Oriented System is based on the concept that a program can serve multiple types of files. This is more desirable than constructing a different program for each different file or group of files. The concept of Gupta and Lander in their program *A Peoples Data Base System* inspired this system.

Database is a file system that can be used with Southwest Technical Products Corporation Minifloppy and the Smoke Signal Minifloppy System. The Database System supports files on the disk. It requires the use of Flex™ and SWPT™ BASIC Ver. 3 or Smoke Signal™ DOS 3.1 or 4.2.

With this system a file must contain at least one alpha character field but may contain as many more than one as required. It may contain zero numeric fields or as many as required.

A maximum of twenty fields may be used. However, by changing the DIM statement at LINE 15 the maximum number of fields may be altered.

The program is written in BASIC language. The Database possesses the capability to set up a new file to your specifications or to call in from the disk an old file previously set up on the Database system and allocate variables and memory space for this file.

When the program is first called it will prompt for the name of the file (this may be either a new or old file). It will then prompt **HOW MANY CHARACTER FIELDS.**If an old file, a reply of *0* (zero) will take you directly to the command request.

If a new file, enter the number of character (alpha) fields desired. The computer will prompt for the name of each field (try to keep to eight [8] letters, although more can be used). After all character fields have been named the computer will prompt for the number of numeric fields and then the name of each field. When this is completed there will be a **COMMAND** prompt.

The commands are:

1. **HELP**—Lists all valid commands.
2. **ADD**—Adds data to the file (either old or new file).
3. **DELETE**—This will flag a string of data so that it will not be used during list or label commands but is still in the file.
4. **RESTORE**—Returns all deleted files to the active file.
5. **CLIST**—Lists in compressed form all data strings regardless of the delete status.
6. **LIST**—This lists data with the name of the field of all data that has no delete flag set.
7. **END**—Ends the program and returns to BASIC.
8. **LABEL**—This program allows you to print labels from the data that has no delete flag set. It will prompt for the number of lines for the label (maximum is five [5]). It will then prompt for the name of the field for each line, also one (1) line may contain two (2) fields (such as first and last name). It also prints an alignment string for proper setting of the label.
9. **RUBOUT**—This program erases from the data file all records that have been flagged with the delete flag.

10. **STATS**—This program allows numeric fields to be summed, averaged, the variance and standard deviation found and the co-variance of two numeric fields to be found.
11. **SORT**—This program uses the Shell-Metzer ascending or descending on any field either alpha or numeric. The program will prompt for the number of items to sort at one time. This allows you to tailor the program to available memory. The larger the number of items sorted at one time the faster the sort but the more memory is used.
12. **PRINT**—This program outputs to the port (for hard copy) of your choice the data in the file but does not include the filed names.
13. **SEARCH**—This program searches the file for a specific item and outputs to the port of your choice. It will first ask what field to search and then what item in the field to look for.
14. **APPEND**—This program requires some internal changes depending on your particular file.

    The object is to take a file on the disk that was not generated by this program and convert it to a file that will run on this program. To do this it is necessary to change LINE 4445 and LINE 4460 through 4480 to fit your program.

    To use this program setup to database the usual way by entering the new program name, the number of character (alpha) fields and their name, and the number of numeric fields and their names when prompted by the program. Do not enter any data at this time. Then with **APPEND** program appended to Database and changes in the program made, call for **APPEND** and the program will enter the data from the old to the new program.
15. **CHANGE**—This program allows you to change or correct the spelling of any field name. It will prompt for input needed.

```
0010 LINE= 80
0020 REM DISK ORIENTED DATA BASE
0030 DIM N$(20)
0040 INPUT "NAME OF FILE",F$
0050 LET G$="1.SCRATCH.DAT"
0060 PRINT "DEFINE DATABASE STRUCTURE"
0070 PRINT "HOW MANY CHARACTER FIELDS"
0080 INPUT T
0090  M1=T
0100 IF M1<1 THEN 270
0110 FOR I=1 TO M1
0120 PRINT "WHAT IS THE NAME OF FIELD";I
0130 INPUT T$
0140  N$(I)=T$
0150 NEXT I
0160 PRINT "HOW MANY NUMERIC FIELDS"
0170 INPUT T
0180  M2=T
0190 IF M2=0 THEN 250
0200 FOR I=1 TO M2
0210 PRINT "WHAT IS THE NAME OF FIELD";I
0220 INPUT T$
0230  N$(I+M1)=T$
0240 NEXT I
0250 PRINT "STRUCTURE DEFINITION COMPLETE"
0260 GOSUB 2460
0270 PRINT "COMMAND"
0280 INPUT T$
0290 RESTORE
0300 READ Z$,T
0310 IF Z$="##" THEN 270
0320 IF LEFT$(Z$,3)<>LEFT$(T$,3) THEN 300
0330 IF T>8 GOTO360
0340 ON T GOSUB 670,900,1080,1540,1740,1980,2070,2090
0350 GOTO 270
0360 ON T-8 GOTO 580,590,600,610,620,630,640,650,660
0370 DATA ADD,1
0380 DATA CLIST,5
0390 DATA LIST,2
0400 DATA RUBOUT,8
0410 DATA DELETE,3
0420 DATA RESTORE,4
0430 DATA HELP,6
0440 DATA END,7
0450 DATA AVERAGE,9
0460 DATA SUM,9
0470 DATA VARIANCE,9
0480 DATA C-VARIANCE,9
0490 DATA LABELS,10
0500 DATA SORT,11
0510 DATA PRINT,12
0520 DATA SEARCH,13
0530 DATA APPEND,14
0540 DATA CHANGE,15
0550 DATA MISSCHECK,16
0560 DATA REPAIR,17
0570 DATA ##,-1
0580 CHAIN MATHC
0590 CHAIN LABELC
0600 CHAIN SORTC
0610 CHAIN PRINTC
0620 CHAIN SEARCHC
0630 CHAIN APPENDC
0640 CHAIN CHANGEC
0650 CHAIN PMISCKC
0660 CHAIN REPAIRC
0670 REM ADDS TO DATABASE
0680 LET Z=0
0690 OPEN #1,F$
0700 OPEN #2,G$
0710 GOSUB 2340
0720 GOSUB 2400
0730 GOSUB 2530
0740 FOR I=1 TO M1
0750 PRINT N$(I)
0760 INPUT A$(I)
0770 IF A$(I)="STOP" THEN 880
0780 IF I=1 THEN WRITE#2,Z,A$(I): GOTO800
0790 WRITE #2,A$(I)
0800 NEXT I
0810 IF M2=0 THEN 870
0820 FOR I=1 TO M2
0830 PRINT N$(M1+I)
0840 INPUT A(I)
0850 WRITE #2,A(I)
0860 NEXT I
0870 GOTO 740
0880 CLOSE #1,#2:KILLF$:RENAMEG$,F$
0890 RETURN
0900 REM LISTS
0910 OPEN #1,F$
0920 INPUT "PRINT PORT",P
0930 GOSUB 2340
0940 PORT= P
0950 FOR I=1 TO M1
0960 IF I=1 THEN READ#1,Z,A$(I): GOTO980
0970 READ #1,A$(I)
0980 IF EOF(1)=1 THEN 1070
0990 IF Z=0 THEN PRINTN$(I);TAB(15);A$(I)
1000 NEXT I
1010 IF M2=0 THEN 1060
1020 FOR I=1 TO M2
1030 READ #1,A(I)
1040 IF Z=0 THEN PRINTN$(I+M1);TAB(15);A(I)
1050 NEXT I
1060  Z=0:PRINT:GOTO950
1070 PORT= 1:CLOSE#1:RETURN
1080 REM DELETES A RECOED OR GROUP OF RECORDS
1090 OPEN #1,F$
1100 GOSUB 2340
1110 GOSUB 2670
1120 INPUT "WHICH FIELD TO DELETE ON(1,2,3,ETC)",P
1130 LET B=P
1140 PRINT "IN ";N$(P);" WHICH ITEM"
1150 INPUT T$
1160 OPEN #2,G$
1170 GOSUB 2400
1180 FOR I=1 TO M1
1190 IF I=1 THEN READ#1,Z,A$(I):GOTO1210
1200 READ #1,A$(I)
1210 IF EOF(1)=1 THEN 1500
1220 NEXT I
1230 IF M2=0 THEN 1270
1240 FOR I=1 TO M2
1250 READ #1,A(I)
1260 NEXT I
1270 IF B>M1 THEN C=B-M1: GOTO1300
1280 IF T$=A$(B) THEN 1400
1290 GOTO 1310
1300 IF VAL(T$)=A(C) THEN 1400
1310 FOR I=1 TOM1
1320 IF I=1 THENWRITE#2,Z,A$(I): GOTO1340
1330 WRITE #2,A$(I)
1340 NEXT I
1350 IF M2=0 THEN 1390
1360 FOR I=1 TO M2
1370 WRITE #2,A(I)
1380 NEXT I
1390 GOTO 1180
1400  Z=1
1410 FOR I=1 TO M1
1420 IF I=1 THEN WRITE#2,Z,A$(I): GOTO1440
```

```
1430 WRITE #2,A$(I)
1440 NEXT I
1450 IF M2=0 THEN 1490
1460 FOR I=1 TO M2
1470 WRITE #2,A(I)
1480 NEXT I
1490 GOTO 1180
1500 CLOSE #1,#2
1510 KILL F$
1520 RENAME G$,F$
1530 RETURN
1540 REM RESTORES ALL DELETED ITEMS
1550 OPEN #1,F$,#2,G$
1560 GOSUB 2340
1570 GOSUB 2400
1580 FOR I=1 TO M1
1590 IF I=1 THEN READ#1,Z,A$(I): GOTO1610
1600 READ #1,A$(I)
1610 IF EOF(1)=1 THEN 1720
1620  Z=0
1630 IF I=1 THEN WRITE#2,Z,A$(I): GOTO 1650
1640 WRITE #2,A$(I)
1650 NEXT I
1660 IF M2=0 THEN 1710
1670 FOR I=1 TO M2
1680 READ #1,A(I)
1690 WRITE #2,A(I)
1700 NEXT I
1710 GOTO 1580
1720 CLOSE #1,#2:KILLF$
1730 RENAME G$,F$:RETURN
1740 REM COMPRESSED LIST OF ALL THE FILE
1750 OPEN #1,F$
1760 INPUT "PRINT PORT",P
1770 GOSUB 2340
1780 PORT= P
1790 FOR I=1 TO M1
1800 IF I=1 THEN READ#1,Z,A$(I):GOTO1820
1810 READ #1,A$(I)
1820 IF EOF(1)=1 THEN 1970
1830 NEXT I
1840 IF M2=0 THEN 1880
1850 FOR I=1 TO M2
1860 READ #1,A(I)
1870 NEXT I
1880 FOR I=1 TO M1
1890 PRINT A$(I);" ";
1900 NEXT I
1910 IF M2=0 THEN 1950
1920 FOR I=1 TO M2
1930 PRINT A(I);" ";
1940 NEXT I
1950 PRINT
1960 GOTO 1790
1970 PORT= 1:CLOSE#1:RETURN
1980 REM HELP ROUTINE
1990 PRINT "VALID COMMANDS ARE"
2000 RESTORE
2010 READ Z$,T
2020 FOR I=1 TO M2
2030 IF Z$="###" THEN 2060
2040 PRINT Z$
2050 GOTO 2010
2060 RETURN
2070 REM ENDS PROGRAM
2080 END
2090 REM REMOVES ALL DELETED ITEMS FROM THE FILE
2100 OPEN #1,F$,#2,G$
2110 GOSUB 2340
2120 GOSUB 2400
2130 FOR I=1 TO M1
2140 IF I=1 READ #1,Z,A$(I): GOTO2160
2150 READ #1,A$(I)
2160 IF EOF(1)=1 THEN 2320
2170 NEXT I
2180 IF M2=0 THEN 2220
2190 FOR I=1 TO M2
2200 READ #1,A(I)
2210 NEXT I
2220 IF Z=1 GOTO2130
2230 FOR I=1 TO M1
2240 IF I=1 THEN WRITE#2,Z,A$(I):GOTO2260
2250 WRITE #2,A$(I)
2260 NEXT I
2270 IF M2=0 THEN 2310
2280 FOR I=1 TO M2
2290 WRITE #2,A(I)
2300 NEXT I
2310 GOTO 2130
2320 CLOSE #1,#2:KILLF$
2330 RENAME G$,F$:RETURN
2340 REM PUTS VARIABLES INTO MEMORY
2350 READ #1,M1,M2
2360 FOR I=1 TO M1+M2
2370 READ #1,N$(I)
2380 NEXT I
2390 RETURN
2400 REM PUTS FIELD NAMES ON DISK
2410 WRITE #2,M1,M2
2420 FOR I=1 TO M1+M2
2430 WRITE #2,N$(I)
2440 NEXT I
2450 RETURN
2460 OPEN #1,F$
2470 WRITE #1,M1,M2
2480 FOR I=1 TO M1+M2
2490 WRITE #1,N$(I)
2500 NEXT I
2510 CLOSE #1
2520 RETURN
2530 FOR I=1 TO M1
2540 IF I=1 THEN READ#1,Z,A$(I):GOTO2560
2550 READ #1,A$(I)
2560 IF EOF(1)=1 THEN 2660
2570 IF I=1 THEN WRITE#2,Z,A$(I):GOTO2590
2580 WRITE #2,A$(I)
2590 NEXT I
2600 IF M2=0 THEN 2650
2610 FOR I=1 TO M2
2620 READ #1,A(I)
2630 WRITE #2,A(I)
2640 NEXT I
2650 GOTO 2530
2660  Z=0:RETURN
2670 FOR I=1 TO M1+M2
2680 PRINT I;" ";N$(I)
2690 NEXT I
2700 RETURN
0010 REM ***LABELS***
0020 LINE= 80
0030 INPUT "NAME OF FILE",F$
0040 OPEN #1,F$
0050  C=0:D=0
0060 GOSUB 710
0070 GOSUB 770
0080 INPUT "LINES PER LABEL(MAX IS 5)",T
0090  B=T
0100 FOR I=1 TO T
0110 IF D=1 GOTO 150
0120 PRINT "IS LINE ";I;" COMBINED(Y OR N)"
0130 INPUT T$
0140 IF T$="Y" GOTO 600
0150 PRINT "WHAT IS NUMBER(1,2,ETC) OF FIELD OF LINE";I
0160 INPUT R
0170  V(I)=R
0180 NEXT I
0190 PRINT "PLEASE POSITION FORM"
0200 INPUT "1 FOR ALLIGNMENT OR 2 FOR BEGIN",T
0210 INPUT " PRINT PORT",P
0220 IF T=2 GOTO 340
0230 PORT= P
0240 FOR I=1 TO B
0250 PRINT "<-------ALIGNMENT------->"
0260 NEXT I
0270 IF B>4 GOTO310
0280 FOR I=B TO4
0290 PRINT
0300 NEXT I
0310 PRINT
0320 PORT= 1
0330 GOTO 200
0340 FOR I=1 TO M1
0350 IF I=1 READ#1,Z,A$(I):GOTO370
0360 READ #1,A$(I)
0370 IF EOF(1)=1 THEN 570
0380 NEXT I
0390 IF M2=0 THEN430
0400 FOR I=M1+1 TO M2
0410 READ #1,A(I)
0420 NEXT I
0430 IF Z=1 GOTO340
0440 PORT= P
0450 IF Z=1 GOTO 340
0460 FOR I=1 TO B
0470 IF C=I GOTO660
0480 IF V(I)<M1+1 THEN PRINTA$(V(I)):GOTO500
0490 PRINT A(V(I))
0500 NEXT I
0510 IF B>4 GOTO 550
0520 FOR R=B TO 4
0530 PRINT
0540 NEXT R
0550 PRINT
0560 GOTO 340
0570 PORT= 1
0580 CLOSE #1
0590 CHAIN DATABASE
0600  C=I:D=1
0610 INPUT "WHAT IS THE NUMBER(1,2,ETC) OF THE 1ST FIELD",R1
0620  C(R1)=R1
0630 INPUT "WHAT IS THE NUMBER(1,2,ETC) OF THE 2ND FIELD",R2
0640  C(R2)=R2
0650 GOTO 180
0660 IF C(R1)<M1+1 THEN PRINTA$(C(R1));" ";: GOTO680
0670 PRINT A(C(R1));" ";
0680 IF C(R2)<M1+1 THEN PRINTA$(C(R2)):GOTO700
0690 PRINT A(C(R2))
0700 GOTO 500
0710 REM PUTS PARAMETERS OF PROGRAM INTO MEMORY
0720 READ #1,M1,M2
0730 FOR I=1 TO M1+M2
0740 READ #1,N$(I)
0750 NEXT I
0760 RETURN
0770 FOR I=1 TO M1+M2
0780 PRINT I;" ";N$(I)
0790 NEXT I
0800 RETURN

0010 REM MATH PACK
0020 LINE= 80
0030 INPUT " NAME OF FILE",F$
0040 OPEN #1,F$
0050 GOSUB 950
0060 PRINT "WHICK ITEM DO YOU WANT"
0070 PRINT "1)SUM OF A FIELD"
0080 PRINT "2)AVREAGE OF A FIELD"
0090 PRINT "3)VARIANCE OF A FIELD"
0100 PRINT "4)COVARIANCE OF TWO FIELDS"
0110 PRINT "5)END MATHFUNCTIONS"
0120 INPUT X
0130 ON X GOSUB 410,510,610,730,930
0140 GOTO 60
0150 REM **ROUTINE TO SUM FIELDS**
0160  T1=0:T2=0:T3=0:T5=0:T6=0:T7=0
0170  Z=0
```

```
0180 RESTORE #1
0185 GOSUB 950
0190 FOR I=1 TO M1
0200 IF I=1 READ #1,Z,A$(I):GOTO220
0210 READ #1,A$(I)
0220 IF EOF(1)=1 GOTO 350
0230 NEXT I
0240 FOR I=1 TO M2
0250 READ #1,A(I)
0260 NEXT I
0270 IF Z=1 GOTO 190
0280  T1=T1+A(T)
0290  T2=T2+(A(T)*A(T4))
0300  T3=T3+1
0310  T5=T5+A(T4)
0320  T6=T6+(A(T)*A(T))
0330  T7=T7+(A(T4)*A(T4))
0340 GOTO 190
0350 RETURN
0360 GOSUB 1010
0370 INPUT "NUMBER OF FIELD(1,2,3,ETC)",I
0380 IF I<M1+1 PRINT"ONLY NUMERIC FIELDS": GOTO360
0390 LET A=I
0400 RETURN
0410 REM ***SUM***
0420 GOSUB 360
0430 INPUT "PRINT PORT",P
0440  T=I-M1
0450  T4=T
0460 GOSUB 150
0470 PORT= P
0480 PRINT "THE SUM OF ";N$(A);" = ";T1
0490 PORT= 1
0500 RETURN
0510 REM ***AVERAGE***
0520 GOSUB 360
0530 INPUT "PRINT PORT",P
0540  T=I-M1
0550  T4=T
0560 GOSUB 150
0570 PORT= P
0580 PRINT "THE AVERAGE OF ";N$(A);" = ";T1/T3
0590 PORT= 1
0600 RETURN
0610 REM ***VARIANCE***
0620 GOSUB 360
0630 INPUT "PRINT PORT",P
0640  T=I-M1
0650  T4=T
0660 GOSUB 150
0670 PORT= P
0680 LET Y=(T6-((T1*T1)/T3))/(T3-1)
0690 PRINT "MEAN","VARIANCE","STANDARD DEVIATION"
0700 PRINT T1/T3,Y,SQR(Y)
0710 PORT= 1
0720 RETURN
0730 REM ***COVARIANCE***
0740 PRINT "FOR FIRST VARIATE"
0750 GOSUB 360
0760  T4=I-M1
0770 PRINT "FOR SECOND VATIATE"
0780 GOSUB 360
0790  T=I-M1
0800 GOSUB 150
0810 INPUT "PRINT PORT",P
0820 LET C1=(T2-((T1*T5)/T3))/(T3-1)
0830 LET C2=(T2-((T1*T5)/T3))/T3
0840 LET V1=(T6-((T1*T1)/T3))/(T3-1)
0850 LET V2=(T7-((T5*T5)/T3))/(T3-1)
0860 PORT= P
0870 PRINT "    ";"MEAN","VARIANCE","ST.DEVIATION","COVARIANCE"
0880 PRINT "IST ";T5/T3,V2,SQR(V2),C2
0890 PRINT "2ND ";T1/T3,V1,SQR(V1),C1
0900 PRINT "CORRELATION COEFFICIENT= ";C1/(SQR(V1)*SQR(V2))
0910 PORT= 1
0920 RETURN
0930 CLOSE #1
0940 CHAIN DATABASE
0950 REM PUTS PARAMETERS OF PROGRAM INTO MEMORY
0960 READ #1,M1,M2
0970 FOR I=1 TO M1+M2
0980 READ #1,N$(I)
0990 NEXT I
1000 RETURN
1010 FOR I=1 TO M1+M2
1020 PRINT I;" ";N$(I)
1030 NEXT I
1040 RETURN
```

```
0010 REM PRINT OUTPUTS TO PORT OF CHOICE
0020 INPUT "NAME OF FILE",F$
0030 OPEN #1,F$
0040 GOSUB 300
0050 INPUT "PRINT PORT",P
0060 LET Z=0
0070 FOR I=1 TO M1
0080 IF I=1 READ#1,Z,A$(I): GOTO 100
0090 READ #1,A$(I)
0100 IF EOF(1)=1 GOTO 280
0110 NEXT I
0120 IF M2=0 THEN 160
0130 FOR I=1 TO M2
0140 READ #1,A(I)
0150 NEXT I
0160 IF Z=1 GOTO 70
0170 PORT= P
0180 FOR I=1 TO M1
0190 PRINT A$(I)
0200 NEXT I
0210 IF M2=0 GOTO 250
0220 FOR I=1 TO M2
0230 PRINT A(I)
0240 NEXT I
0250 PRINT
0260 PORT= 1
0270 GOTO 70
0280 CLOSE #1
0290 CHAIN DATABASE
0300 REM PUTS PARAMETERS OF PROGRAM INTO MEMORY
0310 READ #1,M1,M2
0320 FOR I=1 TO M1+M2
0330 READ #1,N$(I)
0340 NEXT I
0350 RETURN

0010 REM SORT BY SHELL-METZER TECHNIQUE
0020 INPUT "NAME OF FILE",F$
0030 OPEN #1,F$
0040  D=0:C=0
0050 GOSUB 2070
0060 FOR I=1 TO M1
0070 IF I=1 THEN READ#1,Z,A$(I): GOTO 90
0080 READ #1,A$
0090 IF EOF(1)=1 THEN 170
0100 NEXT I
0110  D=D+1
0120 IF M2=0 THEN 160
0130 FOR I=1 TO M2
0140 READ #1,A(I)
0150 NEXT I
0160 GOTO 60
0170 LET H$="1.WRK.DAT"
0180 LET G$="1.SCRATCH.DAT"
0190 LET I$="1.WRK1.DAT"
0200 PRINT "THERE ARE ";D;" ITEMS TO SORT"
0210 RESTORE #1
0220 IF F=1 GOTO270
0230 INPUT "NUMBER OF ITEMS TO SORT AT ONE TIME",E
0240 IF M=0 THEN DIMC$(M1,E),Z2(E):GOTO260
0250 DIM C$(M1,E),C1(M2,E),Z2(E)
0260  F=1
0270 GOSUB 2070
0280 LET B1=INT(D/E)
0290 IF D/E<>INT(D/E) THEN B1=B1+1
0300 GOSUB 2130
0310 INPUT "FIELD TO SORT ON(1,2,3,ETC)",R
0320 IF R>M1 THEN Y=1: GOTO 340
0330  Y=2
0340 INPUT "ASCENDING OR DESCENDING(A OR D)",T$
0350 IF R>M1 THEN R=R-M1
0360  T=R:X=0
0370 IF T$="D" THEN X=1
0380  Q=0
0390 OPEN #2,G$
0400 FOR I=1 TO E
0410 FOR R=1 TO M1
0420 IF R=1 READ#1,Z2(I),C$(R,I): GOTO440
0430 READ #1,C$(R,I)
0440 IF EOF(1)=1 THEN920
0450 NEXT R
0460 IF M2=0 THEN 500
0470 FOR R=1 TO M2
0480 READ #1,C1(R,I)
0490 NEXT R
0500  Q=Q+1
0510 NEXT I
0520  C=C+1
0530 LET M=Q
0540 LET M=INT(M/2)
0550 IF M=0 GOTO920
0560 LET J=1
0570 LET K=Q-M
0580 LET I=J
0590 LET L=I+M
0600 IF Y=1 GOTO 690
0610 IF Y=2 GOTO 740
0620 GOSUB 790
0630 LET I=I-M
0640 IF I<1 THEN 660
0650 GOTO 590
0660 LET J=J+1
0670 IF J>K THEN 540
0680 GOTO 580
0690 IF X=1 GOTO 720
0700 IF C1(T,I)<=C1(T,L) THEN 660
0710 GOTO 620
0720 IF C1(T,I)>=C1(T,L) THEN660
0730 GOTO 620
0740 IF X=1 GOTO 770
0750 IF C$(T,I)<=C$(T,L) THEN 660
0760 GOTO 620
0770 IF C$(T,I)>= C$(T,L) THEN 660
0780 GOTO 620
0790 FOR R=1 TO M1
0800 IF R=1 THEN P=Z2(I):Z2(I)=Z2(L):Z2(L)=P
0810 LET T$=C$(R,I)
0820 LET C$(R,I)=C$(R,L)
0830 LET C$(R,L)=T$
0840 NEXT R
0850 IF M2=0 THEN 910
0860 FOR R=1 TO M2
0870  T2=C1(R,I)
0880  C1(R,I)=C1(R,L)
0890  C1(R,L)=T2
0900 NEXT R
0910 RETURN
0920 WRITE #2,M1,M2
0930 FOR I=1 TO M1+M2
0940 WRITE #2,N$(I)
0950 NEXT I
0960 FOR I=1 TO Q
0970 FOR R=1 TO M1
0980 IF R=1 WRITE#2,Z2(I),C$(R,I):GOTO1000
0990 WRITE #2,C$(R,I)
1000 NEXT R
1010 IF M2=0 THEN 1050
```

```
1020 FOR R=1 TO M2
1030 WRITE #2,C1(R,I)
1040 NEXT R
1050 NEXT I
1060 IF C=1 GOTO 1110
1070 IF C=2 GOTO 1150
1080 GOTO 1140
1090 IF C=B1 GOTO 1240
1100 GOTO 380
1110 CLOSE #2
1120 RENAME G$,I$
1130 GOTO 1090
1140 REM
1150 RENAME I$,H$
1160 OPEN #3,H$
1170 RESTORE #2
1180 GOSUB 1280
1190 CLOSE #2,#3,#4
1200 KILL G$
1210 KILL H$
1220 GOTO 1090
1230 IF C=1 CLOSE#1:GOTO 1250
1240 CLOSE #1
1250 KILL F$
1260 RENAME I$,F$
1270 CHAIN DATABASE
1280 OPEN #4,I$
1290 READ #2,M1,M2
1300 WRITE #4,M1,M2
1310 FOR I=1 TO M1+M2
1320 READ #2,N$(I)
1330 WRITE #4,N$(I)
1340 NEXT I
1350  T1=0:T2=0:R=T
1360 READ #3,M1,M2
1370 FOR I=1 TO M1+M2
1380 READ #3 N$(I)
1390 NEXT I
1400 GOSUB 1690
1410 IF T1=1 GOTO 1610
1420 GOSUB 1790
1430 IF T2=1 GOTO 1650
1440 IF Y=1 GOTO 1460
1450 IF Y=2 GOTO1560
1460 IF X=1 GOTO 1540
1470 IF A(R)<=A1(R) GOTO1500
1480 GOSUB 1980
1490 GOTO 1420
1500 GOSUB 1890
1510 GOSUB 1690
1520 IF T1=1 GOTO 1610
1530 GOTO 1440
1540 IF A(R)>=A1(R) GOTO1500
1550 GOTO 1480
1560 IF X=1 GOTO 1590
1570 IF A$(R)<=B$(R) GOTO 1500
1580 GOTO 1480
1590 IF A$(R)>=B$(R) GOTO1500
1600 GOTO 1480
1610 IF T2=1 RETURN
1620 GOSUB 1980
1630 GOSUB 1790
1640 GOTO 1610
1650 IF T1=1 RETURN
1660 GOSUB 1890
1670 GOSUB 1690
1680 GOTO 1650
1690 FOR I=1 TO M1
1700 IF I=1 READ#2,Z,A$(I):GOTO1720
1710 READ #2,A$(I)
1720 IF EOF(2)=1 THEN T1=1: GOTO1780
1730 NEXT I
1740 IF M2=0 GOTO 1780
1750 FOR I=1 TO M2
1760 READ #2,A(I)
1770 NEXT I
1780 RETURN
1790 FOR I=1 TO M1
1800 IF I=1 READ#3,Z1,B$(I):GOTO1820
1810 READ #3,B$(I)
1820 IF EOF(3)=1 THEN T2=1: GOTO1880
1830 NEXT I
1840 IF M2=0 GOTO 1880
1850 FOR I=1 TO M2
1860 READ #3,A1(I)
1870 NEXT I
1880 RETURN
1890 FOR I=1 TO M1
1900 IF I=1 WRITE#4,Z,A$(I):GOTO 1920
1910 WRITE #4,A$(I)
1920 NEXT I
1930 IF M2=0 GOTO 1970
1940 FOR I=1 TO M2
1950 WRITE #4,A(I)
1960 NEXT I
1970 RETURN
1980 FOR I=1 TO M1
1990 IF I=1 WRITE#4,Z1,B$(I): GOTO2010
2000 WRITE #4,B$(I)
2010 NEXT I
2020 IF M2=0 GOTO2060
2030 FOR I=1 TO M2
2040 WRITE #4,A1(I)
2050 NEXT I
2060 RETURN
2070 REM PUTS PARAMETERS OF PROGRAM INTO MEMORY
2080 READ #1,M1,M2
2090 FOR I=1 TO M1+M2
2100 READ #1,N$(I)
2110 NEXT I
2120 RETURN
2130 FOR I=1 TO M1+M2
2140 PRINT I;" ";N$(I)
2150 NEXT I
2160 RETURN
0010 REM SORT BY SHELL-METZER TECHNIQUE
0020 INPUT "NAME OF FILE",F$
0030 OPEN #1,F$
0040  D=0:C=0
0050 GOSUB 2070
0060 FOR I=1 TO M1
0070 IF I=1 THEN READ#1,Z,A$(I): GOTO 90
0080 READ #1,A$
0090 IF EOF(1)=1 THEN 170
0100 NEXT I
0110  D=D+1
0120 IF M2=0 THEN 160
0130 FOR I=1 TO M2
0140 READ #1,A(I)
0150 NEXT I
0160 GOTO 60
0170 LET H$="1.WRK.DAT"
0180 LET G$="1.SCRATCH.DAT"
0190 LET I$="1.WRK1.DAT"
0200 PRINT "THERE ARE ";D;" ITEMS TO SORT"
0210 RESTORE #1
0220 IF F=1 GOTO270
0230 INPUT "NUMBER OF ITEMS TO SORT AT ONE TIME",E
0240 IF M=0 THEN DIMC$(M1,E),Z2(E):GOTO260
0250 DIM C$(M1,E),C1(M2,E),Z2(E)
0260  F=1
0270 GOSUB 2070
0280 LET B1=INT(D/E)
0290 IF D/E<>INT(D/E) THEN B1=B1+1
0300 GOSUB 2130
0310 INPUT "FIELD TO SORT ON(1,2,3,ETC)",R
0320 IF R>M1 THEN Y=1: GOTO 340
0330  Y=2
0340 INPUT "ASCENDING OR DESCENDING(A OR D)",T$
0350 IF R>M1 THEN R=R-M1
0360  T=R:X=0
0370 IF T$="D" THEN X=1
0380  Q=0
0390 OPEN #2,G$
0400 FOR I=1 TO E
0410 FOR R=1 TO M1
0420 IF R=1 READ#1,Z2(I),C$(R,I): GOTO440
0430 READ #1,C$(R,I)
0440 IF EOF(1)=1 THEN520
0450 NEXT R
0460 IF M2=0 THEN 500
0470 FOR R=1 TO M2
0480 READ #1,C1(R,I)
0490 NEXT R
0500  Q=Q+1
0510 NEXT I
0520  C=C+1
0530 LET M=Q
0540 LET M=INT(M/2)
0550 IF M=0 GOTO920
0560 LET J=1
0570 LET K=Q-M
0580 LET I=J
0590 LET L=I+M
0600 IF Y=1 GOTO 690
0610 IF Y=2 GOTO 740
0620 GOSUB 790
0630 LET I=I-M
0640 IF I<1 THEN 660
0650 GOTO 590
0660 LET J=J+1
0670 IF J>K THEN 540
0680 GOTO 580
0690 IF X=1 GOTO 720
0700 IF C1(T,I)<=C1(T,L) THEN 660
0710 GOTO 620
0720 IF C1(T,I)>=C1(T,L) THEN660
0730 GOTO 620
0740 IF X=1 GOTO 770
0750 IF C$(T,I)<=C$(T,L) THEN 660
0760 GOTO 620
0770 IF C$(T,I)>= C$(T,L) THEN 660
0780 GOTO 620
0790 FOR R=1 TO M1
0800 IF R=1 THEN P=Z2(I):Z2(I)=Z2(L):Z2(L)=P
0810 LET T$=C$(R,I)
0820 LET C$(R,I)=C$(R,L)
0830 LET C$(R,L)=T$
0840 NEXT R
0850 IF M2=0 THEN 910
0860 FOR R=1 TO M2
0870  T2=C1(R,I)
0880  C1(R,I)=C1(R,L)
0890  C1(R,L)=T2
0900 NEXT R
0910 RETURN
0920 WRITE #2,M1,M2
0930 FOR I=1 TO M1+M2
0940 WRITE #2,N$(I)
0950 NEXT I
0960 FOR I=1 TO Q
0970 FOR R=1 TO M1
0980 IF R=1 WRITE#2,Z2(I),C$(R,I):GOTO1000
0990 WRITE #2,C$(R,I)
1000 NEXT R
1010 IF M2=0 THEN 1050
1020 FOR R=1 TO M2
1030 WRITE #2,C1(R,I)
1040 NEXT R
1050 NEXT I
1060 IF C=1 GOTO 1110
1070 IF C=2 GOTO 1150
1080 GOTO 1140
1090 IF C=B1 GOTO 1240
1100 GOTO 380
1110 CLOSE #2
1120 RENAME G$,I$
```

```
1130 GOTO 1090
1140 REM
1150 RENAME I$,H$
1160 OPEN #3,H$
1170 RESTORE #2
1180 GOSUB 1280
1190 CLOSE #2,#3,#4
1200 KILL G$
1210 KILL H$
1220 GOTO 1090
1230 IF C=1 CLOSE#1:GOTO 1250
1240 CLOSE #1
1250 KILL F$
1260 RENAME I$,F$
1270 CHAIN DATABASE
1280 OPEN #4,I$
1290 READ #2,M1,M2
1300 WRITE #4,M1,M2
1310 FOR I=1 TO M1+M2
1320 READ #2,N$(I)
1330 WRITE #4,N$(I)
1340 NEXT I
1350  T1=0:T2=0:R=T
1360 READ #3,M1,M2
1370 FOR I=1 TO M1+M2
1380 READ #3 N$(I)
1390 NEXT I
1400 GOSUB 1690
1410 IF T1=1 GOTO 1610
1420 GOSUB 1790
1430 IF T2=1 GOTO 1650
1440 IF Y=1 GOTO 1460
1450 IF Y=2 GOTO1560
1460 IF X=1 GOTO 1540
1470 IF A(R)<=A1(R) GOTO1500
1480 GOSUB 1980
1490 GOTO 1420
1500 GOSUB 1890
1510 GOSUB 1690
1520 IF T1=1 GOTO 1610
1530 GOTO 1440
1540 IF A(R)>=A1(R) GOTO1500
1550 GOTO 1480
1560 IF X=1 GOTO 1590
1570 IF A$(R)<=B$(R) GOTO 1500
1580 GOTO 1480
1590 IF A$(R)>=B$(R) GOTO1500
1600 GOTO 1480
1610 IF T2=1 RETURN
1620 GOSUB 1980
1630 GOSUB 1790
1640 GOTO 1610
1650 IF T1=1 RETURN
1660 GOSUB 1890
1670 GOSUB 1690
1680 GOTO 1650
1690 FOR I=1 TO M1
1700 IF I=1 READ#2,Z,A$(I):GOTO1720
1710 READ #2,A$(I)
1720 IF EOF(2)=1 THEN T1=1: GOTO1780
1730 NEXT I
1740 IF M2=0 GOTO 1780
1750 FOR I=1 TO M2
1760 READ #2,A(I)
1770 NEXT I
1780 RETURN
1790 FOR I=1 TO M1
1800 IF I=1 READ#3,Z1,B$(I):GOTO1820
1810 READ #3,B$(I)
1820 IF EOF(3)=1 THEN T2=1: GOTO1880
1830 NEXT I
1840 IF M2=0 GOTO 1880
1850 FOR I=1 TO M2
1860 READ #3,A1(I)
1870 NEXT I
1880 RETURN
1890 FOR I=1 TO M1
1900 IF I=1 WRITE#4,Z,A$(I):GOTO 1920
1910 WRITE #4,A$(I)
1920 NEXT I
1930 IF M2=0 GOTO 1970
1940 FOR I=1 TO M2
1950 WRITE #4,A(I)
1960 NEXT I
1970 RETURN
1980 FOR I=1 TO M1
1990 IF I=1 WRITE#4,Z1,B$(I): GOTO2010
2000 WRITE #4,B$(I)
2010 NEXT I
2020 IF M2=0 GOTO2060
2030 FOR I=1 TO M2
2040 WRITE #4,A1(I)
2050 NEXT I
2060 RETURN
2070 REM PUTS PARAMETERS OF PROGRAM INTO MEMORY
2080 READ #1,M1,M2
2090 FOR I=1 TO M1+M2
2100 READ #1,N$(I)
2110 NEXT I
2120 RETURN
2130 FOR I=1 TO M1+M2
2140 PRINT I;" ";N$(I)
2150 NEXT I
2160 RETURN

0010 PRINT "CHANGES A FIELD NAME"
0020 INPUT "NAME OF FILE",F$
0030 OPEN #1,F$
0040 LET G$="1,SCRATCH.DAT"
0050 OPEN #2,G$
0060 GOSUB 200
0070 GOSUB 460
0080 INPUT "WHICH DO YOU WANT TO CHANGE(1,2,3,ETC)",P
0090 PRINT N$(P)
0100 INPUT "ENTER NEW NAME",P$
0110 LET N$(P)=P$
0120 INPUT "CHANGE ANOTHER FIELD NAME(Y OR N)",P$
0130 IF P$="Y" GOTO 80
0140 GOSUB 260
0150 GOSUB 320
0160 CLOSE #1,#2
0170 KILL F$
0180 RENAME G$,F$
0190 CHAIN DATABASE
0200 REM PUTS PARAMETERS OF PROGRAM INTO MEMORY
0210 READ #1,M1,M2
0220 FOR I=1 TO M1+M2
0230 READ #1,N$(I)
0240 NEXT I
0250 RETURN
0260 REM PUTS FIELDS NAMES ON DISK
0270 WRITE #2,M1,M2
0280 FOR I=1 TO M1+M2
0290 WRITE #2,N$(I)
0300 NEXT I
0310 RETURN
0320 FOR I=1 TO M1
0330 IF I=1 THEN READ #1,Z,A$(I):GOTO350
0340 READ #1,A$(I)
0350 IF EOF(1)=1 THEN 450
0360 IF I=1 WRITE#2,Z,A$(I):GOTO 380
0370 WRITE #2,A$(I)
0380 NEXT I
0390 IF M2=0 THEN 440
0400 FOR I=1 TO M2
0410 READ #1,A(I)
0420 WRITE #2,A(I)
0430 NEXT I
0440 GOTO 320
0450  Z=0:RETURN
0460 FOR I=1 TO M1+M2
0470 PRINT I;" ";N$(I)
0480 NEXT I
0490 RETURN

0010 REM APPEND
0020 INPUT "NAME OF FILE TO BE CREATED",F$
0030 OPEN #1,F$
0040 LET G$="1,SCRATCH.DAT"
0050 OPEN #2,G$
0060 INPUT "NAME OF FILE TO APPEND",H$
0070 OPEN #3,H$
0080 GOSUB 250
0090 GOSUB 310
0100 LET Z=0
0110 READ #3,P$,L$,S$,C$,T$
0120 IF EOF(3)=1 THEN 210
0130 FOR I=1 TO M1+M2
0140 IF I=1 WRITE#2,Z,P$
0150 IF I=2 WRITE#2,L$
0160 IF I=3 WRITE#2,S$
0170 IF I=4 WRITE#2,C$
0180 IF I=5 WRITE#2,T$
0190 NEXT I
0200 GOTO 110
0210 CLOSE #1,#2,#3
0220 KILL F$
0230 RENAME G$,F$
0240 CHAIN DATABASE
0250 REM PUTS PARAMETERS OF PROGRAM INTO MEMORY
0260 READ #1,M1,M2
0270 FOR I=1 TO M1+M2
0280 READ #1,N$(I)
0290 NEXT I
0300 RETURN
0310 REM PUTS FIELDS NAMES ON DISK
0320 WRITE #2,M1,M2
0330 FOR I=1 TO M1+M2
0340 WRITE #2,N$(I)
0350 NEXT I
0360 RETURN

0010 REM PMISCKC
0020 REM PRINT ALL NUMBER OF CHECKS MISSING FROM FILE
0030 INPUT "ARE CHECK IN THE FILE IN NUMERIC ORDER(Y OR N)",N$
0040 IF N$="N" PRINT "PLEASE PLACE IN NUMERIC ORDER":CHAIN SORTC
0050 INPUT "NAME OF CHECK FILE",F$
0060 INPUT "PRINT PORT",P
0070 OPEN #1,F$
0080 GOSUB 250
0090 GOSUB 310
0100 IF EOF(1)=1 THEN230
0110 LET B=C
0120 RESTORE #1
0130 GOSUB 250
0140 GOSUB 310
0150 IF EOF(1)=1 THEN 230
0160 IF B=C THEN B=B+1:GOTO140
0170 PORT= P
0180 PRINT B
0190 PORT= 1
0200 LET B=B+1
0210 IF EOF(1)=1 THEN 230
0220 GOTO 160
0230 CLOSE #1
0240 CHAIN DATABASE
0250 REM PUTS PARAMETERS OF PROGRAM INTO MEMORY
0260 READ #1,M1,M2
0270 FOR I=1 TO M1+M2
0280 READ #1,N$(I)
0290 NEXT I
0300 RETURN
0310 FOR I=1 TO M1
0320 IF I=1 READ#1,Z,C:GOTO340
0330 READ #1,A$(I)
0340 IF EOF(1)=1 GOTO 400
```

```
0350 NEXT I
0360 IF M2=0GOTO 400
0370 FOR I=1 TO M2
0380 READ #1,A(I)
0390 NEXT I
0400 RETURN

0010 REM REPAIRC
0020 PRINT "CHANGES AND CORRECTS AN ITEM IN A FILE"
0030 INPUT "NAME OF FILE",F$
0040 LET G$="1.SCRATCH.DAT"
0050 OPEN #1,F$,#2,G$
0060 GOSUB 470
0070 GOSUB 530
0080 GOSUB 690
0090 INPUT "NUMBER OF FIELD TO BE SEARCHED(1,2,ETC)",B
0100 PRINT "IN ";N$(B);" WHICH ITEM";
0110 INPUT T$
0120 GOSUB 590
0130 IF EOF(1)=1 GOTO410
0140 IF B>M1 THEN C=B-M1:GOTO 180
0150 IF T$=A$(B) THEN 200
0160 GOSUB 730
0170 GOTO 120
0180 IF VAL(T$)=A(C) THEN 200
0190 GOTO 160
0200 FOR I=1 TO M1
0210 PRINT N$(I);" ";A$(I)
0220 NEXT I
0230 IF M2=0 THEN 270
0240 FOR I=1 TO M2
0250 PRINT N$(I+M1);" ";A(I)
0260 NEXT I
0270 INPUT "IS THIS THE CORRECT FILE(Y OR N)",Y$
0280 IF Y$<>"Y" THEN 160
0290 PRINT "NOW INPUT TO CORRECT THE FILE"
0300 FOR I=1 TO M1
0310 PRINT N$(I);
0320 INPUT A$(I)
0330 NEXT I
0340 IF M2=0 GOTO390
0350 FOR I=1 TO M2
0360 PRINT N$(M1+I);
0370 INPUT A(I)
0380 NEXT I
0390 GOTO 160
0400 PRINT "THIS IS THE END OF THE FILE"
0410 INPUT "DO YOU WANT TO CHANGE ANOTHER ITEM(Y OR N)",Y$
0420 CLOSE #1,#2
0430 KILL F$
0440 RENAME G$,F$
0450 IF Y$="Y" GOTO40
0460 CHAIN DATABASE
0470 REM PUTS PARAMETERS OF PROGRAM INTO MEMORY
0480 READ #1,M1,M2
0490 FOR I=1 TO M1+M2
0500 READ #1,N$(I)
0510 NEXT I
0520 RETURN
0530 REM PUTS FIELDS NAMES ON DISK
0540 WRITE #2,M1,M2
0550 FOR I=1 TO M1+M2
0560 WRITE #2,N$(I)
0570 NEXT I
0580 RETURN
0590 FOR I=1 TO M1
0600 IF I=1 READ#1,Z,A$(I):GOTO620
0610 READ #1,A$(I)
0620 IF EOF(1)=1 THEN 680
0630 NEXT I
0640 IF M2=0 THEN 680
0650 FOR I=1 TO M2
0660 READ #1,A(I)
0670 NEXT I
0680 RETURN
0690 FOR I=1 TO M1+M2
0700 PRINT I;" ";N$(I)
0710 NEXT I
0720 RETURN
0730 FOR I=1 TO M1
0740 IF I=1 WRITE#2,Z,A$(I):GOTO760
0750 WRITE #2,A$(I)
0760 NEXT I
0770 IF M2=0 THEN GOTO810
0780 FOR I=1 TO M2
0790 WRITE #2,A(I)
0800 NEXT I
0810 RETURN

0455 DATA SEARCH,16
4200 REM SEARCHES FOR A SPECIFIC ITEM
4205 OPEN #1,F$
4210 INPUT "PRINT PORT",P
4215 GOSUB 7000
4220 GOSUB 7800
4225 INPUT "NUMBER OF FIELD TO SEARCH(1,2,3,ETC)",I
4245 LET B=I
4250 PRINT "IN ";N$(B);" WHICH ITEM"
4255 INPUT T$
4260 FOR I=1 TO M1
4265 IF I=1 READ#1,Z,A$(I):GOTO4275
4270 READ #1,A$(I)
4275 IF EOF(1)=1 THEN 4375
4280 NEXT I
4285 IF M2=0 THEN 4305
4290 FOR I=1 TO M2
4295 READ #1,A(I)
4300 NEXT I
4305 IF B>M1 THEN C=B-M1:GOTO4320
4310 IF T$=A$(B) THEN 4330
4315 GOTO 4260
4320 IF VAL(T$)=A(I) THEN 4330
4325 GOTO 4260
4330 PORT= P
4335 FOR I=1 TO M1
4340 PRINT A$(I)
4345 NEXT I
4350 IF M2=0 THEN 4370
4355 FOR I=1 TO M2
4360 PRINT A(I)
4365 NEXT I
4370 GOTO 4260
4375 PORT= 1
4380 CLOSE #1
4385 RETURN
```

6800 Debug Package

The TSC 6800 Debug Package provides a better way to trap program bugs. It is an extremely powerful and complete assembler language program debugging tool which is capable of simulating all functions of the 6800 microprocessor, including interrupts and I/O operations. It is an ideal substitute for hardware logic analyzers or CPU emulators at only a fraction of the cost.

Any number of breakpoints may be user defined. Each breakpoint may invoke any one or combination of eight different actions. These actions may be dependent on a user defined condition such as register A=$FF or memory location $1B55=0. The actions may also be delayed or limited by a pass count. Histogram breakpoints may be set to enable profiling of the executed program. Breakpoints may be set in RAM or ROM!

Complete simulation control allows trace mode to be enabled at anytime. During trace, registers and opcode mnemonics are displayed after each instruction is executed. Single or multiple instruction stepping is permitted as well as simulation speed control. The trace back feature allows the past 256 executed instructions to be viewed. Program execution may be halted at anytime by operator command.

Memory protection and traps are another key feature. Any section(s) of memory may be write, execute, memory, or simulate protected. Execution traps allow program exit on general conditions such as interrupt instruction, transfer instruction, subroutine nest count, and instruction count timeout.

General features include a line at a time assembler, disassembler, memory interrogation commands, hex calculator, machine states counter, stack protection, register modifier, and mode control. In all, there are over 50 commands available. The manual includes detailed operating instructions as well as the complete commented source listing. Requires 9K at $3C00.

SL68-30	**Manual and source listing**	**$35.00**
SL68-30C	**with KCS Cassette**	**$41.95**
SL68-30D	**with mini FLEX™ diskette**	**$43.00**
SL68-30F	**with 8" FLEX™ diskette**	**$55.00**

Send 25¢ for a complete catalog of TSC's assembler language software for the 6800, 8080, and 6502.

Technical Systems Consultants, Inc.

Box 2574 W. Lafayette, IN 47906

Specialists in Software & Hardware for Industry & the Hobbyist

WE'VE JUST BUILT YOU A BETTER BUG TRAP

XMON
6800 Extended Monitor

William T. Gilliland
917 West 51st Street, South
Wichita, Kansas 67217

Through the unusual talents of Bill Gilliland the XMON Monitor can be added to your repetoire of program management. We have used XMON for several months and find it excellent. Space limitations prohibits us from including the source code in this issue.

ICCD recognizes the time and effort it takes to input programs. For a $10.00 fee, including mailing, ICCD will take any program(s) in the Journal *and provide you with a disk and pay shipping charges in the U.S. and Canada. If there is more room on the disk we will attempt to pack it with other programs. We cannot place Flex+++™ or Smoke Signal Operating Systems on the disk, but can format the disk for either. Currently available are 8", 5" SWTPC, and 5" Smoke. Smoke 8" will be available at a later date. Eight inch disks will be slightly higher (as expected) $13.00*

If an article is too long but the source code is cogent to the member, we can offer copies at 8 cents per page plus shipping. We hope this will help those persons in need of copy or disks.

Editor

This monitor resides at $C000 and should be loaded with DOS by a startup file. It will add the label XMON to the FLEX™ command table by establishing continuation of the command table at label CDTBL. XMON may be entered from FLEX by typing XMON and may be entered from SWTBUG™ by typing Z.

Once the monitor is running, the prompt RDY< will be printed. At this time the monitor will respond to commands entered at the keyboard. Typing H will display the available commands. A Control C at any time will exit to FLEX. Many of the monitor routines loop endlessly and may be exited by typing the FLEX line delete character, normally a control X. Others are exited automatically at the end of the task.

The **GET** address routines retain the previous addresses used. These are displayed each time an address is asked for. If no change is desired simply type a carriage return. This is very convenient for multiple operations within the same block of memory.

The print routine supports a serial printer on Port 3. If you have a different type of printer, you must replace the routine at label PINIT with your Port Initialization routine, and the routine at label PRTROU with your printer output routine. All FLEX TTYSET parameters are honored during printer output except the pause feature at the end of each page is disabled. ESC still controls pause, and the

pause is restored upon completion of the monitor routine.

The following commands are available:

B SWTBUG

Typing a B exits XMON and jumps to SWTBUG. XMON may be reentered from SWTBUG by typing Z. The TTYSET duplex mode is restored upon return to XMON.

C Memory Examine-Change

Basically two routines, this routine allows extreme flexibility in program modification. The * prompt signfies that the routine is ready for input. A location is opened by typing the address followed by a / (slash) or a . (period). Leading zeroes need not be typed.

When the location is opened with a slash, the contents are displayed in hex format. The contents may be changed by typing the new value and the location is then closed by typing a terminator. Any of several terminators may be used depending upon the desired action. A carriage return will close the location and restart the routine with a prompt. A line feed will close the location and open the next sequential location. A @ (at sign) will open the location addressed by the current and previous bytes. After a location is closed with a carriage return, it may be reopened by typing a slash. When the contents of the location are changed, the routine checks to see that the change actually took place, and if not, an error message is typed.

When the location is opened with a period, the instruction at the location is diassembled and displayed mnemonically. The same rules for closing the location with a terminator apply here. However, when the line feed is typed, the next sequential instruction is displayed, not the next location, and when the @ is typed, the instruction at the address determined by the operand is displayed. The instruction may be changed by typing in the new instruction in normal assembly language format, remembering that all numbers must be entered in hex and must be preceded with a $.

Branches are entered with a four digit hex address instead of an offset. ASCII values may be entered in the ' format. Single digit indexes need not be preceded with a $, and a single X instead of O, X will suffice. When the branch address is not known at the time of entering the instruction, an * may be typed after mnemonic, and space will be reserved for the offset to be entered later. The type of branch may be changed by typing the new nmemonic without typing the address.

D Disassembler

This routine disasssembles the program between two addresses. Printer output is supported in this routine.

E Examine Memory Block

This routine displays the contents of memory between two addresses. the contents are displayed both in hex and ASCII. Printer output is supported in this routine.

F Find Hex String

This routine inserts hex characters while moving memory block defined by BEG ADR and END ADR to right.

J Jump to Target Program

this routine jumps to program beginning at BEG ADR.

L Hex Loader

this routine loads characters into consecutive memory locations starting a BEG ADR. The preceding address is opened when a - is typed. The current address is typed when a space is typed.

M Move Memory Block

This routine moves a block of memory defined by BEG ADR and END ADR to a new starting locaton defined by the TO address.

N Number Base Conversion

This routine converts decimal to hex and hex to decimal. A pound sign is displayed when the routine is ready for input. Decimal numbers are typed in with a - preceding them if they are negative. Hex numbers are preceded by a $. Numbers must be smaller then 65536 decimal. After the numer is entered the conversion is initiated by typing =.

OS Offset Calculator (Store)

ON Offset Calculator (Non-store)

These routines calculate relative offsets. Out of range branches are indicated by an error message. the ON routine simply displays the offset. The OS routine calculates the offset and stores it in the byte following the BEG ADR.

P Print ASCII Text

This routine prints ASCII text beginning at BEG ADR and terminated with Hex 4.

S Shift Memory Block Left

This routine shifts a block of meory defined by BEG ADR and END ADR left by the number of places indicated by OFST. OFST must be entered as one hex byte.

T Enter ASCII Text

This routine enters ASCII text beginning at BEG ADR. The routine is exited by typing Control D. this places a hex 4 at the end of the string, displays the last address used, and exits the routine.

"OLD RELIABLE"

BASIC FLOPPY DISK SYSTEM

- RANDOM ACCESS FILES
- ANY NUMBER OF FILES MAY BE OPEN (IN USE) AT ONE TIME
- THE NUMBER OF FILES AND SIZE OF FILES IS LIMITED ONLY BY THE SIZE OF THE DISK
- MERGING FILES REQUIRES NO EXTRA DISK SPACE
- NO WAITING FOR THE DISK TO RE-PACK
- LONGER DISK LIFE—MORE EVEN DISK WEAR

We delivered our first mini-floppy disk system a year and a half ago — 6 months ahead of any other 6800 based mini system. Since that time, it has earned the reputation of being the most reliable mini-disk system available.

This system comes completely assembled with a disk controller that is plug compatible with the SWTPC 6800. In fact all our products use the 6800 standard SS-50 (Smoke Signal 50) bus used by SWTPC. The cabinet and power supply are capable of handling up to 3 Shugart Mini-Floppy Drives. One drive is included in the price of the BFD-68 and others may be added easily at any time. Or you may save money by ordering the dual-drive BFD-68-2 or triple drive BFD-68-3 (pictured). Price: BFD-68 $795, BFD-68-2 $1139, BFD-68-3 $1479, SA-400 Drive $355.

A bootstrap PROM is included on the controller board to initiate the Disk Operating System. Thus, you can be up and running from a cold start in just a few seconds.

SUPER SOFTWARE

The BFD-68 includes our new expanded disk operating system and disk file handling BASIC interpreter. In addition, the BFD-68 is supported by the most complete microcomputer software available today. This includes an excellent editor and text processor, several assemblers and a BASIC compiler.

Send for FREE NEW Computer Products Catalog

SMOKE SIGNAL BROADCASTING

31336 Via Colinas, Westlake Village CA 91361
(213) 889-9340

Source for Port 2 Print Routine Using an ACIA

James Petty, M.D.
1016 N.W. 41st Street
Oklahoma City, Oklahoma 73118

John Waldvogel
Motorola Inc.
Suite 301
800 N.E. 63rd Street
Oklahoma City, Oklahoma 73112

We are quite sure that you have been frustrated when the routine or program you are working with requires hard copy. TSC in Flex+++™ allows the P.CMD to send your output to PORT 7 via a PIA (6820). What about using an RS 232-C configured printer? The source code below allows you to use an ACIA (6850) via PORT 2 to output just as conveniently as by using Q for PORT 2. With simple modification the same source code can be rewritten for other ports and other printers. 6800 ICCD uses PORT 2, 3 and 7 for Printers. i.e. PORT 2 with Q drives a Model 35 Teletype.

Editor

Q.CMD

```
VERSION EQU     1
PORT    EQU     $8008
        ORG     $0010
        FDB     INIT
        FDB     OUTCHR
        ORG     $710D
        FDB     OUTCHR
        ORG     $A016
OUTCHR  BRA     OUT
        FCB     VERSION
OUT     STX     SAVEX
        PSH B
        LDX     #PORT
        LDA B   #$11
        STA B   0,X
TEST    LDA B   0,X
        ASR B
        ASR B
        BCC     TEST
        STA A   1,X
        LDX     SAVEX
        PUL B
        RTS
SAVEX   RMB     2
        ORG     $A04A
INIT    STX     SAVEX
        LDX     #PORT
        PSH B
        LDA B   #3
        STA B   0,X
        LDX     SAVEX
        PUL B
        RTS
        END
```

QUINT.SYS

```
INDEX   EQU     $0010
FCB     EQU     $7740
LOAD    EQU     $712A
FMS     EQU     $7806
FMSCLS  EQU     $7803
RENTER  EQU     $7106
NFER    EQU     $4
PAUSE   EQU     $7089
PSTRING EQU     $7118
RPTERR  EQU     $713C
WARMS   EQU     $7103
LSTTRM  EQU     $7091
EOL     EQU     $7082
        ORG     $7600
P       BRA     P1
VN      FCB     1
P1      LDA A   LSTTRM
        CMP A   #$D
        BEQ     P8
        CMP A   EOL
        BEQ     P8
        CLR     PAUSE
        LDX     #FCB
        LDA A   #1
        STA A   0,0
        JSR     FMS
        BNE     P2
        LDA A   #$FF
        STA A   59,X
        JSR     LOAD
        LDX     INDEX
        JSR     0,X
        JMP     RENTER
P2      LDA A   1,X
        CMP A   #NFER
        BNE     P3
        LDX     #NOPST
P25     JSR     PSTRING
        BRA     P4
P3      JSR     RPTERR
P4      JSR     FMSCLS
        JMP     WARMS
P8      LDX     #ERSTR
        BRA     P25
NOPST   FCC     '"QUINT.SYS" NOT FOUND'
        FCB     4
ERSTR   FCC     'COMMAND MUST FOLLOW "P"'
        FCB     4
        ORG     $7744
        FCC     'QUINT'
        FCB     0,0,0
        FCC     'SYS'
        END
```

File Edit Ver. 1.00

Kenneth A. Erickson
5816-56 S.W. Archer Rd.
Gainesville, Florida 32608

Just as the BASIC can be edited via the editor, Ken Erickson, in his Gator-Genius, found a way to edit programs in machine language. This is a very valuable program to add to your system. We tried the program and found it excellent.

ICCD recognizes the time and effort it takes to input programs. For a $10.00 fee, including mailing, ICCD will take any program(s) in the Journal *and provide you with a disk and pay shipping charges in the U.S. and Canada. If there is more room on the disk we will attempt to pack it with other programs. We cannot place Flex+++™ or Smoke Signal Operating Systems on the disk, but can format the disk for either. Currently available are 8", 5" SWTPC, and 5" Smoke. Smoke 8" will be available at a later date. Eight inch disks will be slightly higher (as expected) $13.00*

If an article is too long but the source code is cogent to the member, we can offer copies at 8 cents per page plus shipping. We hope this will help those persons in need of copy or disks.

Editor

WARNING
As a precaution, copy file before editing it.

```
                *SYSTEM EQUATES

7806            FMS     EQU     $7806     FILE MANAGEMENT SYSTEM ENTRY
7803            FMSCLS  EQU     $7803     CLOSE FILE CONTROL BLOCK
7103            WARMST  EQU     $7103     WARM START ENTRY
7124            RSTRIO  EQU     $7124     RESTORE I/O VECTORS
710F            GETCHR  EQU     $710F     GET INPUT CHARACTER
7112            PUTCHR  EQU     $7112     OUTPUT CHARACTER
711E            PCRLF   EQU     $711E     PRINT C/R-L/F
7127            GETFIL  EQU     $7127     GET FILE SPEC
712D            SETEXT  EQU     $712D     SET FILE EXTENSION
7136            RPTERR  EQU     $7136     REPORT DISK ERROR
7130            ADDBX   EQU     $7130     ADD ACC B TO INDEX REGISTER
713F            GETHEX  EQU     $713F     GET HEX NUMBER FROM BUFFER
7133            OUTDEC  EQU     $7133     OUTPUT BCD DIGIT
7139            OUTHEX  EQU     $7139     OUTPUT HEX DIGIT
7094            BUFPTR  EQU     $7094     BUFFER POINTER
711B            CLASS   EQU     $711B     CLASSIFY CHARACTER
7121            NXTCHR  EQU     $7121     GET NEXT CHARACTER
7082            EOL     EQU     $7082     END OF LINE CHARACTER
7091            LSTTRM  EQU     $7091     LAST NONALPHANUMERIC CHARACTER

                *PROGRAM FLAGS

0030                    ORG     $30

0030            LINE    RMB     1         LINE COUNTER
0031            PAGE    RMB     1         PAGE OUTPUT COUNTER
0032            CFLAG   RMB     1         CHANGE FLAG
0033            LFLAG   RMB     1         LIST FLAG
0034            LTRACK  RMB     1         LAST TRACK REGISTER
0035            LSECTR  RMB     1         LAST SECTOR REGISTER
0036            COUNT   RMB     2         RANGE REGISTER
0038            LBOUND  RMB     2         LOWER BOUND REGISTER
003A            UBOUND  RMB     2         UPPER BOUND REGISTER
003C            XTEMP   RMB     2         TEMPORAY INDEX REGISTER

                *FILE CONTROL BLOCK STORAGE

0100                    ORG     $0100

0100            FCB     RMB     192
```

```
1000                    ORG     $1000

1000 20 01      FEDIT   BRA     FEDIT1

1002 01                 FCB     1           SET FOR VERSION 0

1003 4F         FEDIT1  CLR A               CLEAR FLAGS AND REGISTERS
1004 97 30              STA A   LINE
1006 97 31              STA A   PAGE
1008 97 32              STA A   CFLAG
100A 97 33              STA A   LFLAG
100C 97 34              STA A   LTRACK
100E 97 35              STA A   LSECTR
1010 97 36              STA A   COUNT
1012 97 37              STA A   COUNT+1
1014 97 38              STA A   LBOUND
1016 97 39              STA A   LBOUND+1
1018 86 FF              LDA A   #$FF
101A 97 3A              STA A   UBOUND
101C 97 3B              STA A   UBOUND+1
101E CE 01 00           LDX     #FCB        LOAD FILE CONTROL BLOCK
1021 BD 71 27           JSR     GETFIL      GET FILE SPECS
1024 25 58              BCS     FEDIT7
1026 CE 01 00           LDX     #FCB        LOAD FILE CONTROL BLOCK
1029 86 01              LDA A   #1          SET FOR TXT EXTENSION
102B BD 71 2D           JSR     SETEXT      SET DEFAULT EXTENSION

102E FE 70 94   FEDIT2  LDX     BUFPTR      LOAD BUFFER POINTER
1031 A6 00              LDA A   0,X         LOAD FIRST CHARACTER
1033 BD 71 1B           JSR     CLASS       CLASSIFY CHARACTER
1036 25 1F              BCS     FEDIT4
1038 81 39              CMP A   #'9         < '9' ?
103A 23 0F              BLS     FEDIT3
103C BD 71 21           JSR     NXTCHR      GET NEXT CHARACTER
103F 81 4C              CMP A   #'L         LIST OPTION ?
1041 26 69              BNE     ERROR
1043 7C 00 33           INC     LFLAG       SET LIST OPTION FLAG
1046 BD 71 21           JSR     NXTCHR      GET NEXT CHARACTER
1049 20 E3              BRA     FEDIT2

104B BD 71 3F   FEDIT3  JSR     GETHEX      GET HEX NUMBER
104E 25 5C              BCS     ERROR
1050 8D 2E              BSR     CONVRT      CONVERT BCD TO HEX #
1052 DF 38              STX     LBOUND      STORE LOWER BOUND
1054 B6 70 91           LDA A   LSTTRM      LOAD LAST TERMINATER

1057 81 2D      FEDIT4  CMP A   #'-          SEPERATOR ?
1059 26 09              BNE     FEDIT5
105B BD 71 3F           JSR     GETHEX      GET HEX NUMBER
105E 25 4C              BCS     ERROR
1060 8D 1E              BSR     CONVRT      CONVERT BCD TO HEX #
1062 DF 3A              STX     UBOUND      STORE UPPER BOUND
1064 B6 70 91   FEDIT5  LDA A   LSTTRM      LOAD LAST TERMINATOR
1067 81 70 82           CMP A   EOL         END OF LINE CHARACTER ?
106A 27 04              BEQ     FEDIT6
106C 81 0D              CMP A   #13         C/R ?
106E 26 BE              BNE     FEDIT2

1070 CE 01 00   FEDIT6  LDX     #FCB        LOAD FILE CONTROL BLOCK
1073 86 01              LDA A   #1
1075 A7 00              STA A   0,X
1077 BD 78 06           JSR     FMS         OPEN FILE
107A 27 5C              BEQ     EDIT
107C 20 38              BRA     ERROR2

107E 20 31      FEDIT7  BRA     ERROR1


                *CONVERT BCD NUMBER IN THE INDEX REGISTER TO A HEX #

1080 DF 3C      CONVRT  STX     XTEMP       SAVE X
1082 CE 00 3C           LDX     #XTEMP      LOAD REGISTER
1085 A6 01              LDA A   1,X         LOAD LSB
1087 16                 TAB
1088 84 0F              AND A   #15         MASK LSD
108A A7 01              STA A   1,X
108C 17                 TBA                 RELOAD WHOLE BYTE
108D 44                 LSR A               SHIFT RIGHT FOUR TIMES
108E 44                 LSR A
108F 44                 LSR A
1090 44                 LSR A
1091 8D 13              BSR     CONVR1      X 10
1093 AB 01              ADD A   1,X         ADD THE TWO DIGITS
1095 A7 01              STA A   1,X         STORE HEX #
1097 A6 00              LDA A   0,X         LOAD MSB
1099 84 0F              AND A   #15         MASK LSD
109B 8D 09              BSR     CONVR1      X 10
109D 8D 07              BSR     CONVR1      X 10 = X 100
109F AB 00              ADD A   0,X         ADD THE TWO BYTES
10A1 A7 00              STA A   0,X
10A3 DE 3C              LDX     XTEMP       LOAD HEX WORD
10A5 39                 RTS

10A6 48         CONVR1  ASL A               X 2
10A7 16                 TAB
10A8 48                 ASL A               X 8
10A9 48                 ASL A
10AA 1B                 ABA                 = 10
10AB 39                 RTS


                *REPORT ERROR CONDITIONS
10AC CE 14 B9   ERROR   LDX     #SYNMSG
10AF 20 21              BRA     ERROR5

10B1 CE 14 96   ERROR1  LDX     #ILLMSG
10B4 20 1C              BRA     ERROR5

10B6 A6 01      ERROR2  LDA A   1,X
10B8 81 04              CMP A   #4          NO FILE ERROR ?
10BA 26 0E              BNE     ERROR4
10BC CE 14 AA           LDX     #NOFMSG
10BF 20 11              BRA     ERROR5

10C1 86 04              LDA A   #4
10C3 A7 00              STA A   0,X
10C5 BD 78 06           JSR     FMS         CLOSE FILE
10C8 27 08              BEQ     ERROR6      ERROR ON CLOSE ?

10CA BD 71 36   ERROR4  JSR     RPTERR      REPORT ERROR CONDITION
10CD BD 78 03           JSR     FMSCLS      CLOSE ALL OPEN FILES
10D0 20 03              BRA     ERROR6

10D2 BD 12 BF   ERROR5  JSR     PSTRNG      PRINT MESSAGE

10D5 7E 71 03   ERROR6  JMP     WARMST      RETURN TO DOS


                *GET A SECTOR AND CHECK FOR EDITING

10D8 7D 00 33   EDIT    TST     LFLAG       LIST OPTION ON ?
10DB 26 03              BNE     EDIT1
10DD BD 71 24           JSR     RSTRIO      RESTORE I/O VECTORS

10E0 CE 01 00   EDIT1   LDX     #FCB        LOAD FILE CONTROL BLOCK
10E3 A6 11              LDA A   17,X        LOAD FIRST TRACK #
10E5 E6 12              LDA B   18,X        LOAD FIRST SECTOR #

10E7 7D 00 31   EDIT2   TST     PAGE
10EA 26 0C              BNE     EDIT3
10EC 36                 PSH A
10ED DF 3C              STX     XTEMP
10EF CE 13 9E           LDX     #FFDMSG
10F2 BD 12 BF           JSR     PSTRNG      OUTPUT FORM FEED STRING
10F5 DE 3C              LDX     XTEMP
10F7 32                 PUL A

10F8 A7 1E      EDIT3   STA A   30,X        STORE TRACK #
10FA E7 1F              STA B   31,X        STORE SECTOR #
10FC 86 09              LDA A   #9          LOAD READ FUNCTION

10FE A7 00      EDIT4   STA A   0,X         STORE FUNCTION
1100 BD 78 06           JSR     FMS         READ / WRITE SECTOR
1103 26 BC              BNE     ERROR3
1105 7D 00 32           TST     CFLAG       ANY CHANGES MADE ?
1108 26 30              BNE     EDIT7
110A DE 36              LDX     COUNT
110C 08                 INX                 INCREMENT SECTOR COUNT
110D DF 36              STX     COUNT
110F 96 36              LDA A   COUNT
1111 91 38              CMP A   LBOUND      AT BEGINNING SECTOR ?
1113 25 06              BLO     EDIT5
1115 96 37              LDA A   COUNT+1
1117 91 39              CMP A   LBOUND+1
1119 24 13              BHS     EDIT6

111B CE 01 00   EDIT5   LDX     #FCB        LOAD FILE CONTROL BLOCK
111E A6 40              LDA A   64,X        LOAD NEXT TRACK
1120 27 9F              BEQ     ERROR3      END OF FILE ?
1122 E6 1E              LDA B   30,X        LOAD PRESENT TRACK #
1124 D7 34              STA B   LTRACK      UPDATE LAST TRACK #
1126 E6 1F              LDA B   31,X        LOAD PRESENT SECTOR #
1128 D7 35              STA B   LSECTR      UPDATE LAST SECTOR #
112A E6 41              LDA B   65,X        LOAD NEXT SECTOR #
112C 20 CA              BRA     EDIT3       GO READ NEXT SECTOR

112E 96 36      EDIT6   LDA A   COUNT
1130 91 3A              CMP A   UBOUND
1132 25 06              BLO     EDIT7
1134 96 37              LDA A   COUNT+1
1136 91 3B              CMP A   UBOUND+1
1138 22 87              BHI     ERROR3

113A 7F 00 32   EDIT7   CLR     CFLAG       RESET CHANGE FLAG
113D CE 13 A5           LDX     #HUPMSG
1140 BD 12 BF           JSR     PSTRNG
1143 BD 12 9C           JSR     PRFLNM      PRINT FILE NAME
1146 CE 13 8E           LDX     #SCTMSG
1149 BD 12 BF           JSR     PSTRNG
114C CE 00 36           LDX     #COUNT
114F 5F                 CLR B
1150 BD 71 33           JSR     OUTDEC      PRINT SECTOR COUNT
1153 CE 01 00           LDX     #FCB        LOAD FILE CONTROL BLOCK
1156 EE 11              LDX     17,X
1158 DF 3C              STX     XTEMP
115A CE 13 54           LDX     #STRMSG
115D BD 12 72           JSR     PRTKSC      PRINT STARTING TRACK AND SECTOR
1160 CE 01 00           LDX     #FCB        LOAD FILE CONTROL BLOCK
1163 EE 13              LDX     19,X
1165 DF 3C              STX     XTEMP
1167 CE 13 5F           LDX     #ENDMSG
116A BD 12 72           JSR     PRTKSC      PRINT ENDING TRACK AND SECTOR
116D DE 34              LDX     LTRACK
116F DF 3C              STX     XTEMP
1171 CE 13 76           LDX     #LSTMSG
1174 BD 12 72           JSR     PRTKSC      PRINT LAST TRACK AND SECTOR
1177 CE 01 00           LDX     #FCB        LOAD FILE CONTROL BLOCK
117A EE 1E              LDX     30,X
117C DF 3C              STX     XTEMP
117E CE 13 7D           LDX     #PRTMSG
1181 BD 12 72           JSR     PRTKSC      PRINT PRESENT TRACK AND SECTOR
1184 CE 01 00           LDX     #FCB        LOAD FILE CONTROL BLOCK
1187 EE 40              LDX     64,X
1189 DF 3C              STX     XTEMP
118B CE 13 87           LDX     #NXTMSG
118E BD 12 72           JSR     PRTKSC      PRINT NEXT TRACK AND SECTOR
1191 BD 12 CB           JSR     PRTCNT      PRINT CONTENTS OF PRESENT SECTOR
1194 7D 00 33           TST     LFLAG       LIST OPTION ON ?
1197 26 50              BNE     EDIT10
1199 CE 13 C6           LDX     #CHNMSG
```

```
119C BD 12 0F  EDIT8   JSR    PSTRNG    PRINT MESSAGE
119F BD 71 0F          JSR    GETCHR    GET RESPONSE FROM TERMINAL
11A2 81 59             CMP A  #'Y       YES ?
11A4 26 37             BNE    EDIT9
11A6 CE 13 E0          LDX    #LOCMSG
11A9 BD 12 0F          JSR    PSTRNG    PRINT MESSAGE
11AC 5F                CLR B            SET FOR HEX #
11AD BD 13 1C          JSR    GETNUM    GET NUMBER FROM TERMINAL
11B0 25 EA             BCS    EDIT8     INPUT ERROR ?
11B2 16                TAB
11B3 CE 01 40          LDX    #FCB+64   LOAD FIRST BYTE OF SECTOR
11B6 BD 71 30          JSR    ADDBX     ADD ACC B TO INDEX REGISTER
11B9 86 20             LDA A  #32
11BB BD 71 12          JSR    PUTCHR
11BE 86 28             LDA A  #'(
11C0 BD 71 12          JSR    PUTCHR
11C3 BD 71 39          JSR    OUTHEX    PRINT HEX NUMBER
11C6 86 20             LDA A  #32
11C8 BD 71 12          JSR    PUTCHR
11CB 86 2D             LDA A  #'-
11CD BD 71 12          JSR    PUTCHR
11D0 86 20             LDA A  #32
11D2 BD 71 12          JSR    PUTCHR
11D5 A6 00             LDA A  0,X       LOAD CONTENTS AT X
11D7 2A 12             BPL    EDIT11
11D9 86 3A             LDA A  #':
11DB 20 14             BRA    EDIT12

11DD 7D 00 32  EDIT9   TST    CFLAG     ANY CHANGES MADE ?
11E0 26 33             BNE    EDIT13
11E2 81 0D             CMP A  #13       C/R ?
11E4 26 03             BNE    EDIT10
11E6 7E 12 6A          JMP    EDIT16

11E9 20 5C     EDIT10  BRA    EDIT14

11EB 81 20     EDIT11  CMP A  #32       < SPACE ?
11ED 24 02             BHS    EDIT12
11EF 86 2E             LDA A  #'.

11F1 BD 71 12  EDIT12  JSR    PUTCHR    PRINT ASCII CHARACTER
11F4 86 29             LDA A  #')
11F6 BD 71 12          JSR    PUTCHR
11F9 86 20             LDA A  #32
11FB BD 71 12          JSR    PUTCHR
11FE 86 3F             LDA A  #'?
1200 BD 71 12          JSR    PUTCHR
1203 C6 01             LDA B  #1      SET FOR A POSSIBLE ASCII CHARACTER
1205 BD 13 1C          JSR    GETNUM    GET NUMBER FROM TERMINAL
1208 25 92             BCS    EDIT8
120A A7 00             STA A  0,X
120C 86 01             LDA A  #1
120E 97 32             STA A  CFLAG     SET CHANGE FLAG
1210 CE 13 EF          LDX    #NORMSG
1213 20 87             BRA    EDIT8

1215 CE 13 A5  EDIT13  LDX    #MUPMSG
1218 BD 75             BSR    PSTRNG
121A CE 14 0C          LDX    #RPLMSG
121D BD 70             BSR    PSTRNG
121F BD 12 9C          JSR    PRFLNM    PRINT FILE NAME
1222 CE 01 00          LDX    #FCB      LOAD FILE CONTROL BLOCK
1225 EE 1E             LDX    30,X
1227 DF 3C             STX    XTEMP
1229 8D 4A             BSR    PRTKS1    PRINT PRESENT TRACK AND SECTOR
122B CE 14 19          LDX    #WTHMSG
122E 8D 5F             BSR    PSTRNG    PRINT MESSAGE
1230 BD 12 CB          JSR    PRTCNT    PRINT NEW CONTENTS OF SECTOR
1233 CE 14 24          LDX    #OKYMSG
1236 8D 57             BSR    PSTRNG
1238 BD 71 0F          JSR    GETCHR    GET RESPONSE FROM TERMINAL
123B 81 59             CMP A  #'Y       YES ?
123D 26 08             BNE    EDIT14
123F CE 01 00          LDX    #FCB      LOAD FILE CONTROL BLOCK
1242 86 0A             LDA A  #10       LOAD WRITE FUNCTION
1244 7E 10 FE          JMP    EDIT4

1247 CE 01 00  EDIT14  LDX    #FCB      LOAD FILE CONTROL BLOCK
124A A6 40             LDA A  64,X      LOAD NEXT TRACK #
124C 27 1C             BEQ    EDIT16    END OF FILE ?
124E E6 1E             LDA B  30,X      LOAD PRESENT TRACK #
1250 D7 34             STA B  LTRACK    UPDATE LAST TRACK #
1252 E6 1F             LDA B  31,X      LOAD PRESENT SECTOR #
1254 D7 35             STA B  LSECTR    UPDATE LAST SECTOR #
1256 E6 41             LDA B  65,X      LOAD NEXT SECTOR #
1258 7F 00 32          CLR    CFLAG     RESET CHANGE FLAG
125B 36                PSH A
125C 96 31             LDA A  PAGE
125E 4C                INC A
125F 81 04             CMP A  #4        END OF PAGE ?
1261 26 01             BNE    EDIT15
1263 4F                CLR A

1264 97 31     EDIT15  STA A  PAGE
1266 32                PUL A
1267 7E 10 E7          JMP    EDIT2
126A CE 13 9E  EDIT16  LDX    #FFDMSG
126D BD 20             BSR    PSTRNG
126F 7E 10 C1          JMP    ERROR3
```

*PRINT THE TRACK AND SECTOR MESSAGE AND NUMBER

```
1272 BD 12 8F  PRTKSC  JSR    PSTRNG    PRINT MESSAGE

1275 CE 13 68  PRTKS1  LDX    #TRKMSG
1278 BD 12 8F          JSR    PSTRNG    PRINT MESSAGE
127B CE 00 3C          LDX    #XTEMP
127E BD 71 39          JSR    OUTHEX    PRINT TRACK NUMBER
1281 CE 13 6F          LDX    #SECMSG
1284 BD 12 0F          JSR    PSTRNG    PRINT MESSAGE
1287 CE 00 3C          LDX    #XTEMP
128A 08                INX
128B BD 71 39          JSR    OUTHEX    PRINT SECTOR NUMBER
128E 39                RTS
```

*PRINT ASCII STRING

```
128F A6 00     PSTRNG  LDA A  0,X
1291 81 04             CMP A  #4        END OF STRING ?
1293 27 06             BEQ    PSTRN1
1295 BD 71 12          JSR    PUTCHR    PRINT CHARACTER
1298 08                INX
1299 20 F4             BRA    PSTRNG

129B 39        PSTRN1  RTS
```

*PRINT DRIVE #, FILENAME, AND EXTENSION

```
129C CE 01 00  PRFLNM  LDX    #FCB      LOAD FILE CONTROL BLOCK
129F A6 03             LDA A  3,X       LOAD DRIVE #
12A1 8A 30             ORA A  #$30      MAKE IT ASCII
12A3 BD 71 12          JSR    PUTCHR    PRINT IT
12A6 86 2E             LDA A  #'.
12A8 BD 71 12          JSR    PUTCHR
12AB C6 09             LDA B  #9

12AD 5A        PRFLN1  DEC B
12AE 27 0A             BEQ    PRFLN2
12B0 08                INX
12B1 A6 03             LDA A  3,X       LOAD CHARACTER
12B3 27 F8             BEQ    PRFLN1    NULL ?
12B5 BD 71 12          JSR    PUTCHR    PRINT IT
12B8 20 F3             BRA    PRFLN1
12BA 86 2E     PRFLN2  LDA A  #'.
12BC BD 71 12          JSR    PUTCHR
12BF C6 03             LDA B  #3

12C1 08        PRFLN3  INX
12C2 A6 03             LDA A  3,X       LOAD CHARACTER
12C4 BD 71 12          JSR    PUTCHR    PRINT IT
12C7 5A                DEC B
12C8 26 F7             BNE    PRFLN3
12CA 39                RTS
```

*PRINT THE CONTENTS OF THE PRESENT TRACK AND SECTOR

```
12CB CE 14 4E  PRTCNT  LDX    #HEDMSG
12CE 8D BF             BSR    PSTRNG
12D0 CE 01 40          LDX    #FCB+64   LOAD FROM FIRST BYTE OF SECTI
12D3 DF 3C             STX    XTEMP
12D5 7F 00 30          CLR    LINE      RESET LINE COUNTER

12D8 BD 71 1E  PRTCN1  JSR    PCRLF     PRINT C/R-L/F
12DB CE 00 30          LDX    #LINE
12DE BD 71 39          JSR    OUTHEX    PRINT LINE #
12E1 DE 3C             LDX    XTEMP
12E3 C6 10             LDA B  #16

12E5 86 20     PRTCN2  LDA A  #32
12E7 BD 71 12          JSR    PUTCHR    PRINT SPACE
12EA BD 71 39          JSR    OUTHEX    PRINT HEX #
12ED 08                INX
12EE 5A                DEC B
12EF 26 F4             BNE    PRTCN2
12F1 86 20             LDA A  #32
12F3 BD 71 12          JSR    PUTCHR    PRINT SPACE
12F6 C6 10             LDA B  #16
12F8 DE 3C             LDX    XTEMP

12FA A6 00     PRTCN3  LDA A  0,X       LOAD CHARACTER
12FC 81 7F             CMP A  #$7F      NO ASCII ?
12FE 25 04             BLO    PRTCN4
1300 86 3A             LDA A  #':
1302 20 06             BRA    PRTCN5

1304 81 20     PRTCN4  CMP A  #32       PRINTABLE ASCII ?
1306 24 02             BHS    PRTCN5
1308 86 2E             LDA A  #'.

130A BD 71 12  PRTCN5  JSR    PUTCHR    PRINT IT
130D 08                INX
130E 5A                DEC B
130F 26 E9             BNE    PRTCN3
1311 DF 3C             STX    XTEMP
1313 96 30             LDA A  LINE
1315 8B 10             ADD A  #$10      INCREMENT LINE COUNTER
1317 97 30             STA A  LINE
1319 2A 8D             BPL    PRTCN1
131B 39                RTS
```

*GET TWO CHARACTERS FROM THE TERMINAL

```
131C BD 71 0F  GETNUM  JSR    GETCHR    GET FIRST CHARACTER
131F 81 2E             CMP A  #'.
1321 26 06             BNE    GETNU1
1323 5D                TST B            LEGAL MODE ?
1324 27 29             BEQ    GETNU4
1326 BD 71 0F          JSR    GETCHR    GET NEXT CHARACTER
1329 0C                CLC              CLEAR CARRY FLAG
132A 39                RTS

132B 8D 1E     GETNU1  BSR    GETNU2
132D 25 20             BCS    GETNU4    INPUT ERROR ?
132F 48                ASL A
1330 48                ASL A
1331 48                ASL A
```

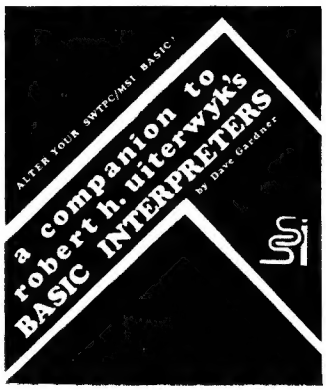
```
1332 48                 ASL A
1333 16                 TAB                     SAVE MSD
1334 BD 71 0F           JSR     GETCHR          GET NEXT CHARACTER
1337 8D 06              BSR     GETNU2
1339 25 14              BCS     GETNU4          INPUT ERROR ?
133B 1B                 ABA                     ADD THE TWO DIGITS
133C 16                 TAB
133D 0C                 CLC                     CLEAR CARRY FLAG
133E 39                 RTS

133F 80 30      GETNU2  SUB A  #$30
1341 2B 0C              BMI     GETNU4
1343 81 09              CMP A  #9               > 9 ?
1345 23 06              BLS     GETNU3
1347 80 07              SUB A  #7
1349 81 0F              CMP A  #15              > $0F ?
134B 22 02              BHI     GETNU4

134D 0C         GETNU3  CLC                     CLEAR CARRY FLAG
134E 39                 RTS

134F CE 13 AE   GETNU4  LDX     #ERRMSG
1352 0D                 SEC                     SET CARRY FLAG
1353 39                 RTS

1354 0D         STRMSG  FCB     13,10
1355 0A
1356 53                 FCC     /STARTING/
1357 54 41
1359 52 54
135B 49 4E
135D 47
135E 04                 FCB     4

135F 20         ENDMSG  FCC     /  ENDING/
1360 20 45
1362 4E 44
1364 49 4E
1366 47
1367 04                 FCB     4

1368 20         TRKMSG  FCC     / TRK $/
1369 54 52
136B 4B 20
136D 24
136E 04                 FCB     4

136F 20         SECMSG  FCC     / SEC $/
1370 53 45
1372 43 20
1374 24
1375 04                 FCB     4

1376 0D         LSTMSG  FCB     13,10
1377 0A
1378 4C                 FCC     /LAST/
1379 41 53
137B 54
137C 04                 FCB     4

137D 20         PRTMSG  FCC     /  PRESENT/
137E 20 50
1380 52 45
1382 53 45
1384 4E 54
1386 04                 FCB     4

1387 20         NXTMSG  FCC     /  NEXT/
1388 20 4E
138A 45 58
138C 54
138D 04                 FCB     4

138E 20         SCTMSG  FCC     /  SECTOR COUNT /
138F 20 53
1391 45 43
1393 54 4F
1395 52 20
1397 43 4F
1399 55 4E
139B 54 20
139D 04                 FCB     4

139E 0D         FFDMSG  FCB     13,12,10,0,0,0,4
139F 0C 0A
13A1 00 00
13A3 00 04

13A5 0D         HUPMSG  FCB     13,10,10,10,$1A,0,0,0,4
13A6 0A 0A
13A8 0A 1A
13AA 00 00
13AC 00 04

13AE 20         ERRMSG  FCC     /  INPUT ERROR - RE-ENTER/
13AF 20 49
13B1 4E 50
13B3 55 54
13B5 20 45
13B7 52 52
13B9 4F 52
13BB 20 2D
13BD 20 52
13BF 45 2D
13C1 45 4E
13C3 54 45
13C5 52

13C6 0D         CHNMSG  FCB     13,10,10
13C7 0A 0A
13C9 41                 FCC     /ANY CHANGES (Y - N) ? /
13CA 4E 59
13CC 20 43
13CE 48 41
13D0 4E 47
13D2 45 53
13D4 20 28
13D6 59 20
13D8 2D 20
13DA 4E 29
13DC 20 3F
13DE 20
13DF 04                 FCB     4

13E0 0D         LOCMSG  FCB     13,10
13E1 0A
13E2 4C                 FCC     /LOCATION ? $/
13E3 4F 43
13E5 41 54
13E7 49 4F
13E9 4E 20
13EB 3F 20
13ED 24
13EE 04                 FCB     4

13EF 20         MORMSG  FCC     / ANY MORE CHANGES (Y - N) ? /
13F0 41 4E
13F2 59 20
13F4 4D 4F
13F6 52 45
13F8 20 43
13FA 48 41
13FC 4E 47
13FE 45 53
1400 20 28
1402 59 20
1404 2D 20
1406 4E 29
1408 20 3F
140A 20
140B 04                 FCB     4

140C 52         RPLMSG  FCC     /REPLACE :/
140D 45 50
140F 4C 41
1411 43 45
1413 20 3A
1415 0D                 FCB     13,10,10,4
1416 0A 0A
1418 04

1419 20         WTHMSG  FCC     /  WITH -/
141A 20 57
141C 49 54
141E 48 20
1420 2D
1421 0D                 FCB     13,10,4
1422 0A 04

1424 0D         OKYMSG  FCB     13,10,10
1425 0A 0A
1427 4F                 FCC     /OKAY TO REPLACE THIS SECTOR (Y - N) ? /
1428 4B 41
142A 59 20
142C 54 4F
142E 20 52
1430 45 50
1432 4C 41
1434 43 45
1436 20 54
1438 48 49
143A 53 20
143C 53 45
143E 43 54
1440 4F 52
1442 20 28
1444 59 20
1446 2D 20
1448 4E 29
144A 20 3F
144C 20
144D 04                 FCB     4
144E 0D         HEDMSG  FCB     13,10
144F 0A
1450 20                 FCC     /   00 01 02 03 04 05 06 07 08 09 0A /
1451 20 20
1453 30 30
1455 20 30
1457 31 20
1459 30 32
145B 20 30
145D 33 20
145F 30 34
1461 20 30
1463 35 20
1465 30 36
1467 20 30
1469 37 20
146B 30 38
146D 20 30
146F 39 20
1471 30 41
1473 20
1474 30                 FCC     /0B 0C 0D 0E 0F 0123456789ABCDEF/
1475 42 20
1477 30 43
1479 20 30
147B 44 20
147D 30 45
147F 20 30
1481 46 20
1483 30 31
1485 32 33
1487 34 35
1489 36 37
148B 38 39
```

```
148D 41 42
148F 43 44
1491 45 46
1493 0D                 FCB     13,10,4
1494 0A 04

1496 0D         ILLMSG  FCB     13,10
1497 0A
1498 49                 FCC     /ILLEGAL FILE NAME/
1499 4C 4C
149B 45 47
149D 41 4C
149F 20 46
14A1 49 4C
14A3 45 20
14A5 4E 41
14A7 4D 45
14A9 04                 FCB     4
14AA 0D         NOFMSG  FCB     13,10
14AB 0A
14AC 4E                 FCC     /NO SUCH FILE/
14AD 4F 20
14AF 53 55
14B1 43 48
14B3 20 46
14B5 49 4C
14B7 45
14B8 04                 FCB     4

14B9 0D         SYNMSG  FCB     13,10
14BA 0A
14BB 53                 FCC     /SYNTAX ERROR/
14BC 59 4E
14BE 54 41
14C0 58 20
14C2 45 52
14C4 52 4F
14C6 52
14C7 04                 FCB     4

                        END     FEDIT

NO ERROR(S) DETECTED
 SYMBOL TABLE:

ADDBX  7130   BUFPTR 7094   CFLAG  0032   CHNMSG 13C6   CLASS  711B
CONVR1 10A6   CONVRT 1080   COUNT  0036   EDIT   10D8   EDIT1  10E0
EDIT10 11E9   EDIT11 11EB   EDIT12 11F1   EDIT13 1215   EDIT14 1247
EDIT15 1264   EDIT16 126A   EDIT2  10E7   EDIT3  10F8   EDIT4  10FE
EDIT5  111B   EDIT6  112E   EDIT7  113A   EDIT8  119C   EDIT9  11DD
ENDMSG 135F   EOL    7002   ERRMSG 13AE   ERROR  10AC   ERROR1 1081
ERROR2 10B6   ERROR3 10C1   ERROR4 10CA   ERROR5 10D2   ERROR6 10D5
FCB    0100   FEDIT  1000   FEDIT1 1003   FEDIT2 102E   FEDIT3 104B
FEDIT4 1057   FEDIT5 1064   FEDIT6 1070   FEDIT7 107E   FFDMSG 139E
FMS    7806   FMSCLS 7803   GETCHR 710F   GETFIL 7127   GETHEX 713F
GETNU1 132B   GETNU2 133F   GETNU3 134D   GETNU4 134F   GETNUM 131C
HEDMSG 144E   HUPMSG 13A5   ILLMSG 1496   LBOUND 0038   LFLAG  0033
LINE   0030   LOCMSG 13E0   LSECTR 0035   LSTMSG 1376   LSTTRM 7091
LTRACK 0034   MORMSG 13EF   NOFMSG 14AA   NXTCHR 7121   NXTMSG 1387
OKYMSG 1424   OUTDEC 7133   OUTHEX 7139   PAGE   0031   PCRLF  711E
PRFLN1 12AD   PRFLN2 12BA   PRFLN3 12C1   PRFLNM 129C   PRTCN1 12D8
PRTCN2 12E5   PRTCN3 12FA   PRTCN4 1304   PRTCN5 130A   PRTCNT 12CB
PRTKS1 1275   PRTKSC 1272   PRTMSG 137D   PSTRN1 129B   PSTRNG 128F
PUTCHR 7112   RPLMSG 140C   RPTERR 7136   RSTRIO 7124   SCTMSG 138E
SECMSG 136F   SETEXT 712D   STRMSG 1354   SYNMSG 1489   TRKMSG 1368
UBOUND 003A   WARMST 7103   WTHMSG 1419   XTEMP  003C
```

The S S I Microcomputer Software Guide

Over 2300 programs on tape, disk, published in books and magazines from 130 software sources (with addresses), classified into 230 categories with cross-references.
Shipped off the shelf.

Second Edition $ 7.95



A Companion to Uiterwyk's BASIC Interpreters by Dave Gardner

70 key memory locations mapped in SWTPC/MSI BASICS plus 30 assembled 6800 routines for ON ERROR GOTO, digit justification, IF THEN ELSE, program length, memory dump and more! With this book you can alter your Uiterwyk BASIC.
Shipped off the shelf.

Second Printing $ 14.95

6800 FLEX™/SWTPC Software

- **Renumbering System by Dave Degler**

  Renumber your BASIC programs with this new FLEX™ utility. You'll wish you had it if you paint yourself into programming "corners". Needs no extra RAM beyond the program being renumbered. With operation notes. Available on FLEX™ minifloppy disk or SWTPC KCS cassette.

- **Some Common Basic Programs by Lon Poole and Mary Borchers**

  Now adapted to FLEX™ and SWTPC 8K BASICS!
  67 key programs from the popular book, which is necessary as the manual. Conversion notes included.

  Disk 1: 37 programs on finance, investments, mortgage amortization, plotting, intergration, more.
  Disk 2: 30 programs on matrix arithmetic, statistics, calendar dates, metrics, more.

  Available on FLEX™ minifloppy disk or SWTPC 8K KCS cassettes.
  The book, Some Common Basic Programs – $ 8.50

- **Weekly Payroll / Income Expense Ledger / Club's Mailing List / Church Membership and Pledge Records by Roger L. Smith**

  These BASIC programs have had years of use and will be valuable additions to your SWTPC software library. Operation notes included. Cassette editions store data on data tapes. Each program is on one FLEX™ minifloppy disk or SWTPC 8K KCS cassette.

  Prices: FLEX™ minifloppy disk $ 16.95 each
  Kansas City Standard SWTPC 8K BASICS Cassette $ 10.95 each

All software shipped off the shelf. Please include check or money order. International: add $ 4.00 per item for air mail postage. U.S. First Class: add $ 2.00.

S S I Publications

4327 East Grove / Phoenix, Arizona 85040

FLEX™ is a trademark of Technical Systems Consultants, Inc.

ORDER BLANK

- ☐ The SSI Microcomputer Software Guide $7.95
- ☐ A Companion to Uiterwyk's Basic Interpreters By Dave Gardner $9.95
- ☐ Mailing List—Tape $7.95
- ☐ Mailing Flex—Tape.....$13.95
- ☐ Income/Expense—Tape.....$7.95
- ☐ Income/Expense—Flex $13.95
- ☐ Inventory—MSI disk $20.00
- ☐ Payroll—Tape.....$7.95
- ☐ Payroll—Flex $13.95
- ☐ Lumber & board footage with bid verification—Tape.....$7.95

S S I
4327 E. Grove St.
Phoenix, AZ 85040

Please send postpaid U.S. above marked items:

Name_____

Address_____

City_____

State_____ Zip_____

Country_____

Please enclose your check or money order. Foreign orders kindly remit in U.S. funds or draft drawn on U.S. bank adding $2.00 per item for postage. Arizone residents add 5% tax. Prices subject to change.
Distributed internationally by Micromedia marketing, Pasadena, California.

HERE'S A QUICK POINT YOU SHOULD KNOW ABOUT

The fastest floating point BASIC for any micro.

TSC BASIC for the 6800 is the fastest floating point BASIC for ANY 8 bit microprocessor. No longer will the 6800 take a back seat to the 6502, 8080, or Z80! And with the TSC name, you know it's top quality.

TSC BASIC is not only fast, but complete with over 50 commands and functions. Features include six digit floating point math, full transcendental functions, unlimited string length, if/then/else construct, logical operators, and two-dimensional arrays including string arrays. The disk versions for FLEX™ 1.0 and 2.0 support random access data files (the mini FLEX™ version does not).

A cassette version requires 10K while the disk versions require at least 12K. No source listings included. With KCS cassette - $39.95; mini FLEX™ - $49.95; FLEX™ 2.0 - $54.95; and FLEX™ 1.0 - $59.95. Soon to come are a business BASIC and 6809 BASIC.

TSC

Technical Systems Consultants, Inc.

All orders should include 3% for postage and handling (8% on foreign orders). Send 25¢ for a complete software catalog.

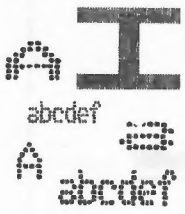
Box 2574
W. Lafayette, IN 47906
(317) 463-2502

Pay a little bit more and get a printer that's brighter than your computer. The BrighterWriter™

When a few dollars more buys you a first-class impact printer, why settle for a toy? The BrighterWriter gives you quality to start with. And versatility that stays even if you outgrow your present personal computer.

Built smart like the big ones.

The BrighterWriter's a smart printer. There's a microcomputer inside. It outwits even the bigger, higher-priced printers. So you get versatility to do all kinds of printing. And power to grow on.



Prints fat, skinny, tall, small.*

This printer can be as creative as your imagination. Stretch out your characters. Squeeze them close. Make them high. Low. Bold. Banner. You name it.

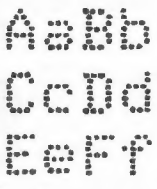
Plugs into your computer.

Most popular personal computers interface to the BrighterWriter. Simply and quickly. Hundreds of BrighterWriters are working in Apple, TRS-80, Heathkit, S-100 and many other personal computer systems right now.

Pictures and fancy symbols.*

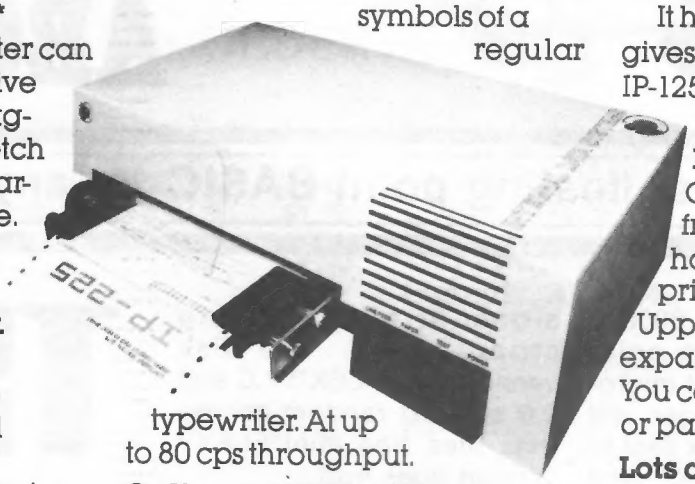
The BrighterWriter draws out your creativity. You can print drawings, graphs, diagrams, bold symbols, or just about any graphic you can imagine.

Picture your page as thousands of dots. The BrighterWriter can fill in the dots, plot them contiguously, stack them, or scatter them. And its special set of graphic characters simplifies the process.



Prints any character a typewriter can. Faster . . .

The BrighterWriter can print plain and simple. With 7x7 dot matrix clarity. You get all the letters, numbers, and standard symbols of a regular typewriter. At up to 80 cps throughput.



Ordinary paper.

Fancy or plain, the BrighterWriter prints on ordinary paper. Better yet, it prints on many shapes of paper. Single sheets. Roll. Fanfold.

Want more copies? The BrighterWriter prints multiple copies without extra adjustments.

Four easy buttons.

Operating the BrighterWriter couldn't be simpler. Up-front controls are easy to get to. A power button to turn it on. A test button to self-test your printer. A paper feed button to advance the sheets or forms. A line feed button to advance the paper a line at a time.

Prints any-which-way.

The BrighterWriter comes in two models. The IP-225, at $949, gives you a BrighterWriter with tractor-feed drive for precision forms control. This one can handle everything from labels to 8½" paper widths.

It has eight form lengths and gives you all the features of our IP-125.

A brighter buy.

Our IP-125, friction-feed BrighterWriter has a 96 character set and prints on 8½" wide paper. Upper and lowercase. It prints expanded characters, too. You can choose a RS-232 serial or parallel interface. $799

Lots of goodies.

There's more. Choose all kinds of options for your BrighterWriter. Up to 132 characters per line, variable character densities, larger buffers, special graphics packages, interface cables, and more.

Give us a call or write. Integral Data Systems, 14 Tech Circle, Natick, MA 01760, (617) 237-7610.

Better yet, see the BrighterWriter at the store nearest you.

Integral Data Systems, Inc.

*Some of these advantages require extra-cost options.