

PROGRAM ABSTRACTS/ALGORITHMS

Data-editing BASIC programs for creating, modifying, and combining Apple II disk files

RICHARD F. WEST

James Madison University, Harrisonburg, Virginia

A wide assortment of statistical analyses can now be conducted on conveniently available microcomputers using programs written in the BASIC language (e.g., Lyczak, 1980). Indeed, *Behavior Research Methods & Instrumentation* has served as a source for numerous statistical programs that can perform tasks that range from *t* tests and multiple-regression analyses to large-design, between- and within-subject ANOVAs (e.g., Galla, 1981, 1982; Gorman & Primavera, 1981; Hacker & Angiolillo-Bent, 1981; Lane, 1981; Steinmetz, Romano, & Patterson, 1981; Wolach, 1981). Random-access data files stored on floppy disks provide an efficient way to store data that will be used in statistical analyses conducted on microcomputers. These files store separate data items in a sequence of individually accessible "records" that are identified by a number that indicates the record's position in a file. Unfortunately, the built-in editing functions of most microcomputers cannot be used to edit disk files, and programs must be written to create and edit these files. Writing such programs can be time-consuming, particularly if the data are to be analyzed in more than one way or if they are stored in more than one file. The menu-driven data-editing package of five programs described here provides a user-friendly means of creating and editing random-access data files for the Apple II microcomputer.

Description. FILE EDITOR presents a menu that runs any of the four other editing programs or a statistical analysis menu. FILE MAKER can be used to create a new random-access data file, to add data to an existing file, to insert or delete a specific record (i.e., data item), to change an existing record, to list a specific record, or to list the data in a file in a subject-by-variable format. FILE VAR COMBINER can be used to form a new data file that contains a user-selected subset of subjects and variables from one or more existing files. The order in which the variables are entered into the new file can be specified. In addition, new variables that are formed by summing, taking the difference between, or averaging two variables can be entered into the new file. FILE MERGER-END can be used to form a new data file that contains the records from one existing

file followed by the records from a second existing file. FILE MERGER-MIDDLE forms a new data file that contains the first-half records from one existing file, then the first-half records from a second existing file, and, finally, the second-half records of the two existing files. Each editing program can return to the FILE EDITOR menu for additional editing-function selection. Data are saved in disk files prior to a switch to a new editing program and can be loaded into the newly selected program as needed. Finally, the editing programs contain routines designed to protect against the accidental overwriting or erasure of data in existing files.

Limitations. A number of the editing functions are carried out by use of arrays that have been dimensioned to contain a maximum of 5,000 elements (i.e., data items). Although a single program could have been written to carry out the editing functions described above, such a program would have been large, thus greatly reducing the amount of memory available for data (e.g., array values require 5 bytes each). One characteristic of Apple II random-access files is that each record in a given file holds the same amount of information (i.e., number of bytes or characters). The number of bytes that can be stored in a record of a given file is referred to as the record "length," which must be specified when the file is created. As written, the data-editing programs create files with a record length of 10. Thus, a data item may not exceed 10 decimal places in length. This maximum length can easily be changed by changing the value of the second variable in each program prior to running that program.

Computer and Language. The programs are written in APPLESOFT BASIC and run on an Apple II Plus microcomputer with two disk drives, 48K RAM, and language card. No modification should be needed to run on a 48K system that has no language card. A single disk-drive system can be used if, prior to running a program, the value of the first variable in the program is changed to "1" (this variable indicates the drive number of the disk containing the data files).

RAM and Time Requirements. Each program has been written in two forms. One form is documented, and the other form is undocumented and contains several statements on each program line. The documented programs take up between 836 and 5,972 bytes in RAM, whereas the more compact versions take up between 687 and 3,445 bytes in RAM. Although the more compact programs run slightly faster than the documented programs, their compact form makes them difficult to edit; their main value lies in the economical use of RAM. As a result, the more compact forms of the

The author's mailing address is: Department of Psychology, James Madison University, Harrisonburg, Virginia 22807.

editing programs make possible either the use of arrays larger than those discussed above (which correspond to the documented forms of the programs) or implementation on systems with less than 48K RAM. The editing programs run reasonably fast. For example, the documented form of FILE MERGER-END was able to combine two files to form a new file with 200 data items in approximately 30 sec. The formation of a much larger file that contained 2,000 data items required nearly 3.5 min.

Availability. A listing of the data-editing programs and instructions on their use (including examples of how they can be used with statistical programs) can be obtained without charge by writing Richard F. West, Department of Psychology, James Madison University, Harrisonburg, Virginia 22807. A diskette containing both the documented and more compact forms of the editing programs, as well as examples of statistical programs that can use data from the created files, is available for \$8.00 (to cover diskette and mailing costs).

REFERENCES

GALLA, J. P. Factorial analysis of variance with unequal observa-

- tions: A BASIC program for microcomputers. *Behavior Research Methods & Instrumentation*, 1981, 13, 699-700.
- GALLA, J. P. Two-factor analysis of variance with one nested factor: A BASIC program for microcomputers. *Behavior Research Methods & Instrumentation*, 1982, 14, 499.
- GORMAN, B. S., & PRIMAVERA, L. H. BASIC BACKSTEP: A simple backward-selection multiple-regression program for minicomputers and microcomputers. *Behavior Research Methods & Instrumentation*, 1981, 13, 703.
- HACKER, M. J., & ANGIOLILLO-BENT, J. S. A BASIC package for N-way ANOVA with repeated measures, trend analysis, and user-defined contrasts. *Behavior Research Methods & Instrumentation*, 1981, 13, 688.
- LANE, D. M. A general analysis of variance program for microcomputers. *Behavior Research Methods & Instrumentation*, 1981, 13, 694.
- LYCZAK, R. A. *Elementary programming for statistics*. North Scituate, Mass: Duxbury Press, 1980.
- STEINMETZ, J. E., ROMANO, A. G., & PATTERSON, M. E. Statistical programs for the Apple II microcomputer. *Behavior Research Methods & Instrumentation*, 1981, 13, 702.
- WOLACH, A. H. Multifactor ANOVA's with and without repeated measures: Programs for limited microcomputers. *Behavior Research Methods & Instrumentation*, 1981, 13, 697.

(Revision accepted for publication February 22, 1983.)

Instructions: Data-Editing BASIC Programs for Creating,
Modifying, and Combining Apple II Disk Files

Richard F. West
Department of Psychology
James Madison University
Harrisonburg, Virginia 22807

DISCLAIMER: No warranty (expressed or implied) is made by the author or by James Madison University as to the accuracy or functioning of the Data-Editing and accompanying programs or their documentation. Although considerable effort has been made to minimize the likelihood of accidental data loss, such loss remains a possibility. While practice with the use of the programs should lessen the chance of data loss, it is recommended that disks not be used as the sole means of important data storage.

INTRODUCTION:

The menu-driven data-editing package of five programs described below provides a user-friendly means of creating and editing random-access data files for the Apple II microcomputer. (See West, BRMI, 1983, for a description of the package and its limitations.) These programs are written in Applesoft BASIC and run on an Apple II (plus) microcomputer with two disk drives and 48K RAM. A single disk drive system can be used when simple modifications are made in the programs (see "FOR SINGLE DRIVE SYSTEMS" below). A number of the editing functions of these programs are carried out by use of arrays that have been dimensioned to contain a maximum of 5000 elements (i.e., data items).

The random-access data files created by these programs store separate data items in a sequence of individually accessible "records" that are identified by a number which indicates the record's position in a file. One characteristic of Apple II random-access files is that each record in a given file holds the same amount of information (i.e., number of bytes or characters). The number of bytes that can be stored in a record of a given file is referred to as the record "length", which must be specified when a file is created or accessed. As written, the data-editing programs default to a record length of ten. Thus, a data item may not exceed ten decimal places in length (it can be shorter). Although it is recommended that you leave this value at ten, this maximum length can easily be changed by changing the second variable in each program prior to running that program. Thus, the value of variable "B" in line 30 of FILE MAKER can be changed (it is now 10) if a different maximum record length is desired. Please note, however, that the use of a program that specifies a record length different from that actually used when the file was created will result in the loss of data from that file.

GETTING STARTED:

Place a back-up disk containing the editing programs in drive number 1. Place a second, initialized disk in drive number 2. The second disk will be used to store data files.

Five programs are involved in the editing of files. Although each program can be run individually, it will normally be convenient to run the FILE EDITOR program first. This program presents a menu that runs any of the four other editing programs or the statistical analysis menu. Each editing program can return to the FILE EDITOR menu for additional editing function selection. A second copy of FILE EDITOR can be used as the HELLO program so that this program will run when the disk is booted.

FILE EDITOR:

This program will present the following menu:

INDICATE DESIRED EDITING FUNCTION:

- 0 = STOP
- 1 = CREATE, ADD TO, CHANGE, & LIST
- 2 = ADD ONE FILE TO END OF OTHER
- 3 = ADD ONE FILE TO MIDDLE OF OTHER
- 4 = SELECT AND COMP VAR FROM OLD FILE
- 5 = STATISTICAL ANALYSIS MENU
- 6 = SPACE REMAINING ON DATA DISK

WHICH?

Enter the number of the desired editing function and hit <RETURN>. (Note that for all programs you will normally need to hit <RETURN> after entering a response from the keyboard.) A response of "1" will run FILE MAKER, "2" will run FILE MERGER-END, "3" will run FILE MERGER-MIDDLE, "4" will run FILE VAR COMBINER, "5" will run STAT MENU, and "6" will run FILE SPACE REMAINING. (Note that you will have to add the latter program, for no listing has been provided for this program.)

FILE MAKER (1 = CREATE, ADD TO, CHANGE, & LIST):

This program can be used to create a new file or to edit a file that already exists on the data disk. This program will ask the following:

ACTION REQUIRED:

- 0 = STOP
- 1 = CREATE NEW FILE
- 2 = READ, ADD TO, CHANGE, OR LIST OLD FILE

WHICH?

Indicate desired action. If you want to create a new file, you will be asked for the "unique" name you would like to use for the new file. No existing file on the data disk can have this name (otherwise data from the older file would be lost.) You may type "CATALOG" if you wish to inspect the names of files already used on the disk. The program will automatically add the characters "L10" to the end of your new file's name to remind you that this file has a record length of 10. (A different value, of course, will be printed if you have changed the record length.) Thus, a file named "TEST" will become "TEST L10" on the disk. Use this new file name when you want to look at the file at a later point in time. If you want to edit an existing ("old") file, you will be asked for the name of the existing

file. (Again, you can type "CATALOG" to see the names of files on the disk.)

The following menu will be presented:

COMMANDS:

- 0 = GENERAL EDITING MENU OR STOP
- 1 = READ A RECORD
- 2 = ADD A RECORD
- 3 = CHANGE A RECORD
- 4 = LIST ALL RECORDS
- 5 = INSERT OR DELETE RECORD

WHICH?

A response of "0" will give you the option of returning to FILE EDITOR. A response of "1" simply will allow you to inspect an individual data item in the file. Respond "2" if you are going to add data to the file. If this is a new file, the first item of data you add will be stored in R1 (record number 1), the second in R2, the third in R3, etc. You will want to enter your data as it will be needed for your statistical programs. The data arrangement for an ANOVA program by Lane (1981) is typical: "The data must be arranged cell by cell, with all the data from the first subject preceding all the data from the second subject, and so on. Both the cells and the within-subjects scores should be arranged hierarchically (e.g., A1B1, A1B2, A1B3, A2B1, A2B2, A2B3)" p. 694. In Lane's example, the file you are creating would store the first subject's scores (A1B1 to A2B3) in R1 to R6, respectively. The second subject's scores (A1B1 to A2B3) would be stored in R7 to R12, respectively. To use a simpler example, assume the following set of data:

<u>Subject No.</u>	<u>Score 1</u>	<u>Score 2</u>
1	3	6
2	8	14
3	4	8
4	6	4
5	9	19
6	12	7

When the program asks "NEW DATA FOR R1 = ?", you will respond "3". Likewise, R2 = 6, R3 = 8, R4 = 14, R5 = 4...R12 = 7. If instead of creating a new file you would like to add data to an already existing file, this program will begin by asking you for the score for the next record (for the example above it would be R13.)

From the menu, a response of "3" will allow you to change the score of any record in the file. A response of "4" will allow you to list all the records of the file. Since the file contains only a sequence of records, you will be asked

for the "NUMBER OF DATA POINTS PER SUBJECT?" In our example, your answer would be "2". (The maximum number of variables a listing may have is limited to 50 without redimensioning a variable in the program.) You will be asked whether you want the listing on the screen or on the printer. "CONTROL-S" will temporarily halt the listing on the screen. The section of the program controlling the printout is written for an EPSON 80 printer. If you have a different printer you may need to edit lines 1440 to 1600 of the program (line 1450 allows the printout of up to 80 characters per line.)

From the menu, a response of "5" will allow you to either INSERT or DELETE a single record of a file. In the example above, if you insert a new score at R5, the scores that had been in R5 TO R12 (e.g., "4" to "7") would be moved into R6 to R13, respectively. Likewise, if you had instead deleted the score at R7, the scores that had been in R8 to R12 would be moved into R7 to R11, respectively.

FILE MERGER-END (2 = ADD ONE FILE TO END OF OTHER):

This program can be used to form a new data file that contains the records of one existing file followed by the records from a second existing file. Thus, if the data from two groups of subjects are contained in two different files, this program can be used to combine the data for an overall analysis.

FILE MERGER-MIDDLE (3 = ADD ONE FILE TO MIDDLE OF OTHER):

This program can be used to form a new data file that contains the first half records from one existing file, then the first half records from a second existing file, and, finally, the second half records of the two existing files. For example, if the first file contained 6 records and the second contained 8, the new file would contain 14 records formed by the data from R1, R2, R3, from the first file, followed by the data from R1, R2, R3, R4 from the second file, followed by the data from R4, R5, R6 from the first file, followed by the data from R5, R6, R7, R8 from the second file.

FILE VAR COMBINER (4 = SELECT AND COMP VAR FROM OLD FILE):

This program can be used to form a new data file that contains a user-selected subset of subjects and variables from one or more existing files. The order in which the variables are entered into the new file can be specified. In addition, new variables that are formed by summing, taking the difference between, or averaging two variables can be entered into the new file. For example, an existing file

that contains 8 scores (variables) from each of 50 subjects could be used to form a new file that contains data from the 10th through 30th subjects arranged as follows: variable 3, variable 1, the sum of variables 2 and 3, the mean of variables 6 and 8. Note that this program, in conjunction with FILE MERGER-END, can be used to select and rearrange data for a variety of different types of statistical analyses.

STAT MENU (5 = STATISTICAL ANALYSIS MENU):

This program will present the following menu of statistical analysis programs:

SELECT ONE:

- 0 = STOP
- 1 = BACKSTEP (REGRESSION ANALYSIS)
- 2 = DEPENDENT T-TEST
- 3 = INDEPENDENT T-TEST
- 4 = GENERAL ANOVA

WHICH?

Program listings are provided for the DEPENDENT and INDEPENDENT T-TESTS so that you may see examples of how data files may be used by statistical programs. Consider the former program. Line 40 specifies the drive number of the data disk (E=2) and the random file record length (F=10). Line 60 lets B\$ = CHR\$(4). This line also instructs you to place the data disk in drive E (e.g., "2"). Line 170 asks you for the name of the data file (C\$). Line 180 allows you to look at the catalog if you have forgotten your file's name. Line 190 OPENS the random-access file named C\$ that is on the disk in drive E with a record length of F. Line 240 READS record number L from the file and assigns its value to variable A. Line 250 does the same for variable B. Line 310 CLOSES the file. Line 330 allows the printer (an EPSON 80) to make a printout up to 80 columns wide. Line 910 returns to the FILE EDITOR. DEPENDENT T uses data arranged subject-by-subject. INDEPENDENT T uses data arranged by group.

Program listings are not provided for BACKSTEP or GENERAL ANOVA. BACKSTEP is an excellent multiple-regression analysis program written in Applesoft BASIC by Gorman and Primavera (BRMI, 1981, 13, 703). GENERAL ANOVA is an Applesoft program adapted from an excellent ANOVA program written for a TRS-80 Model I by Lane (BRMI, 1981, 13, 694). This program allows for large designs (e.g., four or five factors), with both between- and within-subject variables. It also will handle designs with unequal cell sizes.

FOR SINGLE DRIVE SYSTEMS:

A single drive system can be used, if prior to running a program, the value of the first variable in the program is changed to "1" (This variable indicates the drive number of the disk containing the data files). FILE EDITOR, Line 20: change A to 1; FILE MAKER, Line 30: change A to 1; FILE MERGER-END, Line 30: change A to 1; FILE MERGER-MIDDLE, Line 30: change B to 1; FILE VAR COMBINED, Line 30: change C TO 1; STAT MENU, Line 20: change A to 1; DEPENDENT T, Line 40: change E to 1; INDEPENDENT T, Line 40: change E to 1.

Although the program user will find it easiest to create data files on a disk that contains the editing programs, a separate data disk can be placed in the drive prior to inputting the name of a file.

PROGRAMS ON DISKETTE:

A diskette containing both the documented and more compact forms of the editing programs (see West, BRMI, 1983, 15, 387-388), as well as examples of statistical programs that can use data from the created files (programs run by STAT MENU), is available for \$8 (to cover diskette and mailing costs). In addition, both one and two disk drive versions of the programs are included on the diskette.

If you have this diskette, be sure to back it up and use the back-up disk for the actual editing activities. DEPENDENT T uses data that is arranged subject-by-subject. INDEPENDENT T uses data arranged by group. Copies of BACKSTEP and GENERAL ANOVA are included on the diskette for your convenience. Program listings and instructions on their use have been made available without charge by the programs' authors (see Gorman and Primavera, BRMI, 1981, 13, 703, for BACKSTEP, and Lane, BRMI, 1981, 13, 694, for GENERAL ANOVA). Please refer to the appropriate reference if you intend to use either of these programs.

BACKSTEP uses data that are arranged in a subject-by-subject fashion. Gorman and Primavera have stated the following: "Program users should be familiar with the program instructions and limitations and should report any defects or potential errors to the authors. Users are requested to reference the use of the BASIC BACKSTEP program in research reports and articles (References: Gorman, B. S., & Primavera, L. H., BACKSTEP: A simple program for backward-selection multiple regression. Behavioral Research Methods and Instrumentation, 1980, 12, 391-392; Gorman, B. S., & Primavera, L. H. BASIC BACKSTEP: A simple backward-selection multiple regression program for

minicomputers and microcomputers. Behavioral Research Methods & Instrumentation, 1981, 13, 703)."

GENERAL ANOVA uses data arranged as discussed under the above section discussing data arrangement for the FILE MAKER program.

As with the other programs contained on the diskette, the statistical programs have been modified to supply printout on an 80 column EPSON 80 printer.

Data Editing Programs

Richard F. West

James Madison University
Harrisonburg, Virginia

NOTE: IN THE LAST TWO PROGRAMS, THE SYMBOL \ominus HAS BEEN USED TO INDICATE EXPONENTIATION. USE THE Λ KEY ON YOUR APPLE.

/FILE EDITOR/

```
10 REM *** FILE EDITOR *** RICH WEST 12/6/82
20 CALL - 936:A = 2: REM A = DRIVE#. CHANGE A TO 1 IN
SINGLE DRIVE SYSTEM
30 AS = CHR$(4)
40 PRINT : PRINT " *** GENERAL EDITOR MENU ***"
50 PRINT : PRINT "PUT EDITING MASTER DISK IN DRIVE#1 & "
60 PRINT : PRINT "DATA FILE DISK IN DRIVE#";A
70 PRINT : PRINT : PRINT "INDICATE DESIRED EDITING
FUNCTION:"
80 PRINT : PRINT " 0 = STOP"
90 PRINT " 1 = CREATE, ADD TO, CHANGE, & LIST"
100 PRINT " 2 = ADD ONE FILE TO END OF OTHER"
110 PRINT " 3 = ADD ONE FILE TO MIDDLE OF OTHER"
120 PRINT " 4 = SELECT AND COMP VAR FROM OLD FILE"
130 PRINT " 5 = STATISTICAL ANALYSIS MENU"
140 PRINT " 6 = SPACE REMAINING ON DATA DISK"
150 PRINT : INPUT "WHICH? ";B
160 IF B = 0 THEN PRINT AS;"CLOSE": END
170 IF B = 1 THEN PRINT AS;"RUN FILE MAKER"
180 IF B = 2 THEN PRINT AS;"RUN FILE MERGER-END"
190 IF B = 3 THEN PRINT AS;"RUN FILE MERGER-MIDDLE"
200 IF B = 4 THEN PRINT AS;"RUN FILE VAR COMBINER"
210 IF B = 5 THEN PRINT AS;"RUN STAT MENU"
220 IF B = 6 THEN PRINT AS;"RUN FILE SPACE REMAINING"
230 GOTO 70
```

/FILE MAKER/

```
10 REM *** FILE MAKER *** MODIFICATIONS BY RICH WEST ***
12/6/82 *** PORTIONS OF THIS PROGRAM WERE SUGGESTED BY
DEMONSTRATION PROGRAM IN POOLE,MCNIFF,&COOK
20 REM *** CREATES AND MAINTAIN RANDOM ACCESS FILES WITH
LENGTH OF RECORDS SET BY VAR B
30 A = 2:B = 10: REM A = DRIVE#2. USE A =1 IF USING ONLY
ONE DRIVE. B = RECORD LENGTH
40 AS = CHR$(4)
50 CALL - 936
60 PRINT : PRINT "THIS PROGRAM CREATES AND MAINTAINS":
PRINT : PRINT "RANDOM-ACCESS FILES. (DATA POINTS ": PRINT :
PRINT "CAN HAVE A MAX LENGTH OF ";B;" DIGITS)"
70 DIM A(5000): DIM B(50)
80 PRINT : PRINT "ACTION REQUIRED:": PRINT
90 PRINT " 0 = STOP"
100 PRINT " 1 = CREATE NEW FILE"
110 PRINT " 2 = READ,ADD TO,CHANGE,OR LIST OLD FILE"
120 PRINT : PRINT : PRINT "WHICH";
130 INPUT C
140 IF C = 0 THEN 1690
150 IF C = 1 THEN 180
```

```

160 IF C = 2 THEN 310
170 GOTO 80
180 CALL - 936
190 PRINT : PRINT "NAME OF NEW FILE (USE A UNIQUE NAME)?"
200 INPUT D$: HOME
210 A$ = "": REM CTRL-D
220 IF D$ = "CATALOG" THEN PRINT A$;"CATALOG,D";A: GOTO
190
230 ONERR GOTO 260
240 D$ = D$ + " L" + STR$(B): PRINT
A$;"RENAME";D$;" ,";D$;" ,D";A
250 PRINT : PRINT "**** THIS IS NAME OF EXISTING FILE ****":
GOTO 190
260 POKE 216,0: PRINT A$;"OPEN";D$;" ,D";A;" ,L";B: REM
*** RECORDS UP TO B CHARACTERS LONG
270 PRINT A$;"WRITE";D$;" ,R0": REM WRITE RECORD ZERO
280 PRINT 0
290 PRINT A$;"CLOSE";D$: REM CLOSES THE FILE
300 GOTO 340
310 CALL - 936
320 PRINT : PRINT "NAME OF OLD FILE": INPUT D$
330 IF D$ = "CATALOG" THEN PRINT CHR$(4);"CATALOG,D";A;:
GOTO 320
340 A$ = "": REM CTRL-D
350 PRINT A$;"OPEN";D$;" ,D";A;" ,L";B: REM *** RECORDS UP TO
B DIGITS LONG
360 PRINT A$;"READ";D$;" ,R0": REM READ RECORD ZERO
370 INPUT D: REM THE LAST RECORD NUMBER IN USE
380 PRINT A$: REM CANCEL READ COMMAND
390 CALL - 936: REM CLEAR SCREEN
400 PRINT : PRINT : PRINT : PRINT
410 PRINT "COMMANDS:": PRINT
420 PRINT " 0 = GENERAL EDITING MENU OR STOP"
430 PRINT " 1 = READ A RECORD"
440 PRINT " 2 = ADD A RECORD"
450 PRINT " 3 = CHANGE A RECORD"
460 PRINT " 4 = LIST ALL RECORDS"
470 PRINT " 5 = INSERT OR DELETE RECORD"
480 PRINT : PRINT
490 PRINT "WHICH";
500 INPUT C
510 IF C = 0 THEN 1710: REM BRANCH
520 IF C = 1 THEN 610: REM TO
530 IF C = 2 THEN 770: REM THE
540 IF C = 3 THEN 1010: REM SELECTED
550 IF C = 4 THEN 1270: REM ROUTINE
560 IF C = 5 THEN 1790
570 GOTO 390: REM (OR RE-DISPLAY THE MENU)
580 REM
590 REM ***** READ A RECORD *****
600 REM
610 CALL - 936: REM CLEAR SCREEN
620 PRINT : PRINT "READ A RECORD": PRINT
630 PRINT "WHICH RECORD NUMBER (0 TO STOP)";
640 INPUT E

```

```

650 IF E < 1 THEN 390: REM RETURN TO MAIN MENU
660 IF E > D THEN 610: REM RECORD DOES NOT EXIST
670 IF E > 0 AND E < = D AND E = INT (E) THEN GOTO 690
680 GOTO 390
690 PRINT A$;"READ";D$;"R";E: REM PREPARE TO READ RECORD
700 INPUT B$: REM READ THE DATA
710 PRINT A$: REM CANCEL READ COMMAND
720 PRINT "R";E;" = ";B$: PRINT : REM DISPLAY THE DATA
730 GOTO 630: REM ASK FOR ANOTHER NUMBER
740 REM
750 REM *** ADD A RECORD ***
760 REM
770 CALL - 936: REM CLEAR SCREEN
780 F = D
790 PRINT : PRINT "ADD A RECORD": PRINT
800 PRINT "(PRESS <RETURN> TO STOP ADDING)"
810 F = F + 1
820 PRINT " NEW DATA FOR R";F;" = ";
830 INPUT B$: REM GET USER'S RESPONSE
840 G = LEN (B$)
850 IF G > B THEN PRINT : PRINT "**** ";B;" DIGITS MAX
****": PRINT CHR$ (7): GOTO 820
860 IF B$ = "" THEN 890
870 A(F) = VAL (B$)
880 GOTO 800: REM LOOP FOR ANOTHER RECORD
890 IF B$ = "" AND F = D + 1 THEN 390
900 IF D = F - 1 THEN 950
910 D = D + 1: REM INCREMENTS LAST RECORD
920 PRINT A$;"WRITE";D$;"R";D: REM PREPARE TO WRITE
930 PRINT A(D): REM SEND DATA TO FILE
940 GOTO 900
950 PRINT A$;"WRITE";D$;"R0": REM PREPARE TO WRITE
RECORD ZERO
960 PRINT D: REM STORE UPDATED VALUE IN R0
970 PRINT A$: GOTO 390: REM CANCEL WRITE COMMAND
980 REM
990 REM **** CHANGE A RECORD ****
1000 REM
1010 CALL - 936: REM CLEAR SCREEN
1020 PRINT : PRINT "CHANGE A RECORD": PRINT
1030 PRINT "CHANGE WHICH RECORD (0 TO STOP)";
1040 INPUT E
1050 IF E < 1 THEN 390: REM RETURN TO MAIN MENU
1060 IF E > D THEN 1010: REM TRY AGAIN IF RECORD NOT ON
FILE
1070 IF E > 0 AND E < = D AND E = INT (E) THEN GOTO 1090
1080 GOTO 390
1090 PRINT A$;"READ";D$;"R";E: REM PREPARE TO READ
1100 INPUT B$: REM READ THE RECORD
1110 PRINT A$: REM CANCEL READ COMMAND
1120 PRINT "CHANGE R";E;" = ";B$;" TO ?": REM DISPLAY THE
DATA
1130 PRINT "(PRESS <RETURN> NOW TO CANCEL CHANGES)"
1140 PRINT " NEW R";E;" = ";: INPUT C$: REM GET USER'S
RESPONSE

```

```

650 IF E < 1 THEN 390: REM RETURN TO MAIN MENU
660 IF E > D THEN 610: REM RECORD DOES NOT EXIST
670 IF E > 0 AND E < = D AND E = INT (E) THEN GOTO 690
680 GOTO 390
690 PRINT A$;"READ";D$;"R";E: REM PREPARE TO READ RECORD
700 INPUT B$: REM READ THE DATA
710 PRINT A$: REM CANCEL READ COMMAND
720 PRINT "R";E;" = ";B$: PRINT : REM DISPLAY THE DATA
730 GOTO 630: REM ASK FOR ANOTHER NUMBER
740 REM
750 REM *** ADD A RECORD ***
760 REM
770 CALL - 936: REM CLEAR SCREEN
780 F = D
790 PRINT : PRINT "ADD A RECORD": PRINT
800 PRINT "(PRESS <RETURN> TO STOP ADDING)"
810 F = F + 1
820 PRINT " NEW DATA FOR R";F;" = ";
830 INPUT B$: REM GET USER'S RESPONSE
840 G = LEN (B$)
850 IF G > B THEN PRINT : PRINT "**** ";B;" DIGITS MAX
****": PRINT CHR$ (7): GOTO 820
860 IF B$ = "" THEN 890
870 A(F) = VAL (B$)
880 GOTO 800: REM LOOP FOR ANOTHER RECORD
890 IF B$ = "" AND F = D + 1 THEN 390
900 IF D = F - 1 THEN 950
910 D = D + 1: REM INCREMENTS LAST RECORD
920 PRINT A$;"WRITE";D$;"R";D: REM PREPARE TO WRITE
930 PRINT A(D): REM SEND DATA TO FILE
940 GOTO 900
950 PRINT A$;"WRITE";D$;"R0": REM PREPARE TO WRITE
RECORD ZERO
960 PRINT D: REM STORE UPDATED VALUE IN R0
970 PRINT A$: GOTO 390: REM CANCEL WRITE COMMAND
980 REM
990 REM **** CHANGE A RECORD ****
1000 REM
1010 CALL - 936: REM CLEAR SCREEN
1020 PRINT : PRINT "CHANGE A RECORD": PRINT
1030 PRINT "CHANGE WHICH RECORD (0 TO STOP)";
1040 INPUT E
1050 IF E < 1 THEN 390: REM RETURN TO MAIN MENU
1060 IF E > D THEN 1010: REM TRY AGAIN IF RECORD NOT ON
FILE
1070 IF E > 0 AND E < = D AND E = INT (E) THEN GOTO 1090
1080 GOTO 390
1090 PRINT A$;"READ";D$;"R";E: REM PREPARE TO READ
1100 INPUT B$: REM READ THE RECORD
1110 PRINT A$: REM CANCEL READ COMMAND
1120 PRINT "CHANGE R";E;" = ";B$;" TO ?": REM DISPLAY THE
DATA
1130 PRINT "(PRESS <RETURN> NOW TO CANCEL CHANGES)"
1140 PRINT " NEW R";E;" = ";: INPUT C$: REM GET USER'S
RESPONSE

```

```

1150 G = LEN (C$)
1160 IF G > B THEN PRINT : PRINT "*** ";B;" DIGITS MAX
***": PRINT CHR$ (7): GOTO 1020
1170 IF C$ > "" THEN 1200: REM BRANCH IF NEW DATA
1180 PRINT "RECORD ";E;" UNCHANGED ! ! !": REM LOOP IF
1190 GOTO 1030: REM NO CHANGES DESIRED
1200 PRINT A$;"WRITE";D$;"R";E: REM PREPARE TO WRITE
1210 PRINT C$: REM STORE CHANGED DATA
1220 PRINT A$: REM CANCEL WRITE COMMAND
1230 GOTO 1030: REM LOOP FOR ANOTHER RECORD TO CHANGE
1240 REM
1250 REM **** LIST ALL RECORDS ****
1260 REM
1270 CALL - 936: REM CLEAR SCREEN
1280 PRINT : PRINT "LIST ALL RECORDS": PRINT
1290 E = 0: REM RESET THE COUNTER
1300 E = E + 1: REM INCREMENT THE COUNTER
1310 IF E > D THEN 1350: REM STOP AFTER LAST RECORD
1320 PRINT A$;"READ";D$;"R";E: REM PREPARE TO READ
1330 INPUT A(E): REM READ DATA INTO ARRAY
1340 GOTO 1300: REM LOOP FOR NEXT RECORD
1350 PRINT A$: REM CANCEL READ COMMAND
1360 H = 0
1370 PRINT : INPUT "NUMBER OF DATA POINTS PER SUBJECT?
";I:J = I - 1: IF I > 50 THEN PRINT "****50 DATA POINTS
MAX****": GOTO 1370: REM SETS J = TO I AFTER FIRST COUNTER
LOOP BELOW
1380 PRINT : PRINT "MODE OF LISTING:": PRINT
1390 PRINT " 1 = ON SCREEN"
1400 PRINT " 2 = ON PRINTER"
1410 PRINT : PRINT : INPUT "WHICH";K
1420 IF K = 1 THEN 1460
1430 PRINT : PRINT "TURN ON PRINTER"
1440 PR# 1
1450 PRINT CHR$ (9) + "80N";
1460 PRINT : PRINT "FILE = ";D$: PRINT
1470 E = 0
1480 E = E + 1
1490 IF E > D THEN PRINT : PRINT : GOTO 1610
1500 J = J + 1: REM COUNTS TO THE NUMBER OF DATA POINTS
PER SUBJECT
1510 IF J = I THEN PRINT : PRINT "S";H + 1;">";: POKE
36,6:H = H + 1:J = 0
1520 B(J) = B(J) + A(E): REM SUMS DATA POINTS
1530 IF J = 8 THEN PRINT : PRINT " ";
1540 IF J = 16 THEN PRINT : PRINT " ";
1550 IF J = 24 THEN PRINT : PRINT " ";
1560 IF E < 10 THEN PRINT " ";E;"")";: GOTO 1590
1570 IF E < 100 THEN PRINT " ";E;"")";: GOTO 1590
1580 PRINT E;"");
1590 PRINT A(E);: PRINT " ";
1600 GOTO 1480: REM LOOP FOR NEXT RECORD
1610 FOR L = 1 TO I: PRINT "MEAN OF VAR # ";L;" = ";B(L -
1) / H:B(L - 1) = 0: NEXT
1620 PR# 0

```



```

1630 PRINT : PRINT "* * * * * END-OF-FILE"
1640 PR# 0
1650 PRINT "PRESS RETURN TO CONTINUE";: REM REQUEST
RESPONSE
1660 INPUT B$: REM GET USER'S RESPONSE
1670 GOTO 390: REM RETURN TO MAIN MENU
1680 REM
1690 REM * * * * STOP PROGRAM * * * *
1700 REM
1710 PRINT A$;"CLOSE": REM CLOSE THE FILE
1720 CALL - 936: REM CLEAR SCREEN
1730 PRINT "PROGRAM COMPLETE."
1740 PRINT A$;"CLOSE"
1750 PRINT : INPUT "WANT 'RANDOM FILE EDITOR' MENU?
<Y/N>";E$
1760 IF E$ = "Y" THEN PRINT A$;"RUN FILE EDITOR,D1"
1770 END
1780 REM
1790 REM *** INSERT OR DELETE RECORD ***
1800 REM
1810 CALL - 936
1820 PRINT : PRINT "DO YOU WANT TO INSERT OR DELETE
RECORD?": PRINT
1830 PRINT " 0 = NEITHER"
1840 PRINT " 1 = INSERT"
1850 PRINT " 2 = DELETE"
1860 PRINT : INPUT "WHICH? ";C: PRINT : PRINT "*** WAIT
***"
1870 IF C = 0 OR C = 1 OR C = 2 THEN 1890
1880 GOTO 1810
1890 IF C = 0 THEN 390
1900 E = 0
1910 E = E + 1: IF E > D THEN 1930
1920 PRINT A$;"READ";D$;" ,R";E: INPUT A(E): GOTO 1910
1930 PRINT A$: REM CANCEL READ COMMAND
1940 IF C = 1 THEN M = 1: GOTO 1960
1950 IF C = 2 THEN M = 2: GOTO 1990
1960 IF M = 1 THEN PRINT : PRINT "NUMBER OF RECORD TO BE
INSERTED?": INPUT C
1970 IF M = 1 THEN PRINT : INPUT "VALUE OF THIS NEW
RECORD?";N
1980 F$ = STR$(N):G = LEN(F$): IF G > B THEN PRINT :
PRINT "*** ";B;" DIGITS MAX ***": PRINT CHR$(7): GOTO 1970
1990 IF M = 2 THEN PRINT : INPUT "NUMBER OF RECORD TO BE
DELETED?";O
2000 IF C < 1 OR C > D THEN 1960
2010 PRINT : PRINT "*** WAIT ***":E = 0:P = 0
2020 IF M = 1 THEN D = D + 1
2030 IF M = 2 THEN D = D - 1
2040 FOR E = 1 TO D
2050 IF M = 1 AND E = C THEN Q = N: GOTO 2080
2060 IF M = 2 AND E = O THEN P = P + 1
2070 P = P + 1:Q = A(P)
2080 PRINT A$;"WRITE";D$;" ,R";E: PRINT Q
2090 NEXT

```

```

2100 PRINT A$;"WRITE";D$;" ,R0"
2110 PRINT D
2120 PRINT A$: GOTO 390

```

```

/FILE MERGER-END/

```

```

10 REM      *** FILE MERGER-END 12/3/82 *** RICH WEST
20 REM      *** ADDS RECORDS FROM FILE 2 TO END OF FILE 1 TO
FORM FILE 3
30 A = 2:B = 10: REM A = DRIVE#2. CHANGE TO A =1 IF USING
ONLY ONE DRIVE.  B = RECORD LENGTH
40 DIM A(5000)
50 A$ = CHR$(4)
60 CALL - 936
70 PRINT : INPUT "NAME OF EXISTING FILE#1? ";B$
80 IF B$ = "CATALOG" THEN PRINT A$;"CATALOG,D";A: GOTO 70
90 HOME
100 PRINT : INPUT "NAME OF EXISTING FILE#2? ";C$
110 IF C$ = "CATALOG" THEN PRINT A$;"CATALOG,D";A: GOTO
100
120 HOME
130 PRINT : INPUT "UNIQUE NAME OF NEW FILE? ";D$
140 IF D$ = "CATALOG" THEN PRINT A$;"CATALOG,D";A: GOTO
130
150 ONERR GOTO 190
160 D$ = D$ + " L" + STR$(B): PRINT
A$;"RENAME";D$;" ,";D$;" ,D";A
170 PRINT : PRINT "****THIS IS NAME OF EXISTING FILE****":
GOTO 130
180 REM      *** DATA FROM FILE 1
190 POKE 216,0: PRINT A$;"OPEN";B$;" ,L";B
200 PRINT A$;"READ";B$;" ,R0"
210 INPUT C: REM NUMBER OF RECORDS
220 FOR D = 1 TO C
230 PRINT A$;"READ";B$;" ,R";D
240 INPUT A(D)
250 NEXT
260 PRINT A$;"CLOSE"
270 REM      *** DATA FROM FILE 2
280 PRINT A$;"OPEN";C$;" ,L";B
290 PRINT A$;"READ";C$;" ,R0"
300 INPUT E: REM NUMBER OF RECORDS
310 F = D - 1
320 FOR D = 1 TO E
330 F = F + 1
340 PRINT A$;"READ";C$;" ,R";D
350 INPUT A(F)
360 NEXT
370 D = F
380 PRINT A$;"CLOSE"
390 REM DATA TO FILE 3
400 PRINT A$;"OPEN";D$;" ,L";B
410 PRINT A$;"WRITE";D$;" ,R0"

```

```

420 PRINT D: REM R0=NUMBER OF RECORDS
430 FOR G = 1 TO D
440 PRINT A$;"WRITE";D$;" ,R";G
450 PRINT A(G)
460 NEXT
470 PRINT A$;"CLOSE"
480 PRINT : INPUT "WANT 'RANDOM FILE EDITOR' MENU?
<Y/N>";E$
490 IF E$ = "Y" THEN PRINT A$;"RUN FILE EDITOR,D1"
500 END

```

/FILE MERGER-MIDDLE/

```

10 REM *** FILE MERGER-MIDDLE 12/3/82 *** RICH WEST
20 REM *** ADDS FIRST HALF OF RECORDS FROM FILE 2 TO THOSE
FROM FILE 1 AND THEN REPEATS PROCESS FOR SECOND HALF
30 B = 2:C = 10: GOTO 50: REM B = DRIVE# 2. USE 1 IF USING
SINGLE DRIVE ONLY. C = RECORD LENGTH
40 PRINT A$;"WRITE";B$;" ,R";D: PRINT A(A): RETURN
50 DIM A(5000)
60 A$ = CHR$(4)
70 CALL - 936
80 PRINT : INPUT "NAME OF EXISTING FILE#1? ";C$
90 IF C$ = "CATALOG" THEN PRINT A$;"CATALOG,D";B: GOTO 80
100 HOME
110 PRINT : INPUT "NAME OF EXISTING FILE#2? ";D$
120 IF D$ = "CATALOG" THEN PRINT A$;"CATALOG,D";B: GOTO
110
130 HOME
140 PRINT : INPUT "UNIQUE NAME OF NEW FILE? ";B$
150 IF B$ = "CATALOG" THEN PRINT A$;"CATALOG,D";B: GOTO
140
160 ONERR GOTO 200
170 B$ = B$ + " L" + STR$(C): PRINT
A$;"RENAME";B$;" ,";B$;" ,D";B
180 PRINT : PRINT "****THIS IS NAME OF EXISTING FILE****":
GOTO 140
190 REM *** DATA FROM FILE 1
200 POKE 216,0: PRINT A$;"OPEN";C$;" ,D";B;" ,L";C
210 PRINT A$;"READ";C$;" ,R0"
220 INPUT E: REM NUMBER OF RECORDS
230 FOR F = 1 TO E
240 PRINT A$;"READ";C$;" ,R";F
250 INPUT A(F)
260 NEXT
270 PRINT A$;"CLOSE"
280 REM *** DATA FROM FILE 2
290 PRINT A$;"OPEN";D$;" ,L";C
300 PRINT A$;"READ";D$;" ,R0"
310 INPUT G: REM NUMBER OF RECORDS
320 H = F - 1
330 FOR F = 1 TO G
340 H = H + 1

```

```

350 PRINT A$;"READ";D$;" ,R";F
360 INPUT A(H)
370 NEXT
380 F = H
390 PRINT A$;"CLOSE"
400 PRINT A$;"OPEN";B$;" ,L";C
410 FOR D = 1 TO E / 2:A = D: GOSUB 40
420 NEXT
430 A = E
440 FOR D = ((E / 2) + 1) TO ((E / 2) + (G / 2))
450 A = A + 1: GOSUB 40
460 NEXT
470 A = E / 2
480 FOR D = ((E / 2) + (G / 2) + 1) TO (E + (G / 2))
490 A = A + 1: GOSUB 40
500 NEXT
510 A = E + (G / 2)
520 FOR D = (E + (G / 2) + 1) TO (E + G):A = A + 1: GOSUB
40
530 NEXT
540 PRINT A$;"WRITE";B$;" ,R0": PRINT A
550 PRINT A$;"CLOSE"
560 PRINT : INPUT "WANT 'RANDOM FILE EDITOR' MENU?
<Y/N>";E$
570 IF E$ = "Y" THEN PRINT A$;"RUN FILE EDITOR,D1"
580 END

```

/FILE VAR COMBINER/

```

10 REM *** FILE VAR COMBINER *** 12/10/82 *** RICH WEST
20 REM *** CREATES NEW RANDOM FILE CONTAINING SELECTED OR
COMPUTED VARIABLES FROM OLD FILE
30 C = 2:D = 10:E = 6000: REM *** C = DRIVE# FOR DATA DISK;
D = MAX RECORD LENGTH; E = ARRAY TOTAL
40 GOTO 330
50 CALL - 936:F = 0: REM SUB FOR OLD VAR
60 PRINT : PRINT "NEW VAR #";G: PRINT " = OLD FILE#";H;"
VAR #";: INPUT L
70 FOR M = 0 TO P
80 F = F + 1
90 A(F,G) = B(M,L)
100 NEXT
110 RETURN
120 CALL - 936:F = 0: REM SUB FOR DIFF
130 PRINT : PRINT "NEW VAR #";G: PRINT " = OLD FILE#";H;"
VAR #? - OLD VAR #?": INPUT " ?";A: PRINT " - ";: INPUT B
140 FOR M = 0 TO P
150 F = F + 1
160 A(F,G) = B(M,A) - B(M,B)
170 NEXT
180 RETURN
190 CALL - 936:F = 0: REM SUB FOR SUM
200 PRINT : PRINT "NEW VAR #";G: PRINT " = OLD FILE#";H;"

```

```

VAR #? + OLD VAR #?": INPUT " ?";A: PRINT " + ";: INPUT B
210 FOR M = 0 TO P
220 F = F + 1
230 A(F,G) = B(M,A) + B(M,B)
240 NEXT
250 RETURN
260 CALL - 936:F = 0: REM SUB FOR MEAN
270 PRINT : PRINT "NEW VAR #";G: PRINT " = MEAN OF OLD
FILE#";H;" VAR #? & OLD VAR #?": INPUT " ?";A: INPUT " & ";B
280 FOR M = 0 TO P
290 F = F + 1
300 A(F,G) = (B(M,A) + B(M,B)) / 2
310 NEXT
320 RETURN
330 B$ = CHR$(4)
340 CALL - 936
350 H = H + 1:K = 0: REM DATA FROM OLD FILE#2
360 IF H > 1 THEN PRINT : PRINT "NAME OF OLD FILE#";H;:
INPUT C$: GOTO 380
370 PRINT : INPUT "NAME OF OLD FILE#1? ";C$
380 IF C$ = "CATALOG" THEN PRINT B$;"CATALOG,D";C: GOTO
360
390 IF H > 1 THEN 510
400 HOME
410 PRINT : INPUT "NUMBER OF SUBJECTS IN OLD FILE? ";N
420 IF H = 1 THEN PRINT : INPUT "MAX NUMBER OF VARIABLES
IN OLD FILE? ";Q: IF Q < > INT(Q) OR Q < 1 OR Q > 50 THEN
GOTO 420
430 IF H = 1 THEN S = INT((E - (N - 1) - (N * Q)) / N):
IF S > 600 THEN S = 600
440 DIM B(N,Q),A(N,S)
450 PRINT : PRINT "NEW FILE CAN HAVE UP TO ";S;" VARIABLES"
460 PRINT : INPUT "UNIQUE NAME OF NEW FILE?";D$
470 IF D$ = "CATALOG" THEN PRINT B$;"CATALOG,D";C: GOTO
460
480 ONERR GOTO 510
490 D$ = D$ + " L" + STR$(D): PRINT
B$;"RENAME";D$;" , ";D$;" ,D";C
500 PRINT : PRINT "****THIS IS NAME OF EXISTING FILE****":
GOTO 460
510 POKE 216,0: PRINT : PRINT " *** WAIT ***"
520 REM *** DATA FROM OLD FILE#1
530 PRINT B$;"OPEN";C$;" ,D";C;" ,L";D
540 PRINT B$;"READ";C$;" ,R0"
550 INPUT R: REM NUMBER OF RECORDS
560 T = R / N: REM V=VARIABLES IN OLD FILE
570 FOR I = 1 TO N
580 FOR J = 1 TO T:K = K + 1
590 PRINT B$;"READ";C$;" ,R";K
600 INPUT B(I,J)
610 NEXT
620 NEXT
630 PRINT B$;"CLOSE"
640 IF H > 1 THEN 740
650 PRINT : PRINT "DO YOU WANT NEW FILE TO:"

```

```

660 PRINT : PRINT " 1 = USE ALL SUBJECTS"
670 PRINT " 2 = NOT USE ALL SUBJECTS"
680 PRINT : INPUT "WHICH? ";F
690 IF F = 2 THEN 720
700 O = 1:P = N
710 GOTO 740
720 PRINT : INPUT "NEW FILE SHOULD BEGIN WITH SUBJECT#?";O
730 PRINT : INPUT "NEW FILE SHOULD END WITH SUBJECT#?";P
740 CALL - 936
750 G = G + 1
760 PRINT : PRINT "NEW VAR# ";G;" IS:": PRINT
770 PRINT " 1 = SINGLE OLD VAR"
780 PRINT " 2 = DIFF BETWEEN TWO OLD VAR"
790 PRINT " 3 = SUM OF TWO OLD VAR"
800 PRINT " 4 = MEAN OF TWO OLD VAR"
810 PRINT " 5 = FINISHED WITH OLD FILE#";H
820 PRINT : INPUT "WHICH? ";U
830 IF U = 1 THEN GOSUB 50
840 IF U = 2 THEN GOSUB 120
850 IF U = 3 THEN GOSUB 190
860 IF U = 4 THEN GOSUB 260
870 IF U = 5 THEN 900
880 IF U = INT (U) AND U > 0 AND U < 6 THEN 750
890 GOTO 760
900 G = G - 1: REM DATA FROM SECOND OLD FILE
910 PRINT : PRINT "NEED DATA FROM SECOND OLD FILE?<Y/N>"
920 PRINT "(FILE MUST HAVE ";N;" SUBJECTS)";: INPUT A$
930 IF A$ = "Y" THEN 340
940 REM *** DATA TO NEW FLLE
950 PRINT B$;"OPEN";D$;" ,L";D
960 FOR I = 1 TO (P + 1) - O: REM NUMBER OF SUBJECTS
970 FOR J = 1 TO G:V = V + 1
980 PRINT B$;"WRITE";D$;" ,R";V
990 PRINT A(I,J)
1000 NEXT
1010 NEXT
1020 PRINT B$;"WRITE";D$;" ,R0"
1030 PRINT V: REM NUMBER OF RECORDS
1040 PRINT B$;"CLOSE"
1050 PRINT : INPUT "WANT 'RANDOM FILE EDITOR' MENU?
<Y/N>";E$
1060 IF E$ = "Y" THEN PRINT B$;"RUN FILE EDITOR,D1"
1070 END

```

/STAT MENU/

```

10 REM *** STAT MENU *** RICH WEST 12/30/82
20 A = 2: REM A = DRIVE#. SET A = 1 IN SINGLE DRIVE
SYSTEM
30 A$ = CHR$ (4)
40 HOME
50 PRINT : PRINT "PUT DATA DISK IN DRIVE#";A
60 PRINT : PRINT "SELECT ONE:"
70 PRINT : PRINT " 0 = STOP"

```

```

80 PRINT " 1 = BACKSTEP (REGRESSION ANALYSIS)"
90 PRINT " 2 = DEPENDENT T-TEST"
100 PRINT " 3 = INDEPENDENT T-TEST"
110 PRINT " 4 = GENERAL ANOVA"
120 PRINT : INPUT "WHICH? ";B
130 IF B = 0 THEN HOME : PRINT : INPUT "WANT 'FILE EDITOR'
MENU? <Y/N>";B$
140 IF B$ = "Y" THEN PRINT A$;"RUN FILE EDITOR,D1"
150 IF B$ = "N" THEN END
160 IF B = 1 THEN PRINT A$;"RUN BACKSTEP"
170 IF B = 2 THEN PRINT A$;"RUN DEPENDENT T"
180 IF B = 3 THEN PRINT A$;"RUN INDEPENDENT T"
190 IF B = 4 THEN PRINT A$;"RUN GENERAL ANOVA"
200 GOTO 40

```

/DEPENDENT T/

```

10 CLEAR
20 REM *** DEPENDENT T-TEST 12/16/82
30 REM *** MODIFICATIONS BY RICH WEST
40 E = 2:F = 10: REM ***ENTER DATA BY SUBJECT. E =
DRIVE#2. CHANGE TO E =1 FOR ONE DRIVE SYSTEM. F = RANDOM
FILE RECORD LENGTH
50 HOME
60 PRINT : PRINT "DEPENDENT T-TEST"
70 PRINT : PRINT "MODE OF DATA ENTRY:"
80 PRINT : PRINT " 1 = DATA FILE"
90 PRINT " 2 = INPUT FROM KEY BOARD"
100 PRINT " 3 = DATA STATEMENTS"
110 PRINT : INPUT "WHICH? ";H
120 IF H = 1 THEN 160
130 IF H = 2 THEN 200
140 IF H = 3 THEN 200
150 GOTO 50
160 B$ = CHR$ (4): PRINT : PRINT "PUT DATA DISK IN
DRIVE#";E
170 PRINT : INPUT "NAME OF DATA FILE? ";C$
180 IF C$ = "CATALOG" THEN PRINT B$;"CATALOG,D";E: GOTO
160
190 PRINT B$"OPEN";C$;",D";E;",L";F
200 PRINT : INPUT "NUMBER OF SUBJECTS? ";K
210 FOR I = 1 TO K
220 IF H = 3 THEN READ A,B
230 IF H = 2 THEN PRINT : PRINT "SUBJECT#";I;":": INPUT "
FIRST SCORE = ";A: INPUT " SECOND SCORE = ";B
240 IF H = 1 THEN L = L + 1: PRINT B$;"READ";C$;",R";L:
INPUT A
250 IF H = 1 THEN L = L + 1: PRINT B$;"READ";C$;",R";L:
INPUT B
260 LET N = N + A
270 LET O = O + B
280 LET Q = Q + (A - B)
290 LET R = R + (A - B) © 2

```

```

300 NEXT I
310 PRINT B$;"CLOSE"
320 PR# 1
330 PRINT CHR$(9) + "80N";
340 LET D = K - 1
350 LET S = N / K - O / K
360 U = SQR ((R - O ^ 2 / K) / D) / SQR (K)
370 T = S / U
380 PRINT
390 PRINT "GROUP 1 MEAN = ";N / K
400 PRINT "GROUP 2 MEAN = ";O / K;" (DIFFERNECE = ";N / K -
O / K;")"
410 PRINT " DEPENDENT T ="; SPC(2);T;
420 PRINT " (DF = ";D;)"
880 PR# 0
890 PRINT : INPUT "ANOTHER DEPENDENT T-TEST?<Y/N>";A$
900 IF A$ = "Y" THEN 10
910 PRINT CHR$(4);"RUN FILE EDITOR,D1"

```

/INDEPENDENT T/

```

10 CLEAR
20 REM *** INDEPENDENT T-TEST 12/16/82
30 REM *** MODIFICATIONS BY RICH WEST
40 E = 2:F = 10: REM ***ENTER DATA BY SUBJECT. E =
DRIVE#2. CHANGE TO E=1 FOR ONE DRIVE SYSTEM. F = RANDOM
FILE RECORD LENGTH
50 HOME
60 PRINT : PRINT "INDEPENDENT T-TEST"
70 PRINT : PRINT "MODE OF DATA ENTRY:"
80 PRINT : PRINT " 1 = DATA FILE"
90 PRINT " 2 = INPUT FROM KEY BOARD"
100 PRINT " 3 = DATA STATEMENTS"
110 PRINT : INPUT "WHICH? ";H
120 IF H = 1 THEN 160
130 IF H = 2 THEN 200
140 IF H = 3 THEN 200
150 GOTO 50
160 B$ = CHR$(4): PRINT : PRINT "PUT DATA DISK IN
DRIVE#";E
170 PRINT : INPUT "NAME OF DATA FILE? ";C$
180 IF C$ = "CATALOG" THEN PRINT B$;"CATALOG,D";E: GOTO
160
190 PRINT B$"OPEN";C$;","D";E;","L";F
200 FOR I = 1 TO 2
210 PRINT : PRINT "NUMBER OF SUBJECTS IN GROUP#";I;: INPUT
A(I)
220 K = 0:L = 0
230 FOR J = 1 TO A(I)
240 IF H = 3 THEN READ A
250 IF H = 2 THEN PRINT : PRINT "SUBJECT#";I;":": PRINT "
SCORE#";J;: INPUT " = ";A
260 IF H = 1 THEN O = O + 1: PRINT B$;"READ";C$;","R";O:

```



```
INPUT A
270 PRINT B$: REM CANCEL READ COMMAND
280 LET K = K + A
290 LET L = L + A ^ 2
300 NEXT J
310 LET B(I) = K / A(I)
320 LET C(I) = L - (K ^ 2 / A(I))
330 NEXT I
340 IF H = 1 THEN PRINT B$;"CLOSE"
350 LET D = A(1) + A(2) - 2
360 LET Q = B(1) - B(2)
370 R = ((C(1) + C(2)) / D) * (1 / A(1) + 1 / A(2))
380 LET S = Q / SQRT(R)
390 PR# 1
400 PRINT CHR$(9) + "80N";
410 PRINT
420 PRINT "MEAN GROUP 1 ="; SPC(2);B(1); SPC(2);"N =";
SPC(2);A(1)
430 PRINT "MEAN GROUP 2 ="; SPC(2);B(2); SPC(2);"N =";
SPC(2);A(2);
440 PRINT " (DIFFERENCE = ";B(1) - B(2);")"
450 PRINT " INDEPENDENT T ="; SPC(2);S;
460 PRINT " (DF = ";D;")"
920 PR# 0
930 PRINT : INPUT "ANOTHER INDEPENDENT T-TEST?<Y/N>";A$
940 IF A$ = "Y" THEN 10
950 PRINT CHR$(4);"RUN FILE EDITOR,D1"
```