Data Handling

Paul Muller
Ventura, CA

The OSI Superboard is an exceptional value for the computing power it provides. The 8K BASIC in ROM is ideal for writing number crunching programs which is my area of interest. After those first few weeks of wanton programming, however, I soon realized that an organized approach to data handling was needed. This article outlines a method I have used to structure the data handling chores for my numerical analysis programs. It is a series of BASIC routines which form a framework on which programs are built.

A completely flexible numerical program should be able to handle the following:

— Input data into the program from the keyboard
— Read data into the program off a tape
— Write data on to a tape
— Edit array data in the program
  List any part of the string or array
  Add elements to the string or array
  Delete elements
  Change elements
  Insert elements

Program 1 demonstrates these data handling routines. In this simple example, up to 50 numbers can be placed into two one-dimensional arrays, X and Y. The data can be examined, manipulated, stored, or retrieved from tape without leaving the program. To anyone who has written a statistical program, the advantages of these routines soon become evident.

The program consumes about 1800 bytes as written (not including the REMark statements). The amount of memory required will depend on the number and size of the arrays being used. The really determined Superboarder will undoubtedly find a way to squeeze the program into even less memory.

Here is a step-by-step look at the commands available in this program. If it sounds complicated, don't worry. The program takes all of five minutes to master.

INPUT: Calling the INPUT routine allows you to enter the data into arrays X and Y from the keyboard. Simply input the values for each element in the arrays per the program query. The element number of each input is shown also. To exit from the INPUT routine, simply enter END,END in response to the query.

Now that you have entered some data into the program, you will want to check it for accuracy. Calling up the EDIT routine provides the necessary commands.

LIST asks for the beginning and ending array element numbers for the data you want listed. The output shows the element number in the left hand column, and the data in X and Y residing in those elements.

To change an incorrect entry, enter the element number and the correct values of X and Y in response to the program query.

To delete data in X and Y, enter only the element number in response to the DELETE query. The same procedure is used for the insert command, but you must enter X and Y values as well.

The ADD routine picks up where the INPUT routine left off. To exit, enter END,END in response to the query. Note that terminating the ADD routine takes you out of the EDIT mode. The CNTL function in EDIT does the same thing.

To write the data on a cassette, call up the TAPE routine. Upon hitting the WRITE command, the program will ask you to enter a title pad. Simply repeat a suitable title and date until you get an overflow signal. The title pad will be 72 characters long, and will be written on the tape ahead of the data. This serves two purposes; to identify the data and keep recorder noise between data files from being picked up by the program. (The title pad is read into the program in the READ routine, but not used.)

After the title pad is entered, the program will instruct you to start the tape. When the recorder is running, enter any number and hit RETURN; a listing of the data will follow the title pad. When the end of the data in the two arrays is reached, the program will prompt you to stop the tape. All of the SAVE and LOAD commands are taken care of by the program.

The READ routine is similar. Since you are entering new data, exit the program using the STOP command. Using the Superboard's LOAD command, locate the data file in the tape, then bring up the program. The READ routine will prompt you to start the tape, which should be in the title pad of the data file. Enter any number (quickly!), and hit RETURN. The title pad and data will begin listing on your terminal. When END,END is encountered in the data file, the program will prompt you to stop the tape. You can then edit or add to the data using the other routines in the program.

The RUN command simply jumps to the starting line of the analytical portion of the program. In this example, the mean value of the X and Y arrays is calculated and displayed. The STOP command allows you to exit the program.

This program is intended only to demonstrate one technique of handling data. There is a lot of room here for more sophistication. You may want to modify this program to allow the reading of several tape files into the same array. Or maybe

you want to read data into different arrays. You could set up your program such that it would take in one set of data and write out a different set to be picked up by another program for further analysis. You could thus chain programs together in modules and increase the power of your Superboard far above its memory size. And who knows, maybe your friends with their dual disk drives and megabyte memories will actually envy your low-cost, customized "tape operating system!"

```
10 REM***DATA HANDLING PROGRAM***
20 REM***BY PAUL MULLER 8/81*****
30 REM***DIMENSION ARRAYS***
40 DIM X(50),Y(50)
50 REM***START CONTROL ROUTINE***
60 FOR J=1TO5:PRINT:NEXT
70 PRINT"INPUT(1),EDIT(2)":PRINT"TAPE(3)
   ,RUN(4)"
80 INPUT"OR STOP(5)";C
90 ON C GOTO 110,200,480,780,850:GOTO60
100 REM***START INPUT ROUTINE***
110 N=0
120 REM***START ADD ROUTINE***
130 FOR I=N+1TO50
140 PRINT"X";I;",Y";I:INPUTX$,Y$
150 IFX$="END"THEN60
160 X(I)=VAL(X$):Y(I)=VAL(Y$)
170 N=N+1
180 NEXTI
190 REM***START EDIT ROUTINE***
200 PRINT:PRINT:PRINT
210 PRINT"LIST(1),ADD(2)";PRINT"DELETE(3)
   ,INSERT(4)"
220 INPUT"CHANGE(5),CNTL(6)";C
230 ONCGOTO250,130,300,360,430,60
240 REM***ARRAY LIST ROUTINE***
250 PRINT":INPUT"ENTER I1,I2";I1,I2
260 FORI=I1TOI2
270 PRINT" ";I;X(I);Y(I)
280 NEXTI:GOTO200
290 REM***DELETE ROUTINE***
300 PRINT:PRINT"ENTER I FOR"
310 INPUT"DELETED X, Y";I
320 FORG=ITON
330 X(G)=X(G+1):Y(G)=Y(G+1)
340 NEXT G: N=N-1: GOTO200
350 REM***INSERT ROUTINE***
360 PRINT:PRINT"ENTER I,X,Y":INPUTI3,I4,I
   5
370 FORG=I3TON+1
380 I6=X(G):I7=Y(G)
390 X(G)=I4:Y(G)=I5
400 I4=I6:I5=I7
410 NEXT G:N=N+1:GOTO200
420 REM***CHANGE ROUTINE***
430 PRINT:PRINT"ENTER I,X,Y"
440 INPUT I4,I5,I6
450 X(I4)=I5:Y(I4)=I6
460 GOTO200
470 REM***TAPE ROUTINE***
480 PRINT:PRINT"READ(1),"
490 INPUT"WRITE(2),CNTL(3)";C
500 ON C GOTO 640,520,60
510 REM***WRITE ROUTINE***
520 A=PEEK(15):POKE 15,72
530 PRINT"ENTER TITLE PAD":INPUT P$
540 INPUT"START RECORD";C:SAVE
550 PRINTP$
560 P$="-"
570 FORI=1TON
580 PRINTP$;",";X(I);",";Y(I)
590 NEXT I:PRINTP$;",END,END"
600 POKE 517,0
610 POKE 15,A:PRINT"STOP TAPE"
620 GOTO 480
630 REM***READ ROUTINE***
640 A=PEEK(15):POKE15,72
650 PRINT:INPUT"START PLAY";C
660 N=0:LOAD
670 INPUT P$
680 FORI=N+1 TO 50
690 INPUT P$,X$,Y$
700 IFX$="END"THEN 740
710 X(I)=VAL(X$):Y(I)=VAL(Y$)
720 N=N+1
730 NEXT I
740 POKE 515,0
750 PRINT"STOP TAPE":POKE 15,A
760 GOTO 60
770 REM***START ANALYTICAL PART OF PROGRA
   M***
780 Z=0:Z2=0
790 FORI=1TON
800 Z=Z+X(I):Z2=Z2+Y(I)
810 NEXT I
820 PRINT:PRINT"MEAN X=";Z/N
830 PRINT"MEAN Y=";Z2/N
840 GOTO 60
850 END
```

©