

Timex Sinclair User Group NEWSLETTER

Editor-R. Mulder
Designer-J. Turner

Ottawa Chapter

VOLUME II ISSUE V

Table of Contents

Table of Contents	
President' Address.....	■■■■■
From the Editor.....	■■■■■
General.....	■■■■■
Forth Corner.....	■■■■■
Personals.....	■■■■■



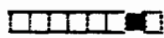
Advertising Rates

Members - 45 words or less, FREE

Outside - 45 words or less, 5.00

Commercial:

- Full Page - 30.00
- Half Page - 22.00
- Quarter Page - 12.00
- Eighth Page - 8.00



PRESIDENT'S ADDRESS

Alas, summer is over and it won't be too long before we'll be up to our armpits in you know what. It's too bad that a way couldn't be found to use the ZX machines to alleviate this perennial headache. I suppose if all else failed you could nail the ZX to a broom handle.

As most of you are probably aware, the attendance at the summer meetings, although better than expected, was insufficient to conduct a useful program of activities. As in my own case, many members were unable to attend these meetings because of other more pressing commitments. Perhaps we should consider formally cancelling the July and August meetings. This would not preclude interested members from meeting informally however if they should so desire. I'll raise this matter at our next meeting.

I mentioned in the previous newsletter that Sir Clive was giving up control of Sinclair Research to publisher Robert Maxwell. Rumour has it that the deal fell through and Sinclair has gone into receivership. This does not come as any surprise to me; the bubble has burst in the home-computer market and only the most hardy manufacture with a diverse product line will still be in business this time next year. IBM, Apple and Commodore are my guesses. Watch for some good deals from Sinclair vendors for the QL if the rumour is true. Expect third-party vendors dropping support for the Sinclair line and switching to other computers just as they did when TIMEX went belly up.

If Sinclair is bankrupt, we are really orphans and it is essential that we stick together and share in order that the club will continue to provide a meaningful experience for each of us. Like the Edsel, computers will continue to come and go and there is no way one can keep abreast of all the latest developments. The fundamentals remain the same however and what you learn on

the Sinclair is directly applicable to any computer regardless of the number of bells and whistles they may contain. A computer is a computer.

LARKIN TS2068/SPECTRUM DISC INTERFACE SYSTEM

As I indicated in the previous newsletter, I bought a disc interface from LARKIN Electronics. LARKIN Electronics is owned by Larry Kenny who is a member of the club. Here is a brief assessment of the system as I found it to date.

The system I purchased consisted of an interface card approximately 7" x 4", a single Shugart SA455 disc drive and the cabling necessary to interconnect the units to each other and the computer. None of the hardware was in cabinets. A temporary power supply was provided. The total price for the system was \$320 Canadian.

When I first installed the system (Larry Kenny was present) the system would not run if my RGB monitor interface card was connected between the computer and the disc system. After several tests by Larry, it appeared that the cause of the problem was the length of the cable connecting the interface card to the computer. A shorter cable was made up and the system worked as advertised.

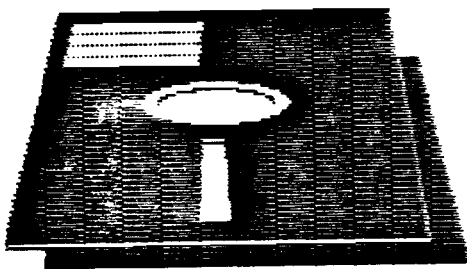
Since the system is uncased, my first step was to fabricate cabinets to install the system. The disc drive and interface card were placed in one cabinet approximately 4" wide, 8" deep and 3" high. The power supply was reworked to provide better heat dissipation and installed in a modified Radio Shack cabinet. Computer connection was via an edge connector that was hard mounted to the disc cabinet. A 3 core cable interconnected the disc cabinet to the power supply. To accomodate having the ZX 81 edge connector on the disc interface board extending from the rear of the cabinet (so that the 2040 printer could be hooked up), the ribbon cable connecting the disc drive and the interface card was lengthened by 4", the computer ribbon cable was extended 1" and the connectors on the computer interface ribbon cable were reversed.

When the equipment was installed in the cabinets and connected to the computer via a John Olinger expansion board, the system experienced the same problems I had at the outset. I found by adjusting a variable resistor on the interface board that the problem vanished. The system is now working and to date has been reliable. The system was left running for 48 hours to coax out any problems but none occurred. As with any new system there are teething problems but those found to date are inconsequential.

The system could be improved but considering the price tag it is a good buy. Some first-impression suggestions for improving the system are as follows:

- 1) the system should be sold as a complete system including cabinets; few people have the tools or skills to muster up the parts and package the system
- 2) the peripheral bus should be a standard 2008 bus rather than the ZX81 bus
- 3) the operating system should be modified to accommodate the keyboard disc instructions rather than USA calls
- 4) disc formatting in 1 K tracks would save memory

As I indicated earlier, this is only a brief assessment of the system and I intend to provide a more comprehensive report later.



FROM THE EDITOR

Well what do you know, input from someone else!!! I may just have a heart problem with all this excitement.

No, seriously folks, I applaud those who have come out of the woodwork so to speak and have put in their two cents worth. May I also add that I am most pleased with everything.

The input from these people represents what it is that they are concerned with and where they may wish to see additional support. You do not have to be an expert. You must realize that there are things that you may discover that have many others stumped.

Who knows, you may even surprise the so called experts in their own fields. (if you ask me..I think that's rather easy-- they are all bluffing anyways).

Keep it up !!!!!!!!!!!!!!!

The Editor, (tongue in cheek)

GENERAL

Tips for loading FORTH
(ZX-81/TS1000)

by David Solly

Every computer enthusiast enjoys buying and running new software but that enthusiasm can be dampened if the program fails to load or run according to the instructions outlined in the user's manual. This was my experience when first trying to load and run Artic ZX FORTH for the T/S 1000. The following information I hope will save the users of ZX FORTH some frustration.

First of all the log on is LOAD "ZXFORTH" rather than what is so boldly stated in the user's manual. This is at least a first step towards a successful load.

Secondly for some reason known only to itself and its creator the T/S 1000 is reluctant to load FORTH screens from the beginning of a tape or

the silent section between programs. A more sure fire method is the following:

- 1) When the first part of the program has loaded and the announcement comes up stop the recorder.
- 2) Type 1 LOAD (or the number of whatever screen you are loading).
- 3) Rewind the tape to the tail end of the last section loaded.
- 4) Start the tape.
- 5) Hit Enter (New Line).

The VDU display will white out but not to worry! The program has not crashed! In a few seconds the loading bars will appear and the T/S 1000 will perform a normal load. Repeat steps 3 to 4 to load the remaining FORTH screens.

When saving FORTH screens on a fresh tape record some line noise or a menu in BASIC at the beginning before saving the first screen. Reload at the line noise or at the tail of the first program. This method should save anyone a lot of trouble in loading and saving ZX FORTH and its screens.

Copy to Ottawa Sinclair User's Group on 3 Sept. 1985.

Report from the Librarian

by David Solly

For those who missed my announcement at August's meeting I wish to inform you that I shall not be present at meetings until at least the end of December because I shall be attending classes on data processing in libraries at Algonquin College. During this time Bill Harmer will be taking care of bringing materials to and from meetings although you may continue to make reservations for library materials by calling me at 731-2120.

In the mean time I will remain active behind the scenes. At the moment I am rerecording some of the programs in the library into collections according to themes in order to help alleviate the problem of lending only one article to each person at a time. Look for:

Astronomy #1 [2000/2068],
Statistics Kit #1 [1000],
Electronics #1 [1000] and others. I also expect to finish very shortly a complete listing of the library holdings including books and periodicals thanks to the recent acquisition of a copy of M Script. I shall also let you know when our first issues of ZX Computing arrive.

Please! Please! Please! Contribute programs to the library. At the moment we are woefully short of any sort of programs for the T/S 2068 and the Spectrum.

Also remember the adage 'Public Domain Software'

NEWS

Last July, I was London, England and found the Spectrum and QL widely available. The Spectrum was selling for approximately what it sells for in Canada. In other words, we are getting a fair shake in Canada for our money.

I also was in Greece, and was surprised that the Spectrum appeared to be the only personal computer computer widely sold in that country. Of course, the manuals were all in English, but most Greek young people speak English. There were also a few books available in Greek.

Here in North America, Zebra Systems of New York City are now offering Spectrum emulator cartridges for the 2068 for U.S.\$19.95. (These cartridges will run 95% of all Spectrum software. At this price they are not making any profit. But they explain they want to sell Spectrum software, and without Spectrum capability, there is no North American market for Spectrum software.

Zebra is also marketing a disk drive for the 2068 for \$299.95 (U.S.) They also have an "advanced compiler" for the 2068 at \$19.95. For Byte-Back modem owners they have software which permits X-MODEM file transfer protocol. In fact they have loads of other hardware and

software for the ZX-81, TS1000, QL, and 2068. Write to them for a free catalogue, at:

Zebra Systems Inc.,
78-06 Jamaica Ave.,
Woodhaven, N.Y.,
U.S.A. 11421
Telephone: (718) 296-2385

A good source for Spectrum software in England is:

Software Supermarket,
87 Howard's Lane,
London, SW15 6NU,
England

Both the above firms accept VISA and Mastercard.

For those with Spectrum capability on their 2068's, the following are the current top 10 Spectrum software sellers in Great Britain:

1. Match Day
2. Atic Atak
3. Daley Thompson's Decathlon
4. Knightlore
5. Ghostbusters
6. Bruce Lee
7. Underworld
8. Raid Over Moscow
9. Spy Hunter
10. Alien 8

If we are going to add to the User Group's Library, some thought should be given to the above titles.

Well, that's it for this time. Happy computing to you all.

-Leo Foss

MCODERII BASIC COMPILER

by: Bill Harmer

The advantage of a basic compiler over the built-in Basic interpreter, is that the compiler is usually faster. It saves time by doing the conversion from basic to machine language in advance in one shot (compiling) so that when it is run (execution) it does not take this time doing the interpretation.

McoderII is an improved ZX-81 version. The first thing one notices however, when one goes ahead and tries to convert a basic program is that McoderII

rejects many of the statements that would normally run good in Basic. This is partially due to hard to pinpoint ways in which the compiler differs from that of the resident interpreter and partly due to the fact that some statements do not exist in the McoderII compiler.

This means that you will have to rewrite the offending statements to suit McoderII or if that is not possible then you will have to develop subroutines that will do the same thing. Some programs after some study, may not be convertible. Alphabetic sortation is a definite problem. The solution is obviously the assignment of code values to the letters.

As far as changing your Basic programs to run under McoderII, a list of fixes follows this article.

The most striking aspect of McoderII is the only INTEGER mathematics; it will not accept decimal constants, variables or output numbers and when you try to program around that, you may run into the limit on the largest integer that it will accept ie ..(32768). However these problems do not appear unique to McoderII judging from literature from other compilers. Here are a few fixes for some of the problems:

```
-IF A<B THEN PRINT "A IS LESS THAN B" must be changed to....
  IF B>A THEN PRINT "A IS LESS THAN B"
-IF A=B THEN PRINT "A EQUALS B" must be changed to....
  10 IF A>B THEN GOTO 40
  20 IF B>A THEN GOTO 40
  30 PRINT "A EQUALS B"
  40 .....
-IF B<C AND B>A THEN PRINT "B IS BETWEEN A AND C" MUST BE CHANGED TO.....
  10 LET Y=0
  20 IF B>A THEN LET Y=1
  30 IF C>B THEN LET Y=Y+1
  40 IF Y>1 THEN PRINT "B IS BETWEEN A AND C"
- THE EXPONENTIAL NOTATION OF 1E3 IS NOT ALLOWED AND MUST BE CHANGED TO ITS ACTUAL DECIMAL REPRESENTATION, SUCH AS 1000. (beware the 32768 limit)
-IF A$="Y" THEN PRINT "YOU CHOSE YES" should be changed to....
  10 LET A=CODE "Y"
  20 IF A>(CODE A$) THEN GOTO 50
  30 IF (CODE A$)>A THEN GOTO 50
  40 PRINT "YOU CHOSE YES"
  50 .....
```

end.....

ODDS AND ENDS

3D TANK DUEL

There is a need to make backup copies of programs purchased on cassettes. A very useful program to make backup copies of just about any program is 007 and its relatives (TRICKY, MAXSPY, etc...). However, there are some programs currently being sold that have purposely been made resistant to 007.

3D TANK DUEL is such a program. To see how the protection mechanism works, simply load the second part of the program as follows:

LOAD "" CODE

Observe that the program name will come out as TANK printed vertically downward, starting from HOME position. This effectively messes up the Display file where MAXSPY (son of 007) transfers its Load routines. You can get around this obstacle by creating your own dummy label or by proceeding as follows:

1. Load the first part of the program with the command MERGE "", then save with SAVE "3DTANKDUEL" LINE 10.

2. Enter a RUN command to load the main program. Near the end of the load, keep the "Q" key pressed. Once loaded, the program will hold for a short length of time (one second) provided "Q" is still pressed. During that time, press CAP/SHIFT and then BRK, and the program should stop. Simply GOTO 100 to save (The pretty screen is lost, however).

Marcel G Boulton.

TASWORD II AND A BLACK BORDER

Some users of this excellent package may prefer a black border and light ink on dark paper. It can be done as follows:

Enter the BASIC mode to get access to the BASIC portion of TASWORD II. Edit line 10 and insert at the beginning of the line the additional commands BORDER 0:PAPER 0:INK 7. Then POKE addresses 64840, 64841, and 64842 with 0. Enter a RUN command and voila!

Marcel G Boulton.

Logical statements

THIS INFORMATION on string arrays and logical arithmetic may be of interest.

Sinclair ZX-81 Basic lacks an IF ... THEN ... ELSE statement. In one specialised context, however, this can be mimicked with the use of string arrays, thus saving program space.

Since one cannot have, for example:

IF N=1 THEN PRINT A\$(7) ELSE PRINT A\$(8), one might seem to need two statements:

IF N=1 THEN PRINT A\$(7)

IF N<>1 THEN PRINT A\$(8)

In fact, they can both be replaced by the single statement PRINT A\$(NOT N-1).

A similar effect can also be produced, so long as N is not zero or negative, by

PRINT A\$(7 + SGN(N-1)); but that prints A\$(6) if N is zero or negative and in any case needs an extra pair of brackets.

The idea can be extended, provided always that the choice is between strings in the same array. For example:

IF N>8 AND N<13 THEN PRINT A\$(25)

IF NOT (N>8 AND N<13) THEN PRINT A\$(6) can be replaced by

PRINT A\$(25-19*NOT (N>8 AND N<13)).

The same can even be done for more than two conditional statements;

for example, the four lines IF A = 3 AND B > 4 THEN PRINT A\$(7)

IF A = 3 AND B <= 4 THEN PRINT A\$(8)

IF A <> 3 AND B > 4 THEN PRINT A\$(23)

IF A <> 3 AND B <= 4 THEN PRINT A\$(24)

can all be replaced by the single

PRINT A\$(7 + 16*(NOT A=3) + NOT B>4)

with a considerable saving of space.

LARKEN DISK-DRIVE

Ive got one too.....

By Jim Turner

As a matter of fact, I think I've got the first one Larry sold. Let me go on about it for a bit.

The first drive controller and drive that Larry Kenny brought me had a few problems to say the least. It wouldn't work with my computer. Larry had developed the whole system using his own 2068 and it appeared that the tolerances he used weren't wide enough. He borrowed my computer and spent some sleepless nights getting his system to work on, not just my computer but, any 2068.

That problem out of the way, the next turned up in the power supply he had whipped up for me. It would get so hot, that I couldn't touch it and eventually, one of the resistors or something (Hey... I never said I was a techie) any way something quit and Larry had to come to the rescue again.

That seemed to take care of the technical problems, but, after I started using the drive a bit, it became obvious that the DOS had a few as well. Nothing major, just annoying little things that made adapting to using a disk drive a bit harder...for instance:

-at first, Larry had used the shifted "5" to move back through a typed in command. this movement also deleted the character to the left, just like the delete key on the 2068. I found myself constantly hitting the delete key instead of shifted "5".

I spoke to Larry about this and he changed the DOS accordingly.

-another problem; if code has to be saved, one must do some fancy POKING of 2 byte hex values before saving. This was quite a problem for me since I don't speak "HEX". I solved this by writing a little program that would take a CODE file from tape and save it to disk. -The biggest problem...Me I had to make myself think in a whole new way. Programs load so fast from disk that I can think of individual programs almost as subroutines. Imagine, a basic

program that is only a menu routine. It accepts your selection and loads the appropriate program in under 3 seconds.... does it's job and loads another. It's a big step from 2068 tape loading. It's an enormous step from the TS 1000 tape loading.

-The system comes uncased. Not a big problem for some. Those who already have 2 feet of peripherals extending from the back of their machines won't find this too disconcerting but for me, my recurring nightmare is about a 10 year old kid spilling chocolate milk on an exposed circuit, and the lights going out all over town.

I've got a plan for a case but as yet, it's just a plan. When it's finished, I'll bring it to one of our meetings and blow you all away by loading a program like UU-3D in just over 6 seconds... (eat your hearts out)

Simply, the system has it's short-comings. But it does everything Larry designed it to do, at a price that is still below what Commodore owners have to pay for their disk systems that are only slightly faster than tape.

The Piece de resistance..

It works on Spectrum too!!!!

No other disk drive system for our computers can boast this feature. Those of us who have invested in a ROM switch and have experienced the software available from Jolly olde England will know the significance of this statement.

TWO ZX-81'S - UHF

=====

full size keyboards

2040 printer

Hunter board

plus much more

CALL FOR DETAILS

824-8734

THE ZX-81/TS1000 NOTEBOOK

From the book "GOSUBS" by Edwin and Shirley Gaby:

This subroutine checks to see if another line can be printed on the screen. If not, it scrolls and sets up the print on line 21.

Substitute your own line numbers for X, for instance X00 can be 500.

```
X00 REM auto-scroll
X10 if peek 16442 = 2 then
  scroll
X20 if peek 16442 = 2 then
  poke 16442,3
X30 return
```

The following program will accept your input and place it into a properly dimensioned array. It places your data into an array A(1) and also provides N (the number of elements in the array), MX (the maximum value in the array) and MN (the minimum value in the array). If MX and MN are not needed, delete the appropriate line numbers.

```
1 REM array input
10 let mx=0
20 print "Input Number of
  items in data"
30 input n
40 dim a(n)
50 cls
60 for i = 1 to n
70 print "a(";i;") = ";
80 input a(i)
90 if i = 1 then let
  mx=a(i)
100 print a(i)
110 if a(i)>mx then let
  mx=a(i)
```

```
120 if a(i)<mn then let
  mn=a(i)
130 gosub X00 : rem autoscroll
140 next i
```

NOTE: line 130 helps avoid error code 5

Submitted: Orville Kenny

CONVERTING PROGRAMS TO ZX-81

This article is the beginning of a series on converting programs intended for other computers to ZX-81 Basic.

Since general programming ideas are discussed as well, it will also be of interest to all that wish to gain more insight into and ideas for writing programs.

MATHEMATICS

Other basics allow a formula to be expressed in the form of a simple equation thus: C=B-A , but with ZX basic you must always preface such a formula with LET. Other than that and the fact that exponentiation is indicated in some basics as an up arrow or a ^ , there should be no problem with arithmetic in Basic. (even FORTRAN uses the same sort of formulas, symbols and variable names as is allowed in the ZX

Sometimes LOG is used in other basics to mean what the ZX calls LN, but both are log to base e=2.71828 rather than log to base 10 that we studied in school. (the inverse function of LN is EXP which raises e to the power specifie by the nuber following). If you get LOG in a program to convert, just convert it

to LN. If however you really need log to base 10 in your program or mathematics, you can use the formula:

```
LET LOG10= (LN(X))/(LN(10)).
```

Note that some basics require that the number that LN operates on to be in brackets.

One thing that can be a real problem is when DEF and FND or FNR is used.

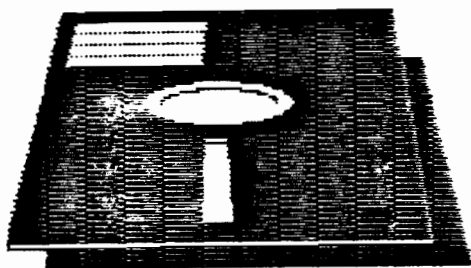
```
eg. 10 DEF FND(X)=X**3
    20 LET Y=FND(3)
    *would return Y=27
```

This allows a programmer to input a function into the program. The following is a method that the writer has used to duplicate this for the ZX-81:

```
10 print "input function
    equation or value"
20 input a$
30 let y = VAL a$
```

This is based on the principle that VAL will evaluate the string that follows it as an arithmetic statement.

If MOD occurs in a program or the program is intended for integer basic, all the divisions should be preceded by INT. The example of Y MOD X can be replaced by Y-(INT(Y/X)*X).



FORTH CORNER

Sorry folks, no Forth Mind this issue. As many of you may know, I have been working overtime so to speak on my disk drive and the conversion of tape programs to disk.

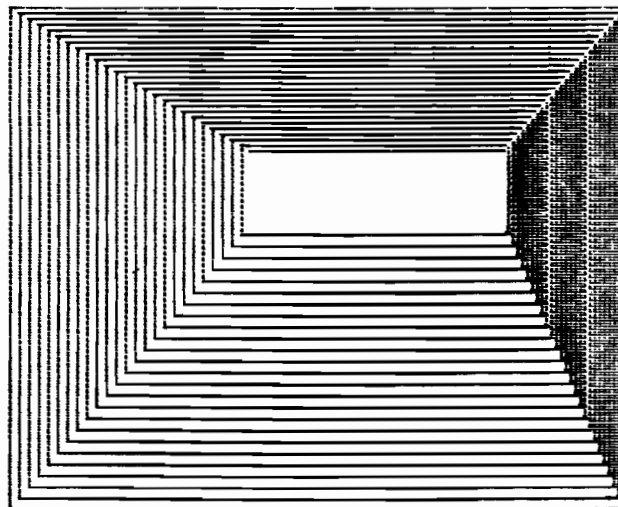
The Forth programs that I have been using ie.. Abersoft and White Lightning are Spectrum programs and my attempts to convert these to the 2068 have so far failed. Obviously I am not a good M/C programmer.

Knighted Computers to the rescue. Shades of the White Knight! They have obtained or converted White Lightning to run on the 2068. Now hopefully I will be able to transfer it to disk and run it from there.

Until then, most likely to the next issue, this section will address tutorials.

Let's hear from all those Forth users out there.

THE EDITOR




```

670 INPUT "ENTER date of last day of
week: ";a$
680 PRINT AT 3,1:
"week ending ";a$
700 RESTORE
710 FOR i=1 TO 7
720 INPUT " ";a$
730 PRINT AT 2,6+3+i:a$
740 IF i<4 THEN
PRINT AT 2,5+i,1;2+i-2
750 NEXT i
80000 DATA "M","T","W","T"
81000 STOP
82000 SAVE "bar" LINE 1
    
```

```

4 REM THE DISPLAY IS PRINTED
OUT ONTO A 8*8 GRID
9 REM ***INITIALIZATION***
10 RANDOMIZE
20 DIM A(100)
30 DIM B(100)
40 DIM E(8)
50 LET E(1)=-11
60 LET E(2)=-10
70 LET E(3)=-9
80 LET E(4)=-1
90 LET E(5)=1
100 LET E(6)=9
110 LET E(7)=10
120 LET E(8)=11
130 LET C=30
139 REM ***RANDOM CELLS****
140 FOR Z=1 TO 25
150 LET R=INT (RND*100)+1
160 LET A(R)=C
170 NEXT Z
175 LET g=0
178 PRINT AT 10,10;"GENERATION"
180 GO TO 390
189 REM ***CALCULATIONS***
190 FOR X=1 TO 100
200 LET CO=0
210 FOR B=1 TO 8
220 LET B1=X+E(B)
230 LET B1=B1+(100 AND B1<1)-(1
00 AND B1>100)
240 IF B(B1)=C THEN LET CO=CO+1
220 LET B1=X+E(B)
230 LET B1=B1+(100 AND B1<1)-(1
00 AND B1>100)
240 IF B(B1)=C THEN LET CO=CO+1
250 NEXT B
260 IF B(X)<>0 AND CO<>2 AND CO
<>3 THEN LET A(X)=0
270 IF B(X)=0 AND CO=3 THEN LET
A(X)=C
280 NEXT X
390 PRINT AT 10,21;G
399 REM ***DRAW ARRAY***
410 FOR Y=0 TO 8
420 FOR X=1 TO 10
430 LET B1=Y*10+X
440 PRINT AT Y,X+10;CHR$(A(B1)
+128)
450 LET B(B1)=A(B1)
460 NEXT X
470 NEXT Y
490 LET G=G+1
500 GO TO 190
    
```

BINARY LOOK-SEE

The other day I needed to be able to see the binary bit patterns of certain decimal numbers compared to the bit patterns of others.....

Here is the resulting program. Short and sweet and it works.

Jim Turner

```

10 INPUT a
20 LET w=a
30 LET b=2
40 DIM a$(8): LET c=128
50 FOR x=8 TO 1 STEP -1
60 LET a$(x)=" " AND INT (A/B
)<>(A/B)): LET a=INT (A/B): LET
C=C/B: NEXT x
70 PRINT " ";a$;" =";w
80 GO TO 10
    
```

GENERATION

```

1 REM FOR THE 2068 RUN WITH I
ORDER 0,PAPER 0,AND INK 7
2 REM FOR THE ZX81 MAKE THE F
OLLOWING CHANGES
5 FAST
130 LET C= 52
400 SLOW
480 FAST
10 RAND
3 REM THIS PROGRAM WAS TAKEN
FROM THE GATEWAY GUIDE TO THE ZX
81 AND ZX80 PAGE 77,78
    
```

MENUS MENUS MENUS MENUS

I was working on a program the other day that required several menus to be called during the run. Some of the menus were even choices on other menus.

While working, it seemed to me I was writing the same code over and over for each menu. I decided that I would write a sub-routine that would print them all.

The menus are contained in DATA statements and are structured like this:

- number of choices on menu
- maximum number of characters in a choice
- name of menu
- then texts to all the choices

The little demo program demonstrates that one need only RESTORE to the correct DATA statement, then call the sub-routine. The selection is made using only the space bar and ENTER returns to the main program with the proper value in the variable "CHOICE".

As you can see, a choice of "3" on the first menu will automatically call the sub-menu and add 10 to the value of the choice made on the second menu.

I'm not really sure if much code is saved by using this technique for a small application like this one but if a program required 5 or 10 menus to be called at one time or another, then this might be of some use.

```

8010 READ X,Y: REM X=NUMBER OF
      CHOICES
      Y= MAX LENGTH
      OF CHOICE
8020 DIM C$(X,Y): DIM A$(Y)
8025 READ N$: PRINT AT 1,15-(LEN
      N$/2);N$
8030 FOR A=1 TO X: READ C$(A): N
      EXT A
8040 LET ROW=INT (10-(X/2)): LET
      COLUMN=15-(Y/2): LET POSITION=R
      OW
8050 FOR A=0 TO X-1: PRINT AT RO
      W+A,COLUMN;C$(A+1): NEXT A
8060 GO SUB 8200
8070 IF INKEY$="" THEN GO TO 807
      0
8080 LET Y=CODE INKEY$: IF Y=32
      THEN GO SUB 8200: LET POSITION=P
      OSITION+(1 AND POSITION<(ROW+X-1
      ))-(X-1 AND POSITION=(ROW+X-1)):
      GO SUB 8200
8090 IF Y=13 THEN GO TO 8110
8100 GO TO 8070
8110 LET CHOICE=(POSITION-ROW)+1
8120 CLS : RETURN
8200 PRINT AT POSITION,COLUMN; I
      NVERSE 1; OVER 1;A$
8210 RETURN
8500 REM
8501 REM
8502 REM
9000 DATA 10,10,"MAIN MENU","CHO
      ICE #1","CHOICE #2","CHOICE #3",
      "CHOICE #4","CHOICE #5","CHOICE
      #6","CHOICE #7","CHOICE #8","CHO
      ICE #9","CHOICE #10"
9010 DATA 5,9,"SUB MENU","CHOICE
      #1","CHOICE #2","CHOICE #3","CH
      OICE #4","CHOICE #5"
  
```

Try this:

PLOT 68,28: DRAW 100,100,4100

```

100 REM *****
      * DEMO *
      *****
110 RESTORE 9000: GO SUB 8000
120 IF CHOICE=3 THEN RESTORE 90
10: GO SUB 8000: LET CHOICE=CHOI
      CE+10
130 PRINT AT 10,10;"CHOICE =";C
      HOICE: STOP
8000 REM *****
      * MENU SUB-ROUTINE *
      *****
8001 REM
8002 CLS
  
```

CASSETTE TAPE RECORD

I needed a way to keep track of my growing library of programs. Vu-File seemed like it might do the trick but it was a bit cumbersome to load both files each time. Besides, I'd never written any sort of record program before so I decided to do the definitive BASIC tape record file. (definitive because it's the only one I've written

and BASIC because it's the only computer language I know)

It's menu driven, saves only the number of records entered (not a huge DIMmed string variable which is usually half empty), prints to the screen or printer and sorts two different ways (alphabetically and by tape number). By the way, when the program sorts by tape number, it also sorts the programs on the tape by side and counter location. You can search for a single program or a whole tape.

The program uses a Basic bubble sort and as a result can take a few minutes to do its job when the record reaches a hundred entries or so.

Jin Turner

- 1. Add a program.
- 2. Delete a program.
- 3. See programs.
- 4. Sort programs.
- 5. Save program file to tape.
- 6. Load a program.

```

10 REM      Tape Record
15 ON ERR GO TO 1000: GO TO 10
00
20 DIM p$(99,16): LET prog=0
1000 REM MEMO
1005 PAPER 1: INK 7: BORDER 1
1010 CLS : PRINT AT 7,10:".. /MEMO
MEMO": AT 7,0: OVER 1: "
1020 PRINT TAB 3:"1. Add a prog
ram." TAB 3:"2. Delete a progra
m." TAB 3:"3. See programs." TAB
3:"4. Sort programs." TAB 3:
"5. Save program file to tape."
TAB 3:"6. Load a program."
1030 LET z$=INKEY$: IF z$>"0" AN
D z$<"7" OR CODE z$=7 THEN GO TO
1037
1035 GO TO 1030
1037 IF CODE z$=7 THEN GO TO 700
0
1040 LET z=VAL (z$)+1
1050 CLS : GO TO (2000 AND z$="1
")+ (8000 AND z$="2")+ (3000 AND z
$="3")+ (4000 AND z$="4")+ (5000 A
ND z$="5")+ (6000 AND z$="6")
1100 REM MEMO
1110 PRINT AT 21,0:p$(p,1 TO 10)
: " MEMO ";p$(p,11 TO 12);TAB 19;
" MEMO ";p$(p,13);TAB 26; " MEMO ";p$
(p,14 TO 16)

```

```

1130 PRINT : RETURN
1200 LPRINT AT 21,0;p$(p,1 TO 10)
: " MEMO ";p$(p,11 TO 12);TAB 19
: " MEMO ";p$(p,13);TAB 26; " MEMO ";p
$(p,14 TO 16)
1210 LPRINT : RETURN
1300 POKE 23692,21: PRINT AT 21,
0: PRINT : RETURN
2000 REM MEMO
2010 GO SUB 1300: POKE 23658,8:
LET prog=prog+1
2020 INPUT "Name";z$: IF LEN z$>
10 OR z$="" THEN BEEP .5,-20: GO
TO 2020
2030 LET p$(prog,1 TO 10)=z$
2040 INPUT "Tape number ";z$: IF
LEN z$>2 THEN GO TO 2040
2042 FOR z=1 TO LEN z$: IF z$(z)
<"0" OR z$(z)>"9" THEN GO TO 204
0
2044 NEXT z: LET p$(prog,11 TO 1
2)=( " " AND LEN z$=1)+z$
2046 INPUT "Side ";z$: IF LEN z$
>1 THEN GO TO 2046
2048 IF z$<>"1" AND z$<>"2" AND
z$<>"A" AND z$<>"B" THEN GO TO 2
046
2050 LET p$(prog,13)=z$
2060 INPUT "Counter #";z$: IF L
EN z$>3 THEN GO TO 2060
2062 FOR z=1 TO LEN z$: IF z$(z)
<"0" OR z$(z)>"9" THEN GO TO 206
0
2064 NEXT z: LET p$(prog,14 TO 1
6)=( " " AND LEN z$<=2)+( " " AND
LEN z$=1)+z$
2070 LET p=prog: GO SUB 1100
2080 PRINT AT 21,0:"Correct?": L
ET z$=INKEY$: GO TO 2080+(5 AND
z$="N")+ (10 AND z$="Y")
2095 PRINT AT 21,0:" " : G
O SUB 8500: GO SUB 1300: PRINT "
MEMO "
2100 PAUSE 0: PRINT AT 21,0:" "
GO TO 2100-(1100 AND INKEY$="N")
-(100 AND INKEY$="Y")
3000 REM MEMO
3010 POKE 23658,8: PRINT AT 19,0
:"1. Whole file" TAB 10:"2. One pr
ogram" TAB 17:"3. One tape":
3015 LET z$=INKEY$: IF z$="" THE
N GO TO 3010
3020 IF z$="1" OR z$="2" OR z$="
3" THEN GO TO (3030 AND z$="1")+
(3500 AND z$="2")+ (3800 AND z$="
3")
3025 GO TO 3010
3030 GO SUB 3900: CLS : GO SUB 1
300: FOR p=1 TO prog:
3035 IF print=1 THEN GO SUB 1100
: PRINT : NEXT p
3037 IF PRINT=1 THEN GO TO 3050
3040 GO SUB 1200: LPRINT : NEXT
p
3050 GO SUB 1300: GO SUB 1300: P
RINT INK 7:"END OF FILE": PAUSE
0: GO TO 1000
3500 CLS : INPUT "NAME? ";z$: IF
LEN z$>10 THEN GO TO 3500
3505 GO SUB 3900
3510 GO SUB 1300: FOR x=1 TO PRO
G: IF P$(x,1 TO LEN z$)=z$ THEN
LET P=x: GO SUB (1100 AND PRINT=
1)+(1200 AND PRINT=2): GO TO 353
0

```

```

3520 NEXT X
3525 GO TO 3050
3530 GO SUB 1300: PRINT "END OF FILE"
"END OF FILE?": PAUSE 0: IF INKEY$="
Y" THEN PRINT AT 21,0;"
": NEXT X
3540 PRINT INK 7;"END OF FILE":
PAUSE 0: GO TO 1000
3600 CLS : INPUT "TAPE # ";Z$: I
F LEN Z$>2 THEN GO TO 3800
3605 FOR Z=1 TO LEN Z$: IF Z$(Z)
<"0" OR Z$(Z)>"9" THEN GO TO 380
0
3607 NEXT Z: LET X$=(" " AND LEN
Z$=1)+Z$
3610 GO SUB 3900
3620 GO SUB 1300: FOR X=1 TO PRO
G: IF P$(X,11 TO 12)=X$ THEN LET
P=X: GO SUB (1100 AND PRINT=1)+
(1200 AND PRINT=2)
3630 NEXT X: PRINT AT 21,0; INK
7;"FINISHED": PAUSE 0: GO TO 100
0
3900 CLS : PRINT AT 21,0;"PROGRAM
": PAUSE 0
: IF INKEY$="M" OR INKEY$="P" TH
EN LET print=1+(1 AND INKEY$="P"
): PRINT AT 21,0;"": RETURN
3910 GO TO 3900
4000 REM
4001 CLS : INPUT "ALPHABETICLY
BY TAPE ";Z$
4003 IF Z$<>"1" AND Z$<>"2" THEN
GO TO 4000
4210 CLS : LET A=1: LET B=2: PRI
NT AT 10,12; INK 7;"SORTING": IF
Z$="2" THEN GO TO 4500
4220 FOR X=1 TO PROG-1
4230 FOR Y=X TO PROG
4240 IF P$(Y,1 TO 10)>=P$(X,1 TO
10) THEN GO TO 4280
4250 LET B$=P$(Y)
4260 LET P$(Y)=P$(X)
4270 LET P$(X)=B$
4280 NEXT Y
4290 NEXT X: BEEP .2,0: GO TO 10
00
4500 FOR X=1 TO PROG-1
4510 FOR Y=X TO PROG
4520 IF VAL P$(Y,11 TO 12)>=VAL
P$(X,11 TO 12) THEN GO TO 4550
4530 LET B$=P$(Y): LET P$(Y)=P$(
X): LET P$(X)=B$
4550 NEXT Y: NEXT X
4600 LET W=1: FOR Z=1 TO PROG
4610 IF P$(Z+1,11 TO 12)<>P$(Z,1
1 TO 12) THEN GO TO 4700
4620 NEXT Z
4630 BEEP .2,0: GO TO 1000
4700 FOR X=W TO Z-1
4710 FOR Y=X TO Z
4720 IF VAL P$(Y,14 TO 16)+(1000
AND VAL P$(Y,13)=2)>=VAL P$(X,1
4 TO 16)+(1000 AND VAL P$(X,13)=
2) THEN GO TO 4750
4730 LET B$=P$(Y): LET P$(Y)=P$(
X): LET P$(X)=B$
4750 NEXT Y: NEXT X: LET W=Z+1:
GO TO 4620
5000 CLS : SAVE "TAPE REC." LINE
0
5005 PRINT AT 20,0;"REWIND TAPE
AND PLAY FOR VER
IFICATION";AT 10,0

```

```

5010 ON ERR GO TO 5500
5020 VERIFY "TAPE REC."
5030 CLS : PRINT AT 21,0; INK 7;
"TAPE OK": ON ERR GO TO 1000: PA
USE 100: BEEP .2,20: GO TO 1000
5500 CLS : PRINT AT 21,0; INK 7;
"TAPE ERROR": BEEP .75,-20: PAUS
E 150: ON ERR GO TO 1000: GO TO
1000
6000 REM
6010 CLS : PRINT AT 20,0;"IF YOU
ADDED TO THIS PROGRAM... DID YO
U RESAVE IT YET?(Y OR N)"
6020 LET Z$=INKEY$: IF Z$="Y" OR
Z$="N" THEN GO TO 6040
6030 GO TO 6020
6040 IF Z$="N" THEN GO TO 1000
6050 CLS : PRINT AT 21,0;"START
TAPE THEN PRESS ANY KEY": PAUSE
0: CLS : ON ERR RESET : LOAD ""
7000 ON ERR RESET : STOP
8000 INPUT "WHICH PROGRAM? ";Z$
8010 IF LEN Z$>10 OR Z$="" THEN
GO TO 8000
8020 FOR Z=1 TO PROG: IF Z$=P$(Z
,1 TO LEN Z$) THEN GO TO 8050
8025 NEXT Z
8030 CLS : PRINT AT 10,6; INK 7;
"PROGRAM NOT FOUND": PAUSE 100:
GO TO 1000
8035 GO TO 8500
8050 CLS : LET P=Z: GO SUB 1300:
GO SUB 1100
8060 INPUT "DELETE THIS PROGRAM?
Y OR N";X$
8070 IF X$="Y" THEN GO TO 8090
8080 NEXT Z
8090 FOR Z=P TO PROG: LET P$(Z)=
P$(Z+1): NEXT Z: LET PROG=PROG-1
: GO TO 1000
8500 ON ERR GO TO 8600: DIM U$(P
ROG,16): FOR X=1 TO PROG: LET U$
(X,1 TO 16)=P$(X,1 TO 16): NEXT
X
8510 DIM P$(PROG+1,16): FOR X=1
TO PROG: LET P$(X,1 TO 16)=U$(X,
1 TO 16): NEXT X: DIM U$(1)
8520 RETURN
8600 DIM U$(1): ON ERR RESET : P
RINT AT 10,10; INK 7;"FILE FULL"
: BEEP 1,-20: PAUSE 0: GO TO 100
0
9000 SAVE "TAPE REC." LINE 15

```

LETTERS TO THE EDITOR

By the way folks, I must apologise for the error on the last issue that indicated VOL.II Issue III. Please change your copy to read VOL.II ISSUE IV. This month's is issue V.

Keep those cards and letters coming....what???

FILER

Listing 1. File Manager program for the ZX-81 and TS-1000.
Microcomputing, March 1983

```

100 REM A LISTS PROGRAM FOR THE ZX81
200 REM MAIN MENU
210 CLS
215 SLOW

220 PRINT TAB 7;"THE LISTS PROGRAM"
230 PRINT
240 PRINT "1..ADD RECORD"
250 PRINT
260 PRINT "2..CHANGE RECORD"
270 PRINT
280 PRINT "3..DELETE RECORD"
290 PRINT
300 PRINT "4..LIST ALL RECORDS"
310 PRINT
320 PRINT "5..SEARCH FOR A RECORD"
330 PRINT
340 PRINT "6..SORT RECORDS"
350 PRINT
360 PRINT "7..SAVE RECORDS TO TAPE"
370 PRINT
380 PRINT "8..SET UP NEW LIST FILE"
390 PRINT
395 PRINT "9..LIST FIELDS"
400 PRINT
410 PRINT
500 PRINT "WHICH DO YOU WISH TO DO?";

510 INPUT A
515 LET A=INT A
520 IF A > 0 AND A < 10 THEN GOTO 560
530 CLS
540 PRINT "PLEASE CHOOSE 1-9"
550 GOTO 230
560 CLS
570 GOSUB A*1000
580 GOTO 210

1000 REM ADD RECORDS
1010 LET N=N + 1
1015 SCROLL
1020 PRINT "RECORD NUMBER ";N
1030 IF N<=M THEN GOTO 1060
1032 LET N=M
1035 SCROLL
1040 PRINT "NO MORE RECORDS CAN BE ADDED"
1050 GOTO 1920
1060 FOR I=1 TO N1
1070 SCROLL
1080 PRINT N*(I);
1090 INPUT I*(N,D(I,1) TO D(I,2))
1095 PRINT I*(N,D(I,1) TO D(I,2));
1100 SCROLL
1110 NEXT I
1120 SCROLL
1130 PRINT "CHANGE ANYTHING? ";
1140 INPUT A*
1145 SCROLL
1150 IF A*(1)="Y" THEN GOTO 1060
1160 SCROLL
1170 PRINT "RECORD ";N;" ADDED"
1180 SCROLL
1900 PRINT "ADD MORE RECORDS?";
1910 INPUT A*

1920 SCROLL
1930 IF A*(1)="Y" THEN GOTO 1000
1950 RETURN

2000 REM CHANGE RECORD
2010 SCROLL
2011 SCROLL
2020 PRINT "TO CHANGE A RECORD, YOU MUST"
2030 SCROLL
2040 PRINT "ENTER THE RECORD NUMBER FOR"
2050 SCROLL
2060 PRINT "THAT RECORD. DO YOU WISH TO"
2070 SCROLL
2080 PRINT "SEARCH FOR THE RECORD NUMBER?";
2090 INPUT A*
2095 SCROLL
2100 IF A*(1)="Y" THEN GOSUB 5000
2110 CLS
2120 SCROLL
2130 PRINT "RECORD NUMBER TO CHANGE: ";
2140 INPUT A
2145 LET A=INT A
2150 SCROLL
2160 IF A>0 AND A<=N THEN GOTO 2200
2170 PRINT "INVALID RECORD NUMBER"
2180 GOTO 2900
2200 FOR I=1 TO N1
2210 SCROLL
2220 PRINT N*(I);I*(A,D(I,1) TO D(I,2))
2230 SCROLL
2240 PRINT TAB 10;"CHANGE?";
2250 INPUT A*
2255 SCROLL
2260 IF A*(1)<>"Y" THEN GOTO 2300
2270 PRINT N*(I);
2280 INPUT I*(A,D(I,1) TO D(I,2))
2285 PRINT I*(A,D(I,1) TO D(I,2))
2290 SCROLL
2300 NEXT I
2800 SCROLL
2810 PRINT "FINISHED WITH RECORD ";A
2900 SCROLL
2910 PRINT "CHANGE OTHER RECORDS?";
2920 INPUT A*
2930 SCROLL
2940 IF A*(1)="Y" THEN GOTO 2000
2950 RETURN

3000 REM DELETE A RECORD
3010 IF N>0 THEN GOTO 3060
3020 SCROLL
3030 PRINT "NO RECORDS IN FILE"
3040 GOTO 3330
3060 SCROLL
3070 PRINT "ITEMS ARE DELETED BY RECORD"
3080 SCROLL
3090 PRINT "NUMBER. RECORD NUMBERS MAY"
3100 SCROLL
3110 PRINT "CHANGE AFTER AN ITEM IS DELETED"
3120 SCROLL
3130 PRINT "DO YOU WISH TO SEARCH FOR THE"
3132 SCROLL
3135 PRINT "RECORD?";
3140 INPUT A*
3150 SCROLL
3160 IF A*(1)="Y" THEN GOSUB 5000
3170 SCROLL
3180 PRINT "RECORD NUMBER TO DELETE:";
3190 INPUT A
3195 LET A=INT A
3200 IF A>0 AND A<=N THEN GOTO 3250
3210 SCROLL
3220 PRINT "INVALID RECORD NUMBER"
3230 GOTO 3900
3250 FOR I=1 TO N1
3260 SCROLL
3270 PRINT N*(I);I*(A,D(I,1) TO D(I,2))

```

```

3280 NEXT I
3290 SCROLL
3300 PRINT "DELETE THIS RECORD?";
3310 INPUT A$
3320 IF A$(1)="Y" THEN GOTO 3360
3330 SCROLL
3340 PRINT "DELETE CANCELLED"
3350 GOTO 3900
3360 IF A=N THEN GOTO 3450

3400 FAST
3405 FOR I=A TO N-1
3410 LET I$(I)=I$(I+1)
3420 NEXT I
3430 SLOW
3450 LET N=N-1

3460 SCROLL
3470 PRINT "RECORD DELETED"
3900 SCROLL
3910 PRINT "DELETE ANY OTHER RECORDS?";
3920 INPUT A$
3930 SCROLL
3940 IF A$(1)="Y" THEN GOTO 3010
3950 RETURN

4000 REM LIST RECORDS
4010 SCROLL
4015 LET A=0
4020 PRINT "PRINT ALL RECORD FIELDS?";
4030 INPUT A$
4040 IF A$(1)="Y" THEN LET A=1
4045 SCROLL
4050 FAST
4060 FOR I=1 TO N1
4070 LET A(I)=A
4075 IF A=1 THEN LET A(I)=I
4080 NEXT I
4090 SLOW
4095 LET J=N1
4100 IF A=1 THEN GOTO 4220
4110 SCROLL
4120 PRINT "ENTER Y FOR EACH FIELD"
4121 SCROLL
4123 PRINT "YOU WANT PRINTED:"
4125 LET J=0
4130 SCROLL
4140 FOR I=1 TO N1
4150 SCROLL
4160 PRINT N$(I); " ?";
4170 INPUT A$
4175 PRINT A$(1)
4180 IF A$(1)<>"Y" THEN GOTO 4210
4190 LET J=J+1
4200 LET A(J)=I
4210 NEXT I
4220 SCROLL
4230 PRINT "PRINT TO SCREEN OR PRINTER(S/P)"
4240 INPUT A$
4250 IF A$(1)="P" THEN GOTO 4600
4300 SCROLL
4305 SCROLL
4310 PRINT TAB 5;L$;" LIST"
4320 SCROLL
4330 FOR I=1 TO N
4340 FOR K=1 TO J
4350 SCROLL
4360 PRINT N$(A(K));I$(I,D(A(K),1) TO D(A(K),2))
4370 NEXT K
4380 SCROLL
4390 NEXT I
4400 GOTO 4900
4600 FAST
4610 LPRINT TAB 5;L$;" LIST"
4620 LPRINT

4630 FOR I=1 TO N
4640 FOR K=1 TO J
4650 LPRINT
4660 LPRINT N$(A(K));I$(I,D(A(K),1) TO D(A(K),2))
4670 NEXT K
4680 LPRINT
4690 NEXT I
4700 SLOW
4900 SCROLL
4910 PRINT "ANOTHER LIST?";
4920 INPUT A$
4930 IF A$(1)="Y" THEN GOTO 4000
4950 RETURN

5000 REM SEARCH FOR RECORD
5010 SCROLL
5020 PRINT "WHICH FIELD TO SEARCH ON:"
5030 FOR I=1 TO N1
5040 SCROLL
5050 PRINT N$(I); " ?";
5060 INPUT A$
5070 IF A$(1)="Y" THEN GOTO 5100
5080 NEXT I
5085 SCROLL
5090 PRINT "SELECTION CANCELLED"
5095 GOTO 5900
5100 SCROLL
5110 PRINT "ENTER THE SEARCH STRING";
5120 INPUT A$
5122 LET A=LEN A$
5125 IF A<1 THEN GOTO 5110
5126 IF A<=(D(I,2)-D(I,1)+1) THEN GOTO 5129
5127 LET A=(D(I,2)-D(I,1)+1)
5128 LET A$=A$(1 TO A)
5129 LET A=A-1+D(I,1)
5130 LET F=0
5140 LET Q=0
5150 FAST
5160 FOR J=1 TO N
5170 IF A$=I$(J,D(I,1) TO A) THEN GOSUB 5500
5180 IF Q=1 THEN GOTO 5200
5190 NEXT J
5200 SLOW
5210 SCROLL
5220 PRINT "SEARCH COMPLETE"
5230 SCROLL
5240 IF F=0 THEN PRINT "RECORD NOT FOUND"
5250 GOTO 5900
5500 SCROLL
5504 SCROLL
5505 PRINT "RECORD NUMBER ";J
5510 FOR K=1 TO N1
5520 SCROLL
5530 PRINT N$(K);I$(J,D(K,1) TO D(K,2))
5540 NEXT K
5550 SCROLL
5560 PRINT "CONTINUE SEARCH?";
5570 INPUT B$
5580 IF B$(1)="N" THEN LET Q=1
5590 LET F=1
5600 RETURN
5900 SCROLL
5910 PRINT "ANOTHER SEARCH?";
5920 INPUT A$
5930 IF A$(1)="Y" THEN GOTO 5000
5950 RETURN

6000 REM BUBBLE SORT RECORDS
6010 SCROLL
6020 PRINT "SORTING TIME DEPENDS ON "
6021 SCROLL
6025 PRINT "NUMBER OF RECORDS"
6030 SCROLL
6040 PRINT "WHICH FIELD TO SORT BY:"
6050 SCROLL
6060 FOR I=1 TO N1
6070 SCROLL

```



```

6080 PRINT N$(I); " ?";
6090 INPUT A$
6100 IF A$(1)="Y" THEN GOTO 6150
6110 NEXT I
6120 SCROLL
6130 PRINT "SORT CANCELLED"
6140 GOTO 6900
6150 SCROLL
6160 FAST
6170 FOR J=1 TO N-1
6180 FOR K=J TO N
6190 IF I$(J,D(I,1) TO D(I,2)) <= I$(K,D(I,1) TO D(I,2))
THEN GOTO 6250
6200 LET B$=I$(J)
6210 LET I$(J)=I$(K)
6220 LET I$(K)=B$
6250 NEXT K
6260 NEXT J
6270 SLOW
6280 SCROLL
6290 PRINT "SORT COMPLETE"
6900 SCROLL
6910 PRINT "SORT ON ANOTHER FIELD?";
6920 INPUT A$
6925 SCROLL
6930 IF A$(1)="Y" THEN GOTO 6000
6950 RETURN

```

```

7000 REM SAVE PROGRAM AND DATA TO TAPE
7010 SCROLL
7020 PRINT "PUT CASSETTE IN TAPE RECORDER"
7030 SCROLL
7040 PRINT "BEGIN RECORDING, THEN"
7042 SCROLL
7045 PRINT "PRESS ANY KEY TO SAVE LIST";
7050 IF INKEY$="" THEN GOTO 7050
7060 SCROLL
7070 SAVE L$
7900 GOTO 100

```

```

8000 REM SET UP NEW LIST
8010 SCROLL
8020 PRINT "WHAT LIST NAME TO USE?";
8030 INPUT L$
8040 SCROLL
8050 PRINT "HOW MANY RECORD FIELDS?";
8060 INPUT N1
8061 PRINT N1
8070 SCROLL
8080 LET K=15
8090 PRINT "MAXIMUM NUMBER OF CHARACTERS IN"
8100 SCROLL
8110 PRINT "DESCRIPTION IS (0-15)?";
8120 INPUT K
8130 IF K>15 THEN LET K=15
8135 IF K< 0 THEN LET K=0
8140 SCROLL
8150 LET L=31-K
8160 PRINT "MAXIMUM SIZE OF DATA FIELD"
8170 SCROLL
8180 PRINT "IS ";L;" CHARACTERS"
8182 DIM N$(N1,K)
8184 DIM A(N1)
8186 DIM D(N1,2)
8190 SCROLL
8195 SCROLL
8200 PRINT "NOW ENTER THE FIELD DESCRIPTIONS"
8210 SCROLL
8220 PRINT "AND FIELD LENGTHS:";
8225 LET J=1
8230 SCROLL
8240 FOR I=1 TO N1
8250 SCROLL
8255 LET A$=""
8260 PRINT "FIELD ";I;" :";
8270 INPUT A$

```

```

8272 PRINT A$
8275 LET A$=A$+" "
8280 LET N$(I)=A$
8290 SCROLL
8300 PRINT "FIELD LENGTH (1-";L;" ):";
8310 INPUT A
8312 IF A<1 THEN LET A=1
8315 IF A>L THEN LET A=L
8317 PRINT A
8320 LET D(I,1)=J
8330 LET J=J+A
8335 LET D(I,2)=J-1
8340 SCROLL
8350 NEXT I
8360 LET J=J-1
8370 LET N=0
8380 LET M=INT ((8000-N1*(31-L+12))/J)
8390 SCROLL
8400 PRINT "MAXIMUM NUMBER OF RECORDS"
8410 SCROLL
8420 PRINT "POSSIBLE IS ABOUT ";M
8430 SCROLL
8435 PRINT "HOW MANY RECORDS DO YOU WANT?";
8437 INPUT A
8439 LET A=INT A
8440 IF A>0 AND A<M THEN LET M=A
8450 DIM I$(M,J)
8900 SCROLL
8910 PRINT L;" LIST SET UP"
8920 SCROLL
8930 PRINT "PRESS ANY KEY TO RETURN TO MENU"
8940 IF INKEY$="" THEN GOTO 8940
8950 RETURN

```

```

9000 REM LIST FIELDS
9020 SCROLL
9030 PRINT TAB 5;L;" LIST FIELDS"
9040 SCROLL
9050 FOR I=1 TO N1
9060 SCROLL
9065 SCROLL
9070 PRINT N$(I)
9080 NEXT I
9090 SCROLL
9100 PRINT "PRESS ANY KEY TO RETURN TO MENU";
9105 SCROLL
9110 IF INKEY$="" THEN GOTO 9110
9900 RETURN

```

The preceding program is a great little file manager program for the ZX81 or TS1000. Next News letter, we will include some refinements and a few hints on adapting it for the 2068.

(By the way, there is a copy of this program in our tape library in case you don't feel like typing it all in. Just borrow the tape from David Solly and copy it)

