



Vol. 3 No. 5
MAY 1985



THE
SPRITE

a monthly newsletter of
THE 9900 USER'S GROUP, INC.

A voluntary organization for the sharing of knowledge and resources of people having interests in, or ownership of 9900 processor based Home Computers.

The 9900 Users Group - May 1985

longer was I tied to a 254 byte record, it could be as long as I wished, (about 22k max due to file limitations imposed by the program.) Data base files were finally mine. I was bursting at the seams. I had to tell someone. Gary, he would appreciate my discovery.

That's great he said, but if you filed all the names in groups, like all the A's, the B's and C's together, then you could use FindString to locate a first name, if you didn't remember the last name. Or you could use it to find a street name, zip code or anything else you wanted to use to locate someone by and you wouldn't even have to sort each time you wanted to change the base of the item you were looking for. No fair!, I said, you've been working with TI-WRITER also.

TI-WRITER TUTORIAL Part 2: by Michael Kelly

Last month (Ed: March issue) the letter below was used to introduce the Text Editor of TI-Writer. The same letter will be used this month to demonstrate other procedures used in the Text Editor. Load the letter from last month or type it and begin.

A. MOVE LINES

1. LOAD FILE
-TYPE LF (LOAD FILE) THEN PRESS ENTER.
-TYPE DSK1.LETTERA (FILE NAME) THEN PRESS ENTER.
2. PRESS FCTN 9 TO GET INTO COMMAND MODE.
3. TYPE M (Move) THEN PRESS ENTER.
4. TYPE LINE NUMBER OF FIRST LINE TO BE MOVED.
5. TYPE COMMA OR SPACE BAR.
6. TYPE LINE NUMBER OF LAST LINE TO BE MOVED.
7. TYPE COMMA OR SPACE BAR.
8. TYPE NUMBER OF LINE BEFORE LINE WHERE MOVED TEXT WILL START.
-E.G. 45,49,27
9. MOVE LINES AS SHOWN IN THE LETTER BELOW.

```
0001          Street
0002          City, State Zip
0003          Date
0004
0005
0006 Dear Joe,
0007
0008 This past week I won $9,000,000.06 playing bingo
0009 at church.
0010
0011 How are you? Hope all is well with you.
0012
0013 I have no idea what to do with the money. If you
0014 can help me spend some of the money, give me a call
0015 at (111)BIG-BUCK.
0016
```

0017 Thank you for reading this urgent call for help.

0018

0019

Gratefully yours,

0020

0021

0022

0023

0024

Name

10. PRINT FILE. (You know now by now.)

11. TO MOVE LINES AS SHOWN ABOVE YOU SHOULD HAVE ENTERED (M) THEN (8,179,182) OR (10,12,7).

B. COPY LINES

1. LOAD FILE DSK1.LETTERA.
2. PRESS FCTN 9 TO ENTER COMMAND MODE.
3. TYPE C (COPY) THEN PRESS ENTER.
4. TYPE NUMBER OF LINE WHERE COPYING WILL BEGIN.
5. TYPE COMMA OR SPACE BAR.
6. TYPE NUMBER OF LINE WHERE COPYING WILL END.
7. TYPE COMMA OR SPACE BAR.
8. TYPE NUMBER OF LINE BEFORE DUPLICATION WILL BEGIN.
-E.G. 11,16,24
9. COPY LINES AS SHOWN IN THE LETTER BELOW.

0001

Street

0002

City, State Zip

0003

Date

0004

0005

0006 Dear Joe,

0007

0008

How are you? Hope all is well with you.

0008

How are you? Hope all is well with you.

0009

0010

This past week I won \$9,000,000.06 playing bingo

0011

at church.

0012

0013

I have no idea what to do with the money. If you

0014

can help me spend some of the money, give me a call

0015

at (111)BIG-BUCK.

0016

0017

Thank you for reading this urgent call for help.

0018

0013

I have no idea what to do with the money. If you

0014

can help me spend some of the money, give me a call

0015

at (111)BIG-BUCK.

0016

Gratefully yours,

0019

0020

0021

0022

0023

0024

Name

formidable task but is greatly simplified by the use of subroutines.

INTRODUCTION:

Time to hustle here. Have to get this newsletter out or this one will be behind too! It's not so much in getting it written but once it's done getting it copied, sorted and mailed. That takes almost as much time as writing it. Later on I'll have a short blurb on the Computer Show & Flea Market over in Pennsylvania this past 18th of May. Watch for the next show up at the Meadowlands on 1 June.

***** NOTICE *****

The next general meeting originally scheduled for Wednesday, 29 May has been rescheduled by Marlton Middle School and will be ONE DAY EARLIER. The NEW date is Tuesday, 28 May 1985. Sorry for any inconvenience. Also, as of this writing, there will be NO 'combined' Splinter Group meeting in June. A general meeting place for June has not yet been firmed up.

Summer certainly seems like it has arrived. I know that I've had a recent bad case of spring fever. I mean you really don't want to do anything! However, spring fever leads to tall grass and more work.

Before I forget an apology is in order to those who have been looking for the Extended Basic Splinter Group. I was at the 'combined' meeting on the 13th and two members were there looking for their Splinter Group Leader. Well, I forgot to mention at the last meeting that the 'next' splinter meeting was to be canceled. That was the meeting scheduled for Doc Jacobs house I believe. The 'combined' monthly splinter meeting still meets once a month and if one part is cancelled the individual attendees (those who normally attend) should be contacted. That was not done. The apology is for that. Ray has been working evenings and his time is all gone. He has already declined from the position of secretary due to his work restrictions. I don't think Ray would be overly offended if someone volunteers to assume the Extended Basic Splinter Group leadership. If someone out there would like to do this just contact the group and arrangements and other guidelines can be worked out.

This month we have the continuing tutorial on TI-Writer and the final edition of the spreadsheet we had a couple issues ago. This edition of that spreadsheet is the program 'in-total.' There were so many changes and mods that the final article is almost a rewrite so the whole program is included for completeness. It is an excellent example of subroutine usage to accomplish what at first looks like a

Received the latest newsletter from CorComp and they explain in pretty good detail all the problems they've had over the last few years. A very good job. The newsletter is in or library. They also have a good article on p-code double density initialization. If you attempt to use the normal method using DFORMAT everything 'looks' good except that the disk is really still single density. Cure: Format the disks using the CorComp or TI disk controller. (Ed Note: Why the TI controller I have no idea???) Then going into the p-system call up the FILER and use the ZERO command to 'zero' the directory. You are ready to go!

TI-WRITER(Section1): by Marshall via A9CUG CALL Newsletter

This is a very powerful word processing program. As an example look at what happened to me while I was working on this last night. I had just finished a long article but I had been a naughty boy. Caught up in what I was writing I did not save to disk from time to time as I should have. Just as I was about to save the completed first draft -- a power glitch hit, the keyboard locked up, the screen turned into psychedelic checkerboard fireworks and finally went blank.

After a few calm curse words I remembered a new word I had seen recently, "RecoverEdit." The file was still in the memory expansion and if I could get to it I could get the work back. I turned off the console, (NEVER turn off the peripheral expansion under these conditions), waited an interminable two minutes and turned the console back on. Selected TI-WRITER from the menu, and then "1" for edit. Then the moment of truth! Was TI-WRITER up to the strain of the big moment? In command mode I entered "RE" then pressed "ENTER" and low and behold there was my work back on the screen again. Except for the first line, which was easily replaced I then did what I should have while typing. I "saved" the text to disk. TI-WRITER may not be the most fancy program out, and the P-box may be big and noisy but they can really do many things that NO other computer can do.

TI-WRITER(Section 2-Data Base Files): by Marshall via A9CUG

While working with TI WRITER I discovered that the disk controller automatically alphabetized all the files. I needed a data base that was larger than the programs available allowed. Here I thought could be the answer to my problem. I set up a form that I wanted to use, and proceeded to fill in several records. Saving them to disk and then all I had to do was ask for a ShowDirectory and I could see my alphabetized file, no matter what order I typed them in. No

10. PRINT FILE.
11. TO COPY THE FIRST LINE YOU SHOULD HAVE ENTERED (C) THEN (8,8,8).
12. TO COPY THE SECOND SET OF LINES YOU SHOULD HAVE ENTERED (C) THEN (13,16,18).

0024

Name

C. FIND AND REPLACE

Find String

1. LOAD FILE **DSK1.LETTERA**.
2. PRESS **FCTN 9** TO ENTER COMMAND MODE.
3. TYPE **SH** FOR Search THEN PRESS **ENTER**.
4. TYPE **FS** FOR Find String THEN PRESS **ENTER**.
5. TYPE **/\$.00/** TO FIND THE FIRST TIME \$5.00 IS PRINTED IN THE TEXT THEN PRESS **ENTER**.

Replace String

1. PRESS **FCTN 9** TO ENTER COMMAND MODE.
2. TYPE **SH** FOR Search THEN PRESS **ENTER**.
3. TYPE **RS** FOR Replace String THEN PRESS **ENTER**.
4. TYPE **/TI-Writer/TI/WRITER/** THEN PRESS **ENTER**.
5. CHOOSE DESIRED OPTION.
 - A FOR All -REPLACES THE STRING EVERY TIME IT APPEARS IN THE TEXT.
 - Y FOR Yes -REPLACES THE STRING IN SHOWN LOCATION THEN MOVES CURSOR TO NEXT STRING LOCATION. HERE ANOTHER DECISION WILL BE MADE.
 - N FOR No -DOES NOT REPLACE THE STRING IN THE SHOWN LOCATION THEN MOVES CURSOR TO THE NEXT STRING LOCATION
 - S FOR Stop -LEAVES REPLACE STRING.
6. REPLACE **money** with **winnings** as shown below.

```

0001                Street
0002                City, State Zip
0003                Date
0004
0005
0006 Dear Joe,
0007
0008 How are you? Hope all is well with you.
0009
0010 This past week I won $9,000,000.06 playing bingo at
0011 church.
0012
0013 I have no idea what to do with the winnings. If
0014 you can help me spend some of the winnings, give me
0015 a call at (111)B16-BUCK.
0016
0017 Thank you for reading this urgent call for help.
0018
0019                Gratefully yours,
0020
0021
0022
0023
    
```

7. PRINT FILE.
8. SAVE FILE (**DSK1.LETTERA**
9. TO REPLACE STRING YOU SHOULD HAVE ENTERED (SH), (RS), (/money/winnings/) THEN (A).

Next month a new letter will be used.

COMPUTER SHOWS & FLEAMARKETS:

If you missed the show on 18 May over in PA. you have another chance on 1 June up at the Meadowlands Hilton near Syosset, NJ. The show over in PA was well worth the trip IF you had items in mind to pick up. The bargains were there! There were a few booths selling many TI items. This was a smaller show than what you will find at the meadowlands next month. The meadowlands will have TWO areas of selling. Each one is more than TWICE that of the PA show. One area is strictly vendors for current new items etc and the other area is a huge flea market which is under cover in case it rains. Oh yea. Don't forget cash for the flea market. The vendors will take a check and some a credit card. Have fun!

LIBRARY NOTES:

I always talk in whispers when I talk about our library. However I am assured that it is taking shape, there may even be a list at the next meeting. That would be nice. All that aside. If any of the members wish to call any of the librarians to find out what we have and even get it don't hesitate to call them. Even though there is no list published as yet the stuff is there and available.

MEETING DATES:

We are again in the process of moving to our 'summer quarters.' We will advise everyone what the summer schedule will be as soon as we know, we hope.

MEETING AGENDA:

- 7:00PM - 7:15PM Introduction, what's new, etc.
- 7:15PM - 8:00PM Subject of interst. (GRAPHX???)
- 8:00PM - 8:15PM Giveaways & Doorprizes. Some 'different' books this time!
- 8:15PM - 8:50PM General browsing, buy, trade, etc.
- 9:00PM Close meeting. (Last nite at MARLTON)

** BASIC SPLINTER MEETING *
 NOTE: → 3 JUNE @ McDonalds Meeting Rm
 Clements Bridge Rd & Rt 41*

TIPS FROM THE TIGERCUB

021

Copyright 1985

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

The entire contents of Tips from the Tigercub Nos. 1 through 14, with more added, are now available as a full disk of 50 programs, routines and files for just \$15.00 postpaid!

Nuts & Bolts is a diskfull of 100 (that's right, 100!) XBasic utility subprograms in MERGE format, ready for you to merge into your own programs. Contents include 13 type fonts, 14 text display routines, 12 sorts and shuffles, 9 data saving and reading routines, 9 wipes, 8 pauses, 6 music, 2 protection, etc., and now also a tutorial on using subprograms, all for just \$19.95 postpaid!

And I have about 140 other absolutely original programs in Basic and XBasic at only \$3.00 each! (plus \$1.50 per order for cassette, packing and postage, or \$3.00 for diskette, PPM) Some users groups charge their members that much for public domain programs! I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

I thought that my 28-Column Converter, as published in Tips #18, was

finally foolproof, but someone found a way to print a program incorrectly with it!

I'm sure you know that characters 127-143, and on up to 159 in Basic, can be redefined and used in graphics. You probably also know that these redefined characters can be put into PRINT or DISPLAY AT statements, by holding down the CTRL key as you type them. If you load a program containing such redefined characters and LIST it, they will appear as blanks. If you RUN the program, so that they are redefined by the CALL CHAR statements, and then LIST it again, they will show up in their redefined form - but if you print out the program on your printer, they will still appear as blanks. So, before you publish a program, it's a good idea to RUN it and LIST it, and look for any of those gremlins.

If you do want to publish such a program, this fix will take care of it by underlining all characters that must be typed with CTRL down (except that lower case v is typed with FCTN down). It's slow, so only use it when you need to.

```
190 IF W$="E" THEN 195 :: PR
INT #2:".TL 126:94;" :: PRIM
T #2:".TL 123:64;" :: PRINT
#2:".TL 125:38;" :: PRINT #2
:".TL 124:42;" :: PRINT #2:"
.TL 92:46;" :: PRINT #2:".NF
"
195 PRINT "Does the program
contain": "redefined characte
rs above": "ASCII 126? (Y/N)"
196 ACCEPT AT(24,1)VALIDATE(
"Y/N"):QQ$
282 IF QQ$="N" THEN 290
283 FOR J=1 TO LEN(L$)
284 A=ASC(SE6$(L$,J,1)):: IF
A<127 THEN L2$=L2$&CHR$(A):
: GOTO 288
285 IF A=127 THEN A=118 ELSE
IF A=128 THEN A=44 ELSE IF
```

```
A=155 THEN A=46 ELSE IF A=15
6 THEN A=59 ELSE IF A=157 TH
EN A=61 ELSE IF A=158 THEN A
=56 ELSE IF A=159 THEN A=57
ELSE A=A-64
286 L2$=L2$&CHR$(27)&CHR$(45
)&CHR$(1)&CHR$(A)&CHR$(27)&C
HR$(45)&CHR$(0)
288 NEXT J :: L$=L2$ :: L2$=
""
```

That should do it, unless the number of added control characters stretches the line beyond 80 characters. Such is the case with the following, which I had to type in manually (it also contains low ASCII characters which the printer misinterprets as controls).

TIGERCUB CHALLENGE

```
100!The Unprintable Unkeyabl
e Program!
110!To shuffle the numbers 1
to 255 into a random sequen
ce without duplication
120!The strings contain the
ASCII characters 1 to 127 an
d 128 to 255
130!Most of the ASCII charac
ters below 32 or above 159 c
annot be input from the keyb
oard
140!So how was this program
programmed?
150 M$="
!"#$%&'()*+,-./0
123456789:;<=>@ABCDEFGHIJKLM
NOPQRSTUVWXYZ[\]^_`abcdefg
hijklmnopqrstuvwxyz{|}"
160 M2$="
```

```
170 M$=M$&M2$
180 L=LEN(M$):: RANDOMIZE ::
X=INT(L&RND+1):: M=ASC(SE6$
(M$,X,1)):: M$=SE6$(M$,1,X-1
)&SE6$(M$,X+1,LEN(M$))
190 PRINT M$:: IF LEN(M$)=0
THEN STOP ELSE 130
```

GROCERY SHOPPING LIST

Are you desperate for some way to convince your wife that your computer and PEB and printer and all are not just a too-expensive plaything? Maybe this will do the job.

The first thing to do is to prepare a file of the grocery items she might want to buy. It will be especially useful if you can list the items in the sequence in which she will come to them in the aisles of her favorite store. This little program will set up the file. Type END when you are finished.

```
100 OPEN #1:"DSK1.BUYLIST",O
UTPUT
110 INPUT A$
120 IF A$="END" THEN 150
130 PRINT #1:A$
140 GOTO 110
150 CLOSE #1
```

If you have TI-Writer, you can also use that to create the file, edit it and add to it - but BE SURE to delete all the carriage return symbols and any blank lines at the end. Save it under the filename BUYLIST.

Next, this program will hopefully get your wife to actually sit down at the keyboard and try out your computer. It will go through the list and ask if she wants to buy. If she types in any quantity other than 0, it will output the item name and quantity to the printer. At the end, she will be given the opportunity to add any other items.

```
100 CALL CLEAR
110 OPEN #1:"DSK1.BUYLIST",I
NPUT
120 OPEN #2:"PIO"
130 LINPUT #1:A$
140 IF EOF(1)THEN 210
```

```

150 DISPLAY AT(12,1):A$
160 DISPLAY AT(12,LEN(A$)+2)
170 ACCEPT AT(12,LEN(A$)+2)S
180 IF Q=0 THEN 130
190 PRINT @2:A$&" "&STR$(Q)&
" "&CHR$(175)
200 GOTO 130
210 DISPLAY AT(12,1):"ADDITI
ONAL? Y"
220 ACCEPT AT(12,13)VALIDATE
("Y")SIZE(-1):Q$
230 IF Q$="N" THEN 300
240 DISPLAY AT(12,1):"ITEM?"
250 ACCEPT AT(12,7):A$
260 DISPLAY AT(14,1):"QUANTI
TY?"
270 ACCEPT AT(14,11):Q
280 PRINT @2:A$&" "&STR$(Q)&
" "&CHR$(175)
290 GOTO 210
300 CLOSE #1
310 CLOSE #2
320 END

```

The list will be in enlarged print, so that no one in the store will see her putting on her reading spectacles. And after each item and quantity is a blank square to be checked off when she picks up the item.

You might also point out that she could use the checkoff blocks to mark the items she has coupons for, and she could jot down prices on it to be sure she isn't cheated at the checkout counter, or to shop for better bargains elsewhere.

The program is set up for the Gemini printer. You may need to change the "PIO" to the name of your printer, and other printers may not have the open block character CHR\$(175) available.

Of course, you can also use this program for more important things, such as shopping for computer software....!

If you type the period key while holding down the

CTRL key, the printer interprets the resulting blank space as CHR\$(27), even though the computer knows it is really CHR\$(155). Since CHR\$(27) is the ESC or "escape code" which tells the printer to interpret the following characters as function command codes, you can for instance set up the printer for emphasized double-struck double-width underlined italics by OPEN #1:"PIO" ;; PRINT #1:" E 6 W"&CHR\$(1)&" -"&CHR\$(1)&" 4 ", using CTRL . in the blanks. I have been overlooking another very useful feature, the skip-over perforation. PRINT #1:" N"&CHR\$(6), again with CTRL . in the blank, causes the paper to advance to the top of the next page when there are only 6 lines left at the bottom of the page (providing that you started at the top, of course). This makes it possible to LIST "PIO" a program, or PF PIO from TI-Writer Editor, without printing right across the perforations.

Ghosts! Did you ever read data from a file, and find that you were getting data from a file that was no longer on the disk? It can happen, at least if you are reading from a RELATIVE file in the UPDATE mode. When you delete a file, only its address is actually deleted - the data remains on the disk until it is overwritten by a new file. If the new file is shorter than the old one, and you try to read beyond the end of the file, you may awaken the ghost!

Are you making use of those special characters that are available on your Gemini printer? You didn't know about them? Try this.

```

100 OPEN #1:"PIO" ;; 110
PRINT #1:" (hold down the
CTRL key and type 1234567/
and then hold down the FCTN
key and type <>/0;BHJKLNMQY
) ". RUN . Surprised? Some
of those can be very
useful, such as the true
division sign that you get
with FCTN H. There are many
more of these that you can
access by CHR$. For a
complete list of them and
their CHR$ codes, run this -
100 OPEN #1:"PIO" ;; FOR
CH=160 TO 254 ;; PRINT
#1:CHR$(CH);; NEXT CH ;;
CLOSE #1. Unfortunately,
these can't be used out of
TI-Writer.

```

Here's a handy little routine to practice up on your typing.

```

100 CALL CLEAR
110 CALL CHAR(94,"3C4299A1A1
99423C")
120 CALL SCREEN(5)
130 CALL VCHAR(1,31,1,96)
140 CALL COLOR(1,8,16)
150 FOR SET=2 TO 12
160 CALL COLOR(SET,2,16)
170 NEXT SET
180 PRINT TAB(10);"TIGERCUB"
; TAB(8);"TOUCH-TYPING"; TAB
AB(11);"TUTOR"; TAB(9);" T
iger Cub Software";
190 REM by Jim Peterson
200 PRINT " Watch the screen,
not the"; keyboard!"; "
Letters and numbers will"
210 PRINT " appear on the sc
reen grid"; " in position cor
responding"; " to their keybo
ard position."; " Type the
m and they will"
220 PRINT " disappear."; " :
" Press any key"
230 CALL KEY(0,K,ST)
240 IF ST=0 THEN 230
250 CALL CLEAR
260 CALL CHAR(32,"FFB0B0B0B0
B0B0B")
270 CALL VCHAR(1,30,1,192)
280 CALL MCHAR(14,1,1,384)
290 CALL VCHAR(1,4,1,14);; C
ALL VCHAR(5,6,1,11);; CALL V
CHAR(8,7,1,6);; CALL VCHAR(1
1,8,1,3);; CALL VCHAR(8,29,1

```

```

,6)
300 CALL VCHAR(11,28,1,3)
310 CALL CHAR(48,"003A444C54
64448B")
320 KEYS="1234567890=QWERTYU
IOP/ASDFGHJKL; "&CHR$(13)&"ZX
CVBNM,."
330 RANDOMIZE
340 K=ASC(SEEK(KEYS,INT(428R
ND*1),1))
350 GOSUB 370
360 GOTO 420
370 X=POS(KEYS,CHR$(K),1)
380 Y=ABS(X>11)+ABS(X>22)+AB
S(X>33)+1
390 R=Y*3
400 C=((X-ABS(Y)>1)+Y-1)*11)
82)+4+Y
410 RETURN
420 CALL MCHAR(R,C,M)
430 CALL KEY(3,K,ST)
440 IF ST=0 THEN 430
450 GOSUB 370
460 CALL MCHAR(R,C,6)
470 IF C<32 THEN 500
480 CALL SOUND(-100,110,0,-4
,0)
490 GOTO 340
500 CALL MCHAR(R,C,32)
510 CALL SOUND(-100,1000,0,1
005,0)
520 GOTO 340

```

Here's one for the kids to have fun with. I'm sorry I lost track of who published it.

```

100 CALL INIT ;; FOR J=1 TO
100 ;; PRINT J ;; FOR P=1000
TO 1 STEP -J ;; CALL LOAD(-
31456,P);; NEXT P ;; NEXT J

```

MEMORY FULL,
Jim Peterson

A SPREADSHEET PROGRAM
By Bob Delpit

PART III: SUPPORTING SUBROUTINES

For those of you who have been waiting for Part III, I'm sorry that I missed last month's publication deadline. However, the delay offered me an opportunity to add some improvements and remove some bugs. Since the changes have been rather extensive, I am including the entire program with this issue, with changed lines preceded by an asterisk. The changes apply only to lines 100-1000 and 3000-5000. All the other lines are new. Briefly, the new features added are as follows:

A "Single Pair" operation has been added to the CALCULATE mode to enable you to add, subtract, multiply, or divide non-consecutive rows or columns. For example, if you want to add row 2 to row 5, you type "Y" for this option. If, however, you want to add rows 2 through 5, then you enter "N" (or just hit the ENTER key) when you have to make the choice.

The ENTER/EDIT DATA mode has been revised to allow the Data Entry Marker (DEM) to be moved to the extreme left or top so that the row names and column headings can be changed while entering data. These modifications can also be used to add new rows or columns to the matrix, provided that you allowed for this when you entered the no. of rows and columns at the start of the program.

A change which you may want to make is to open up the array dimensions in line 150. However, if you are limited to 16K, I recommend that you do not go much beyond the 20x13 matrix size. The total size of the matrix is limited by the available memory, but in addition, the no. of

columns is limited to what will fit across the width of a page. As the program is now written the limit is 13 columns, and some program lines would require change to add more.

Now for a brief description of the new program lines.

1000-1170: This subroutine allows you to see what is on the disk and to let you know how much space is left on the disk. This can easily prevent you from overwriting a file, or from creating an error condition because the disk is full. The routine was taken from the DISK MEMORY SYSTEM manual, page 43 and is a very useful feature to have. Of course it has no meaning for cassette operation, so if that is your case, delete lines 1010-1170 and type "RETURN" for line 1000. Even with a disk system you may want to do this to obtain additional free memory.

2000-2090: This subroutine provides the means to load data files into the computer for further editing or for printing. If you are using a cassette, delete line 2010 and alter line 2030 to read:

```
2030 OPEN #1:"CS1",INTERNAL,INPUT ,
FIXED :: OF1=OF1+1
```

File no. and disk no. may be changed to suit your arrangement.

5000-5080: This subroutine is for writing data files to disk. Again, if you are using a cassette recorder change line 5010 to read as follows:

```
5010 OPEN #1:"CS1",INTERNAL,OUTPUT ,
FIXED :: OF1=OF1+1
```


6000-6300 This subroutine is used to send data files to the printer. The OPEN statement, line 6010, is for an Axiom parallel interface. If yours is different, then this line should be changed to meet your equipment requirements. For example, an RS232 interface with an Epson printer will look like this:

```
6010 OPEN #2:"RS232.BA=xxxx.DA=8",
VARIABLE 132 :: OF2=OF2+1
```

with the x's specifying the baud rate to match the printer setting. It is important to correctly specify the 132 column statement so that the automatic carriage return at 80 columns will be suppressed. Otherwise, you will not be able to print more than 8 data columns across the width of a page. Line 6040 automatically switches the printer to compressed print if your file has more than 8 columns. Most printers use CHR\$(15) to do this, but if you have doubts check your printer manual and change line 6040 if required. Likewise, you should check to see that CHR\$(18) in line 6120 is the correct code to turn off compressed print and reset the print to normal size.

7000-7020: This short subroutine is used to clear the data from memory so that new data can be entered. Before the data is actually erased, however, you will be reminded to save the data before you take that final step.

9000-9020: I call this subroutine the "second chance" option. This subroutine is called up after each selection from the main menu and allows you to return to the main menu in case you made the wrong selection.

9500-9550: This is the subroutine which stops the program from crashing when an error is encountered. Two of the most common types of er-

ror are (1) trying to open a file that has not been closed (and vice versa), and (2) making an illegal entry or one that cannot be handled by the computer. The possibility of making an illegal entry has been reduced by having the ACCEPT statements accept only certain types of data. However, in the CALCULATE mode you could easily set up a condition where the computer would be trying to divide by zero. Without the error subroutine, this would crash the program. The error subroutine also closes files when there is an imbalance between the OPEN and CLOSE executions.

The program as written will accept only numeric data, except for the row and column names. Acceptable numeric data are: digits 0-9, ".", ",", "+", "-", and "E". (See page 47 in the Extended Basic manual for more detail). It is not necessary to align decimals when entering data, but you cannot have more than one decimal in each data cell. Signs "+" and "-" must precede digits.

I have removed all the bugs that I have found, but having just completed the program, I have not had time to thoroughly test it. Anyone out there in 99'er land who finds a problem (and better yet, a solution), please let me know about it and I'll publish your solution in these pages with full credit to its author. The same goes for suggested improvements that would seem to have a universal interest or application. For those of you who are out-of-towners, write to me care of the newsletter.

In addition to working on some enhancements, I am planning to write a brief operating procedure for the user. I am not able now to say when this will be available or how to obtain it, but I will put details in the next issue.

The 9900 User's Group

```

100 REM***SPREADSHEET PROGRAM***BY BOB DELPIT***JAN.30,1985
*150 DIM A$(20,13),A(20,13),B$(4),B(11)
200 CALL CLEAR :: CALL SCREEN(6):: PRINT TAB(10);"MENU": : :TAB(6);"1.INSPECT DI
SK":TAB(6);"2.LOAD DATA":TAB(6);"3.ENTER/EDIT DATA"
210 PRINT TAB(6);"4.CALCULATE MODE":TAB(6);"5.SAVE DATA":TAB(6);"6.PRINT DATA":T
AB(6);"7.CLEAR MEMORY"
*220 PRINT : : : : INPUT "CHOOSE OPTION:":D :: ON ERROR 9500
230 IF D<1 OR D>8 THEN 200
240 ON D GOSUB 1000,2000,2990,4000,5000,6000,7000
250 GOTO 200
1000 CALL CLEAR :: TYPE$(1)="DIS/FIX" :: TYPE$(2)="DIS/VAR" :: TYPE$(3)="INT/FIX
" :: TYPE$(4)="INT/VAR" :: TYPE$(5)="PROGRAM"
1010 INPUT "MASTER DISK":MD :: MD=INT(MD)
1020 OPEN #1:"DSK"&STR$(MD)&".",INPUT ,RELATIVE,INTERNAL :: INPUT #1:MD$,J,J,K
1030 DISPLAY "DSK":STR$(MD);"-DISKNAME=";MD$:"AVAILABLE=";K;"USED=";J-F
1040 DISPLAY "FILENAME SIZE TYPE F"
1050 DISPLAY "-----"
1060 FOR LOOP=1 TO 127
1070 INPUT #1:MD$,MD,J,K
1080 IF LEN(MD$)=0 THEN 1140
1090 DISPLAY MD$;TAB(12);J;TAB(17);TYPE$(ABS(MD));
1100 IF ABS(MD)=5 THEN 1110
1110 IF MD<0 THEN 1130
1120 DISPLAY TAB(28);"Y";
1130 NEXT LOOP
1140 CLOSE #1
1150 PRINT : "PRESS ANY KEY TO RETURN" :: K,S=0
*1160 CALL KEY(O,K,S):: IF S=0 THEN 1160
1170 RETURN
2000 CALL CLEAR :: CALL SCREEN(6):: PRINT TAB(6);"LOAD DATA": : : : SS=1 :: GOSUB
9000
2010 CALL CLEAR :: INPUT "FILE NAME?":FILE$
2020 PRINT : :TAB(6);"LOADING DATA"
2030 OPEN #1:"DSK1."&FILE$,INTERNAL,INPUT ,VARIABLE :: OF1=OF1+1
2035 INPUT #1:C$,NR,NC
2040 FOR R=0 TO NR :: FOR C=0 TO NC :: INPUT #1:A$(R,C),A(R,C)
2050 NEXT C :: NEXT R
2080 CLOSE #1 :: D,C=1 :: Z=0 :: X=12 :: Y=21 :: CF1=CF1+1
2090 RETURN
*2990 CALL CLEAR :: PRINT "ENTER/EDIT DATA": : : : GOSUB 9000 :: CALL CLEAR :: IF
SS=1 THEN 3030
*3000 CALL CLEAR CALL SCREEN(4):: INPUT "NUMBER OF ROWS":NR :: PRINT :: INPUT
"NUMBER OF COLUMNS":NC :: CALL CLEAR :: INPUT "TITLE":C$
*3005 IF C>14 THEN 3000 :: SS=1
3010 CALL CLEAR :: INPUT "ROW HEADING":A$(0,0):: FOR P=1 TO NR :: PRINT "ROW";R
;"NAME:" :: INPUT A$(R,0):: NEXT R :: D=1
*3020 CALL CLEAR :: FOR C=1 TO NC :: PRINT "COLUMN";C;"NAME:" :: INPUT A$(0,C)::
NEXT C :: C=1 :: X=12 :: Y=21
*3030 CALL CLEAR :: DISPLAY AT(1,1);"PRESS R TO RETURN TO MENU" :: ON WARNING NEX
T
*3035 DISPLAY AT(3,1):USING 3085:A$(0,0);A$(0,C);A$(0,C+1)
3040 FOR R=0 TO D+9 :: DISPLAY AT((R-Z)*2+3,1):USING 3090:A$(R,0);A(R,C);A(R,C+1
):: NEXT R :: R=0
*3050 CALL HCHAR((R-Z)*2+3,X,91):: CALL HCHAR((R-Z)*2+3,Y,93):: C=C+P
*3055 ACCEPT AT((R-Z)*2+3,X-1)SIZE(-8)BEEP VALIDATE(NUMERIC):A(R,C):: C=C-F
3060 CALL KEY(O,K,S):: IF S=0 THEN 3060
*3070 IF K=B THEN 3100 :: IF K=9 THEN 3130 :: IF K=10 THEN 3160 :: IF K=11 THEN 3
190 :: IF K=E THEN 3030
*3075 IF K=B2 OR K=114 THEN 200
3080 GOTO 3060
*3085 IMAGE "#####"
*3090 IMAGE "#####.###"
*3100 IF X=2 OR R=0 THEN 3060
*3110 CALL HCHAR((R-Z)*2+3,X,32):: CALL HCHAR((R-Z)*2+3,Y,32)
*3120 IF X=21 THEN X=12 :: Y=21 :: F=0 :: GOTO 3050
*3125 IF X=12 AND C>1 THEN C=C-1 :: F=0 :: GOTO 3030
*3128 IF X=12 AND C=1 THEN C=0 :: X=2 :: Y=11 :: F=0 :: GOTO 3250
*3130 IF R=0 THEN 3060 :: IF X=21 AND C+1=NC THEN 3060 :: IF X=21 AND C+1=NC THEN
C=C+1 :: F=1 :: GOTO 3030
*3140 CALL HCHAR((R-Z)*2+3,X,32):: CALL HCHAR((R-Z)*2+3,Y,32)
*3145 IF X=2 THEN C=1 :: X=12 :: Y=21 :: GOTO 3050
*3150 IF X=12 THEN X=21 :: Y=30 :: F=1 :: GOTO 3050
*3160 IF F=NF OR X=2 THEN 3060

```

```

*3170 CALL HCHAR((R-Z)*2+3,X,32):: CALL HCHAR((R-Z)*2+3,Y,32)
*3180 R=R+1 :: IF R=11 THEN D=11 :: Z=10 :: GOTO 3030
*3185 GOTO 3050
*3190 IF R=0 OR X=2 THEN 3060
*3200 CALL HCHAR((R-Z)*2+3,X,32):: CALL HCHAR((R-Z)*2+3,Y,32)
*3210 R=R-1 :: IF R=0 THEN 3250 :: IF R=10 THEN Q=1 :: Z=0 :: GOTO 3030
*3220 GOTO 3050
*3250 CALL HCHAR((R-Z)*2+3,X,91):: CALL HCHAR((R-Z)*2+3,Y,92):: C=C+F
*3260 ACCEPT AT((R-Z)*2+3,X-1)SIZE(-8)BEEP:A*(R,C):: C=C-F :: IF C=0 THEN C=1
*3270 GOTO 3060
*3280 RETURN
4000 CALL CLEAR :: CALL SCREEN(12):: PRINT TAB(8);;"CALCULATE MODE": : : : ON WAF
NING NEXT
4005 PRINT "OPERATION(+,-,*,/,^) : ";B*(1): "OPERATE ON ROWS(Y/N) : ";B*(2)
: "OPERATE ON COLUMNS : ";B*(3): :
*4010 PRINT "SINGLE PAIR OPER.(Y/N)? : ";B*(4): "ENTER CONSTANT:";B(4): "PUT RES
ULT IN ROW NO. : ";B(5): "PUT RESULT IN COLUMN NO:";B(6): :
4015 PRINT "FROM ROW NO. : ";B(7): "TO ROW NO. : ";B(8): :
4020 PRINT "FROM COLUMN NO. : ";B(9): "TO COLUMN NO. : ";B(10): :
DISPLAY AT(24,1):"(fctn6=calc.,fctn5=no calc.)"
4025 L=1
*4030 IF L<5 THEN ACCEPT AT(L*2+1,26)SIZE(-2)BEEP:B*(L):: GOTO 4040
*4035 IF L=5 THEN ACCEPT AT(11,17)SIZE(-8)BEEP VALIDATE(NUMERIC):B(4):: GOTO 4040
*4037 IF L>5 THEN ACCEPT AT(L*2+1,26)SIZE(-2)BEEP VALIDATE(DIGIT):B(L-1)
4040 CALL KEY(O,K,S):: IF S=0 THEN 4040
*4045 IF K=10 AND L<11 THEN L=L+1 :: GOTO 4030 :: IF K=10 AND L>11 THEN 4040
4050 IF K=11 AND L>1 THEN L=L-1 :: GOTO 4030
4052 IF K=12 THEN 4055 :: IF K=14 THEN 4090
4054 GOTO 4040
4055 F=B(7):: C=B(9)
4060 IF B*(1)="+" THEN 4100 :: IF B*(1)="-" THEN 4200 :: IF B*(1)="*" THEN 4300
: IF B*(1)="/" THEN 4400
4062 IF B*(1)="^" THEN 4500 ELSE 4040
4090 Z=0 :: O,C,SS=1
4095 RETURN
4100 IF B*(3)="Y" THEN 4160 :: IF B*(4)="Y" THEN 4115 :: IF B(4)<>0 THEN 4150
4110 FOR R=B(7)TO B(8):: FOR C=B(9)TO B(10):: A(B(5),C)=A(B(5),C)+A(R,C):: NEXT
C :: NEXT R :: GOTO 4090
*4115 IF B(4)<>0 THEN 4140
*4120 FOR C=B(9)TO B(10):: A(B(5),C)=A(B(7),C)+A(B(8),C):: NEXT C :: GOTO 4090
*4140 FOR C=B(9)TO B(10):: A(B(5),C)=A(B(7),C)+B(4):: NEXT C :: GOTO 4090
*4150 FOR R=B(7)TO B(8):: FOR C=B(9)TO B(10):: A(R,C)=A(F,C)+B(4):: NEXT C :: NEX
T F :: GOTO 4090
*4160 IF B*(4)="Y" THEN 4175 :: IF B(4)<>0 THEN 4150
*4170 FOR C=B(9)TO B(10):: FOR R=B(7)TO B(8):: A(R,B(6))=A(F,B(6))+A(R,C):: NEXT
F :: NEXT C :: GOTO 4090
*4175 IF B(4)<>0 THEN 4190
*4180 FOR R=B(7)TO B(8):: A(R,B(6))=A(R,B(9))+A(R,B(10)):: NEXT R :: GOTO 4090
*4190 FOR R=B(7)TO B(8):: A(R,B(6))=A(R,B(9))+B(4):: NEXT R :: GOTO 4090
*4200 IF B*(3)="Y" THEN 4260 :: IF B*(4)="Y" THEN 4215 :: IF B(4)<>0 THEN 4250
*4210 FOR R=B(7)TO B(8):: FOR C=B(9)TO B(10):: A(B(5),C)=A(B(5),C)-A(R,C):: NEXT
C :: NEXT R :: GOTO 4090
*4215 IF B(4)<>0 THEN 4240
*4220 FOR C=B(9)TO B(10):: A(B(5),C)=A(B(7),C)-A(B(8),C):: NEXT C :: GOTO 4090
*4240 FOR C=B(9)TO B(10):: A(B(5),C)=A(B(7),C)-B(4):: NEXT C :: GOTO 4090
*4220 FOR R=B(7)TO B(8):: FOR C=B(9)TO B(10):: A(R,C)=A(F,C)-B(4):: NEXT C :: NEX
T F :: GOTO 4090
*4220 IF B*(4)="Y" THEN 4275 :: IF B(4)<>0 THEN 4250
*4270 FOR C=B(9)TO B(10):: FOR R=B(7)TO B(8):: A(R,B(6))=A(F,B(6))-A(R,C):: NEXT
F :: NEXT C :: GOTO 4090
*4275 IF B(4)<>0 THEN 4290
*4280 FOR R=B(7)TO B(8):: A(R,B(6))=A(R,B(9))-A(R,B(10)):: NEXT R :: GOTO 4090
*4290 FOR R=B(7)TO B(8):: A(R,B(6))=A(R,B(9))-B(4):: NEXT R :: GOTO 4090
*4300 IF B*(3)="Y" THEN 4360 :: IF B*(4)="Y" THEN 4315 :: IF B(4)<>0 THEN 4350
*4310 FOR R=B(7)TO B(8):: FOR C=B(9)TO B(10):: A(B(5),C)=A(B(5),C)*A(F,C):: NEXT
C :: NEXT R :: GOTO 4090
*4315 IF B(4)<>0 THEN 4340
*4320 FOR C=B(9)TO B(10):: A(B(5),C)=A(B(7),C)*A(B(8),C):: NEXT C :: GOTO 4090
*4340 FOR C=B(9)TO B(10):: A(B(5),C)=A(B(7),C)*B(4):: NEXT C :: GOTO 4090
*4250 FOR R=B(7)TO B(8):: FOR C=B(9)TO B(10):: A(R,C)=A(F,C)*B(4):: NEXT C :: NEX
T R :: GOTO 4090
*4360 IF B*(4)="Y" THEN 4375 :: IF B(4)<>0 THEN 4350

```

The 9900 User's Group

```

* 4370 FOR C=B(9)TO B(10):: FOR R=B(7)TO B(8):: A(R,B(6))=A(R,B(6))*A(R,C):: NEXT
R :: NEXT C :: GOTD 4090
* 4375 IF B(4)<>0 THEN 4390
* 4380 FOR R=B(7)TO B(8):: A(R,B(6))=A(R,B(9))*A(R,B(10)):: NEXT R :: GOTD 4090
* 4390 FOR R=B(7)TO B(8):: A(R,B(6))=A(R,B(9))*B(4):: NEXT R :: GOTD 4090
* 4400 IF B*(3)="Y" THEN 4460 :: IF B*(4)="Y" THEN 4415 :: IF B(4)<>4 THEN 4450
* 4410 FOR R=B(7)TO B(8):: FOR C=B(9)TO B(10):: A(B(5),C)=A(B(5),C)/A(R,C):: NEXT
C :: NEXT R :: GOTD 4090
* 4415 IF B(4)<>0 THEN 4440
* 4420 FOR C=B(9)TO B(10):: A(B(5),C)=A(B(7),C)/A(B(8),C):: NEXT C :: GOTD 4090
* 4440 FOR C=B(9)TO B(10):: A(B(5),C)=A(B(7),C)/B(4):: NEXT C :: GOTD 4090
* 4450 FOR R=B(7)TO B(8):: FOR C=B(9)TO B(10):: A(R,C)=A(R,C)/B(4):: NEXT C :: NEX
T R :: GOTD 4090
* 4460 IF B*(4)="Y" THEN 4475 :: IF B(4)<>0 THEN 4450
* 4470 FOR C=B(9)TO B(10):: FOR R=B(7)TO B(8):: A(R,B(6))=A(R,B(6))/A(R,C):: NEXT
R :: NEXT C :: GOTD 4090
* 4475 IF B(4)<>0 THEN 4490
* 4480 FOR R=B(7)TO B(8):: A(R,B(6))=A(R,B(9))/A(R,B(10)):: NEXT R :: GOTD 4090
* 4490 FOR R=B(7)TO B(8):: A(R,B(6))=A(R,B(9))/B(4):: NEXT R :: GOTD 4090
* 4500 IF B*(4)="Y" THEN 4520
* 4510 FOR R=B(7)TO B(8):: FOR C=B(9)TO B(10):: A(R,C)=A(R,C)^B(4):: NEXT C :: NEX
T R :: GOTD 4090
* 4520 IF B*(3)="Y" THEN 4540
* 4530 FOR C=B(9)TO B(10):: A(B(5),C)=A(B(7),C)^B(4):: NEXT C :: GOTD 4090
* 4540 FOR R=B(7)TO B(8):: A(R,B(6))=A(R,B(9))^B(4):: NEXT R :: GOTD 4090
* 4550 A(R,C)=A(R,C)^B(4):: C=C+1 :: IF C=B(10)THEN 4070 ELSE 4550
5000 CALL CLEAR :: CALL SCREEN(12):: PRINT TAB(6);"SAVE DATA": : : : : : GOSUB 90
00
5010 CALL CLEAR :: INPUT "FILE NAME?":FILE$ :: OPEN #1:"DSK1."&FILE$,INTERNAL,O
UTPUT,VARIABLE :: OF1=OF1+1
5015 PRINT #1:C$,NR,NC
5020 FOR R=0 TO NR :: FOR C=0 TO NC :: PRINT #1:A$(R,C),A(R,C):: NEXT C :: NEXT
R
5060 CLOSE #1 :: CF1=CF1+1 :: C,Q=1 :: Z=0 :: X=12 :: Y=21
5070 CALL CLEAR :: PRINT TAB(6);"OPERATION COMPLETE": : : : : : FOR DELAY=1 TO 50
: : NEXT DELAY
5080 RETURN
6000 CALL CLEAR :: CALL SCREEN(4):: PRINT TAB(8);"PRINT DATA": : : : : : GOSUB 90
00
6010 OPEN #2:"PI0.DA=B.LL=132",VARIABLE 132 :: OF2=OF2+1
6020 CALL CLEAR :: PRINT :: INPUT "DOUBLE SPACE?(Y/N)":DS$
6040 IF NC>8 THEN PRINT #2:CHR$(15);
6045 PRINT #2:C$ :
6050 FOR C=0 TO NC :: PRINT #2,USING 6200:A$(0,C):: NEXT C :: PRINT #2: :
6060 FOR U=1 TO NC*10 :: PRINT #2:CHR$(45):: NEXT U :: PRINT #2: : :
6070 FOR R=1 TO NR :: PRINT #2:A$(R,0);TAB(9);
6080 FOR C=1 TO NC :: PRINT #2,USING 6300:A(R,C):: NEXT C :: PRINT #2: :
6100 IF DS$="Y" OR DS$="y" THEN PRINT #2: :
6110 NEXT R
6120 PRINT #2:CHR$(18)
6130 CLOSE #2 :: CF2=CF2+1 :: C,Q=1 :: Z=0 :: X=12 :: Y=21
6140 RETURN
6200 IMAGE "***** "
6300 IMAGE "###.###.## "
7000 PRINT TAB(6);"CLEAR MEMORY": "REMINDER! SAVE TEXT FIRST!" :: GOSUB 9000 ::
CALL CLEAR :: PRINT TAB(6);"MEMORY CLEARING"
7010 FOR R=0 TO 20 :: FOR C=0 TO 13 :: A$(R,C)=" " :: A(R,C)=0 :: NEXT C :: NEXT
R :: SS,OF1,CF1,OF2,CF2=0
7020 RETURN 200
9000 PRINT "PRESS R TO RETURN TO MENU": "PRESS C TO CONTINUE"
9010 CALL KEY(O,K,S):: IF S=0 THEN 9010 :: IF K=B2 OR K=114 THEN 200
9020 RETURN
9500 CALL SCREEN(10):: CALL ERR(CODE,TYPE,SEVER,LINE)
9510 CALL CLEAR :: PRINT TAB(10);"ERROR!": : "ERFOR CODE";CODE;"IN LINE";LINE:
:"TYPE";TYPE: :
9515 IF OF1<CF1 THEN CLOSE #1 :: IF OF2<CF2 THEN CLOSE #2
9520 PRINT "PRESS ANY KEY TO RETURN": "TO MENU"
9530 CALL KEY(5,K,S):: IF S=0 THEN 9530
9550 RETURN 200

```

This spreadsheet is similar to Snapcalc but appears to be much faster. Mr Delpit will give a demonstration of the program's capabilities at one of our regular meetings.

Ed. (LA 99'ers)

Last Minute Notes:

1. If you have any software to donate etc. See the librarians.
2. There will be NO combined splinter group in JUNE
3. We are trying to set up a meeting place for JUNE.

One of two places

Cinnaminson Community Center (minor cost)
or
Deptford Municipal Building (free)

THE 9900 USER'S GROUP, INC.
P.O. BOX K
MOORESTOWN, N.J. 08057



Edmonton 99'ers Users Society
PO Box 11983, Edmonton
Alberta, CANADA T5J-3L1

Exchnq