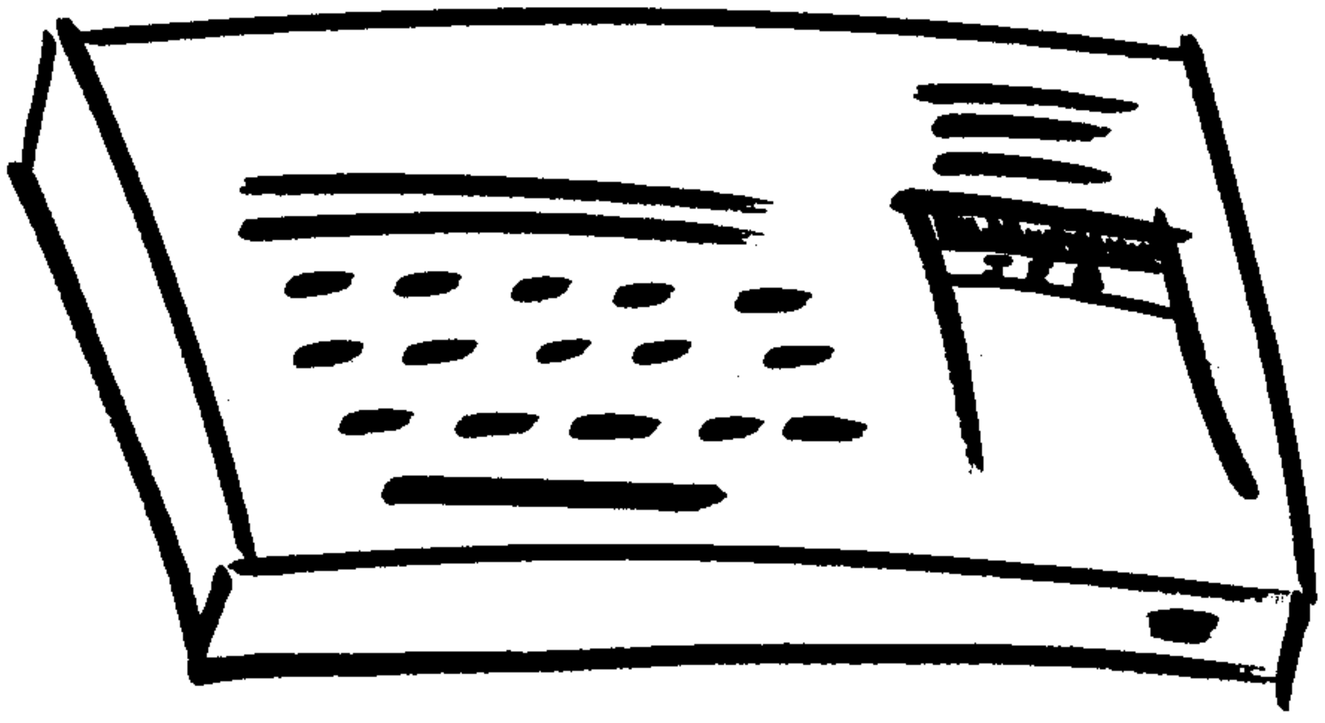
99 (14A)

TIPS



99 TIPS

(Editor's note - this portion of the newsletter will give tips [99 per year] for using your computer, peripherals, software, etc 'lease help me out and pass on your favorite tips when you see me at the monthly meetings. Thanks.)

01) If you have received your speech synthesizer and have bought the Terminal Emulator II you may be in for quite a surprise. The next time you are writing a program and get tired looking at your program lines scrolling past you when you type LIST, type this in LIST "SPEECH" and sit back and listen to the computer tell you what you've written!

02) If you have bought Extended Basic (check page 5b for a good price) try this out sometime.
   a. Type 100
   b. Type !
   c. Type Ctrl G Ctrl H Enter
   d. LIST line 100
   You should see this 100 !GOSUB RETURN
   Since all the keywords (words like PRINT, GOTO, READ, etc) are represented by a number (129 - 254) and Ctrl any-key gives you these numbers, you can program in Extended Basic by using one key. Just be sure and start each line off with an exclamation mark (!). When you are done just delete the (!). Try and see what keywords you can come up with (also try the Fctn key).

03) Don't forget some of the basic commands that you may not have used for awhile. Surprised to find one that you did not know about? Here are three that are quite helpful.
   a. NUM X,Y will automatically generate line numbers beginning with line X and incrementing the line number by Y each time you enter a line.
   b. RES X,Y will renumber all the lines (including GOTO and GOSUB). The first new line number will be X and all subsequent lines will be incremented by Y.
   c. LIST X-Y will list only those lines between lines X and Y. Very handy for debugging parts of your program.

04) Do you want all of your decimal points to line up when you print out columns of numbers? Here's a one line routine that I wrote to solve such a problem.

```
100 PRINT TAB(C-POS(STR$(N)&".",".",1)-(N<0));N
```

N = number to be printed
C = what column on the screen you want the decimal to appear
note - limit C to columns 10 - 19

05) If you have been having trouble with your cassette recorder in saving and loading programs try these tips.
   a. Try the cheaper (50 cent) brand tapes.
   b. Demagnetize and clean your tape heads.
   c. Unplug the ear or mic jack when saving or loading.
   d. Move the tape recorder far away from the computer.
   e. Don't use the first minute on the tape (it may be bad).
   f. Give your recorder to a needy person and buy a new one!

06) Do you wish that regular Basic had Extended Basic's "hold" feature when listing your programs on the screen? Well there is a simple way to do this. Type in the first line number you want to see and hold down the Fctn X (the down arrow) key. As long as the X key is held down the program will continue to list out. If you should want to stop the listing just release the X key. To start the listing back up hold the Fctn X key down. If you want to stop the listing altogether release the X key and press the Enter key.

07) One of the nicest features in the Extended Basic module is the Redo (Fctn 8) key. Let's say that you typed in a long line only to have the computer tell you that it didn't like it for some reason. Rather than to cuss at the computer and then retype the entire line you simply press Redo and presto, the entire line appears ready for you to make the correction. Redo can also be used to repeat similar lines of code.

08) Did you know that you can have more than four lines of code in a program line? When you reach the end of the fourth line and the computer beeps at you, type Enter. Then type the line number and Fctn X. Move the cursor to the end of the fourth line and you will be able to type a fifth line. When the beep is heard, type Enter. Repeat the process for a sixth line. I am able to get part of a seventh line sometimes. It depends on what is in the line (some code takes more space in memory - see last month's TIP 2).

09) If you program a lot then SAVE a lot. A good rule to follow is to save what you are working on every 20 minutes. It may seem like a pain to do, but one of these days you will be very happy that you did. I once lost over 3000 lines of a program because I thought it was a pain to take the time to save. You can be sure that I never think that again! Always keep good track of what you save. If you keep three tapes/disks handy you can rotate through them so that at any given time you will have the last three saves available. This is also handy if you decide that your last set of changes aren't that great and you want to go back to before the changes and try something else. It always pays to SAVE!

10) Back to Extended Basic. One of the limitations of any form of Basic is the lack of structure it offers in identifying what the program does. With the "::" and "!" features you can take steps to make your programs readable. The "::" lets you put more than one statement on a line. The "!" acts like a REM. Here's an example to illustrate the idea -

```
100 DIM ANSW$(10) :: FOR QUES=1 TO 10 :: INPUT "ANSWER TO
#";QUES:ANSW$(QUES) :: NEXT QUES :: !GET ALL 10 ANSWERS
```

Here in one line you are accomplishing a complete action. It will be much easier for someone else to figure out what you wanted this particular piece of code to do.

11) Send the editor some tips!

12) When you use the CALL JOYST command in a program, you must specify which joystick the computer is to use (#1 or #2). Then, whenever you RUN the program, you must be using that joystick or the computer will not be able to find your input. By including these lines at an appropriate place in your program (generally after any instructions and with appropriate renumbering) you can eliminate the problem of making sure you have the correct joystick by using CALL JOYST(JS,K,S).

```
100 PRINT "Press Fire Button to Continue"
110 CALL KEY(1,K1,S)
120 CALL KEY(2,K2,S)
130 IF K1+K2<>17 THEN 110
140 JS=INT(K1/18+K2/9+1)
```

13) If you would like to replace one of your TI joysticks with an Atari joystick (usually $7 at Target), here is the wiring that you need to do. Be sure and unsodder the 5 diodes (the little brown cylinders on the TI circut board) and sodder them in the same direction between the TI cable and Atari joystick.

TI cable color	Atari cable color	Diode needed
blue	brown	yes
white	black	no
brown	green	yes
green	blue	yes
black	orange	yes
orange	white	yes

14) Here's an oldie but goodie. If you would like to find out how much memory you have left while you're in regular BASIC, then insert these two lines in your program and type RUN.

```
1 A=A+8
2 GOSUB 1
```

When you get a MEMORY FULL error, type PRINT A, and you will have the amount of memory that remains (+ 34 bytes for lines 1 and 2). The computer uses 8 bytes of memory to store the return address for a GOSUB (that's why you add 8 to A).

15) A screen dump to your printer (no redefined characters).

```
100 OPEN #1:"device name",OUTPUT
110 FOR R=1 TO 24
120 FOR C=1 TO 32
130 CALL GCHAR(R,C,G)
140 PRINT #1:CHR$(G);
150 NEXT C
160 PRINT
170 NEXT R
180 CLOSE #1
```

16) To improve my sort in last month's newsletter (page 4) by 3 seconds, make this change ---> 1001 C=INT(B*.75)

99 TIPS

Editor's note - Only three people have given me anything at all to publish in the newsletter (that's 3 items in 4 months, folks). If you like this section of the newsletter, you had better submit some ideas or page 2 will quickly fade away. Remember, anything that you may have is worth something if someone else can learn from it. This month I "borrow" from the other newsletters around the country.

17) Jim Peterson, of Tigerclub Software in Columbus, OH, passes on this gem to the readers of his monthly newsletter. If you mistakenly type "OLD CS1" when you meant "SAVE CS1", your program won't be lost if you type "Shift E Enter". You'll get an I/O error, but still have your program!

18) Paul Schippnick, of the SGV Users Group in West Covina, CA, reports that if you have a disk drive and Editor Assembler or Mini Memory and want to free up the memory that your disk uses (assuming you will use a cassette to SAVE), all you need to do is type "CALL LOAD(-31888,63,255)". Typing "BYE" returns you to the main screen and allows the disk to be used again.

19) The International Users Group in Bethany, Oklahoma, reports there is a hidden "test mode screen" in the Munch Man module. Within 3 seconds from the MM title screen, type *#* and you will see a screen asking you for RND (enter round 0 to 2), SCN (enter screen 0 to 19), and MM (enter munchmen 1 to 9). RND 2 and SCN 19 are the toughest. Have fun!

20) From TI comes information on getting your broken computer up and running again. Send the unit to (you pay for shipping and insurance to TI, they provide for the return trip) -

Texas Instruments Repair Center
2305 North University
Lubbock, Texas 79408

If the computer is under the 90 day warranty (send proof of purchase), then there is no charge. If it's no longer in the warranty period, the charge is $28 for a minor repair and $48 for a major repair. You do not need to send payment, as they will inform you of the cost when they receive the unit. The usual waiting period to get your unit back is three weeks.

21) I received from TI (in a game module) an addendum for Extended Basic. If you BREAK your program (ex - Fctn 4), type a command that gives an error (ex - ON), and try to continue your program (ex - CON), your program may be erased or your keyboard may lock up. Type PRINT then CON to prevent this. Also, if you use ON GOTO or ON GOSUB with a space between the numbers (ex - ON A GOTO 500 ,600 ,700) the line won't "goto" to the line with the space (even though it LISTS ok). Editing will fix things up.

2) If you have Extended Basic, then you probably know that you have 13928 bytes of memory to work with. If you have TE II then you have 14024 bytes. Regular basic (no module) will give you 14536 bytes. Reduce these values by 2808 bytes if you have the disk drive attached (unless you use tip 18)!

3) Tom Boelling reports that tip 19 can also be found in the Alpiner module. Type *## after the title screen shows and input the number of players (1 - 2), number of lives (0 - 9), level of difficulty (0 - 18), and the player's names. Levels 13 to 18 are tough to get to the top. At level 6 you will see the Abominable Snowman on skis! Happy climbing.

24) Bob Kodis had module problems. He suggests if you are annoyed with your system "locking up" while you are using modules, the problem may be that the modules are not being held into the console tightly enough. Try putting a matchbook cover underneath the module as you slide it in. This should hold the module tightly enough. It worked for him.

25) TI will sell you replacement parts for some of their equipment (example - joystick handle). Write them and describe what you need (any part number will help). You will be sent the part (if they have it) along with a bill for the part and postage.

Texas Instruments Parts Dept.
P.O. Box 53
Lubbock, Texas 79408

26) To further document your program you can add comments after a GOTO or a GOSUB. They won't interfere with the program. Here are two examples -

100 GOSUB 1234 this sets up the screen
200 GOTO 2345 execute the print routine

27) In most of the newsletters I have seen, there are routines for printing lower case letters on the 99/4A (rather than the reduced upper case letters TI included). Well, not to be left out, here are my lower case letters.

Lines 10001 to 10003 contain the information needed to redefine the lowercase letters.

Line 10004 sets the READ to the proper line.

Lines 10005 to 10008 actually do the redefining of the characters.

Line 10009 ends the routine.

Lines 1,3,4 are to demonstrate the small letters (type a few words with the alpha lock off).

Line 2 should be included in your program before you need to display the letters.

Feel free to change any of the patterns to suit your taste.

```
1 CALL CLEAR
2 GOSUB 10000
3 INPUT X$
4 GOTO 3
10000 REM LOWERCASE
10001 DATA 0000003C444443C,004040
78444478,0000003C40403C,00040430
444443C,0000102350443B,0018247020
2020,0000003C443C0438,00404058
4444,0010001010101 0
10002 DATA 0003000808484830,0041
4850704844,0010101010101 0,000001
6C545454,000005B644444,00000038
444438,0000078447840 40,0000003C
443C0404
10003 DATA 000000586440 40,000001
3C301878,00101 7C101010,0000004
444438,0000044442810,00000445
5428,0000044333844,0000044281
2040,0000007C18307C
10004 RESTORE 10001
10005 FOR C=97 TO 122
10006 READ C$
10007 CALL CHAR(C,C$)
10008 NEXT C
10009 RETURN
```

99 TIPS

28) Tom Boelling reminds us that you can type RUN "CS1" or RUN "DSK1.name" in Extended Basic, and the computer will load the program and begin executing it.

29) Also in Extended Basic is a feature that looks for the program name "LOAD" on DSK1. If it's there, it will load and run that program whenever you enter Extended Basic.

30) David Thrasher reports that the Brother Company makes two portable typewriters that can be interfaced to the 99/4A (with a 232 interface). They are models EP22 ($250 at Younkers) and CE60 ($530 at Ardans). They will print letter quality text at approximately 15 characters per second. David has a brochure if you have further questions.

31) I just received my Editor Assembler software and have found that the editor portion works as a very nice word processor. It has 80% of the features most wp packages have (actually, I think only the formatting commands are really missed). It is a cheap way to get a word processor, and the assembler, the debugger, and Tombstone City are tossed in free!

32) If you are running out of memory when you program, maybe you need to take a look at the way you use variables and arrays. Here are all the memory requirements for variables and arrays (how many bytes of memory they take).

```
Numeric Variable = 6 + L + 8
Numeric Array    = 6 + L + 8*E + 2*D
String Variable  = 7 + L + 6   + 1*C
String Array     = 7 + L + 6*E + 1*C + 2*D

L - number of letters in the variable/array name
E - number of elements in the array
D - number of dimensions in the array
C - number of characters in the string
```

33) For those of you who have the Mini Memory or Editor Assembler cartridge, here is a little program that allows you to see the screen in Normal Mode, Clear Mode (everything is there but is invisible), Text Mode (40 characters across), Multicolor Mode (each character is made up of 4 blocks), and Bit Map Mode (you need the 4/A to see each pixel).
No need to know assembler to access these, but if you would like them explained (and some ideas on how to work with them), drop in on one of my assembler classes.
After you run the program, press N,C,T,M,B (the screen may not be readable, but the keys still work).

```
100 PRINT "PRESS A KEY ===>  N,C,T,M,B  ":
110 CALL KEY(5,K,S)
120 IF K<>78 THEN 140
130 CALL POKEV(-32768,0)
140 IF K<>67 THEN 160
150 CALL POKEV(-32352,0)
160 IF K<>84 THEN 180
170 CALL POKEV
(-32272,0,"",-30945,0)
180 IF K<>77 THEN 200
190 CALL POKEV(-32280,0)
200 IF K<>66 THEN 220
210 CALL POKEV(-32766,0)
220 GOTO 100
```

34) If you have a question concerning your 4A please feel free to drop me a line and I will answer it in this column. Write t

John Hamilton
4228 E Clinton
Des Moines, Iowa 50317

35) A correction to November 1983 (page 7). Add these two lines to make the disappearing graphics routine work.

```
240 X=4-1*(X<0)-2*(Y>0)-3*(X>0)
250 CALL COLOR(X,1,1)
```

36) I have received quite a few calls concerning problems with loading/saving cassette programs. Most are due to not having the 'Alpha Lock' key latched. The 4A does not recognize small case letters as a valid device. However, it must be unlatched to properly use the joysticks. TI does complicate life!

37) If you are having problems with your computer causing hum or static with your tv (and adjusting the fine tuning doesn't get rid of it), then pop the top off of the rf modulator (note - this might void your warranty) and try adjusting the small coils (the ones with the screws in them). Be very careful not to touch any of the other components while you are adjusting these. Also, do not turn more than a 1/4 turn at a time to avoid messing the whole thing up.

38) If you write programs that utilize speech (using the Extended Basic module) the following will tell you if the speech synthesizer is attached.

```
100 CALL PEEK(-28672,S)
110 IF S=0 THEN 130      or    IF S<>96 THEN 130
120 your speech routine goes here
130 skip to here if no speech attached
```

39) A quick way to make two copies of your program on tape at the same time is to hook up the second set of cassette cables to a second cassette recorder. When you type "SAVE CS1" the computer actually sends a tone out to both recorders. That way if something ever happens to the first tape you will have a second copy to fall back on.

40) For those of you who would like to sort names using the sort I gave on page 4 of the August 1983 issue (also see page 6 of the December 1983 issue), simply replace the A(value) with an A$(value) everywhere the first occurs. TI basic will compare strings alphabetically for you.

41) The 99/4A has 256 bytes of CPU RAM memory in it. When you run a basic program, several of these bytes are used by the computer to keep track of things (joysticks for example). Try th one-liner in Extended Basic. Let me know what you think thi byte is used for (I'll describe more of the bytes next month).

```
100 CALL PEEK(-31879,C) :: PRINT C :: GOTO 100
```

42) TI announces a reduction in the exchange price for the 99/4A. A reconditioned computer can be had for $28.50 and your old computer. Check the literature that came with your manuals for the nearest Exchange Center.

43) Miller Graphics announces an unpublished memory location that allows you to disable the Fctn QUIT key if you have either Editor Assembler, Mini Memory, or Extended Basic (EB also must have the 32K and will need a CALL INIT).

CALL LOAD(-31806,16) will disable Fctn QUIT.
CALL LOAD(-31806,0) will enable it.

44) I've received a couple of questions concerning the LIST command. Here are some things to remember.

1) LIST will list all of the program (on the screen).
2) LIST 500- will list all lines from 500 on.
3) LIST -500 will list all lines up to 500.
4) LIST 500-900 will list all lines from 500 to 900.
5) LIST "device": with any of the above will cause the lines to go to that "device" (but no lines will list on the screen).

45) For those of you who may be trying to hook up printers to the 4A and did not buy a cable that was especially made for the 4A (either parallel or serial), you might want to check that the pin assignments of the cable are correct.

Parallel

TI	Printer
1-9	1-9
10	11
16	16
ground	17

to per.exp.box

Serial

TI	Printer
1	1
2	2
3	3
5	5
6	6
7	7
8	8
20	11

46) Last month's TIP 41 was taking a look at the VDP interrupt timer. It gets incremented once every 1/60th second (0 to 255 and then back to 0). Here are some other bytes to play with (you will need the E A, M M, or E B module).

CALL PEEK(-31880,X) ==> random number (0-99), use RANDOMIZE.
CALL PEEK(-31878,X) ==> highest numbered sprite in motion.
CALL PEEK(-31794,X) ==> timer for CALL SOUND duration (255-0).

47) Need some sound affects? Try these with the TEII and speech synthesizer using the program on page 37 of the TEII manual.

1 "KKKKKKKK" or "QQQQQQQQ" - steam locomotive
2 "UUUUUUUU" or "WWWWWWWW" - helicopter
3 "VVVVVVVV" or "YYYYYYYY" - small plane
4 "JJJJJJJJJJJJJJJJJJJJJJJ" - machine gun
5 "XXXXXXXXXXXXXXXXXXXXXXX" - sewing machine

Experiment with the # of letters & mix them together.

99 TIPS

48) Two of the BBS's phone numbers have been changed.
Computerland ... 270-8942
Omni 270-6883

49) For those of you who have the Forth disk, Mike McCann from the Omaha Users Group has this correction.
"In screen #72, on line 5, the variable PAB_ADDRESS must be changed to PAB-ADDR to work properly."

50) Academy on Computers is a television series on IPBN (channel 11). It consists of twelve 30 minute shows that begin on April 17 (Tuesdays at 8 pm) and April 21 (Saturdays at 9 am). Course material is available from IPBN for $70. For further information, see Tom Boelling at the next meeting.

51) Ken DePue found out the hard way how to print 132 characters on his printer - he had to call TI (for those of you who have been put on eternal hold by TI, you will understand).

```
OPEN #1:"PIO",VARIABLE 132
send the printer code for condensed printing
PRINT #1:"anything you like up to 132 characters"
```

52) If you have the TEII module and speech synthesizer you can write a basic program to read stories from childrens' books to your kids. Here is an example.

```
100 OPEN #1:"SPEECH",OUTPUT
110 FOR PAGE=1 TO 6
120 READ WORDS$
130 PRINT #1:WORDS$
140 CALL KEY(5,K,S)
150 IF S=0 THEN 140
160 NEXT PAGE
170 DATA ONCE UPON A TIME
180 DATA THERE WERE THREE BEARS
190 DATA WHO LIVED IN THE WOODS
200 DATA ATE LOTS OF HONEY
210 DATA AND LIVED HAPPILY
          EVER AFTER
220 DATA THE END
```

Just type a DATA statement for each page (but do not use any commas). After you read the page, hit any key to continue.

53) Here are some handy TI-Writer tips from Ken DePue.

```
CTRL T - move the cursor left one tab setting.
CTRL V - move the cursor to the beginning of the current line.
CTRL L - move the cursor to the upper left corner of screen.
CTRL Y - release the left margin.
CTRL U - allows use of ASCII codes 0 to 31 (manual page 146).
.CU    - same as a REM statement in BASIC.
.CE n  - centers "n" number of lines.
```

54) If you would like to use your computer to dial your phone for you (you must hold the receiver up to the tv speaker), here are the values for the CALL SOUND statements you will need.

Ex - CALL SOUND(250,697,0,1209,0) to dial a "1"

1 - 697 & 1209	4 - 770 & 1209	7 - 852 & 1209	* - 941 & 1209
2 - 697 & 1336	5 - 770 & 1336	8 - 852 & 1336	0 - 941 & 1336
3 - 697 & 1447	6 - 770 & 1447	9 - 852 & 1447	# - 941 & 1447

99 TIPS

55) There is a new BBS in town. It's called the "Great Roosevelt Roughrider System". It is similar to the Omni and Computerland boards. The hours are from 7 pm to 7 am. The number is 223-5977.

56) "47st.photo" ran an ad in the 4/3 Wall Street Journal for the PEB, controller card, disk drive, and 32K for $449.95. No promises that they have any, but if you have been trying to buy this, give them a call at 1-800-221-7774.

57) The rumor is if you have a white console and your screen shows (C) 1983 when you turn it on, then you have one of TI's protected GROM machines. What this means is that you won't be able to run any cartridges that TI doesn't license (so look before you buy).

58) Ken DePue supplies us with these tips for Multiplan -
CTRL E,S,D,X - instead of moving one cell at a time, move the screen one page at a time (ie E & X move the cell pointer up or down 18 cells at a time).
ENTER - to select the option that is highlighted on the command line, press Enter, or the first letter of any command.
LOWER RIGHT - when Multiplan is first selected, Home and Lower Right are the same cell. To find out how many more cells are available for actual use, press FCTN 1 (LOWER RIGHT).
RECALC - to avoid the ever increasing time of recalculating the entire spreadsheet each time a new value is entered, turn off this feature by 1)pressing 0, 2)pressing space bar to move to NO, 3)pressing Enter. To activate the Recalc feature if you need it, press FCTN 8 (RECALC).

59) For all you Atari game fans, here are some cartridges that are for the TI - Centipede, Defender, Dig Dug, Donkey Kong, Pac Man, Picnic Paranoia, Protector II, Robotron 2084, Shamus, Stargate, & Super Storm. They sell anywhere from $20 to $35.

60) Somebody supplied me with the pin assignments for all of the "plugs" on the TI (please identify yourself and I'll give you the credit in the next newsletter). Here are a couple of them. Pins run left to right with 1-5 on top and 6-9 on bottom.

Cassette

1-CS1 motor pos	6-CS2 motor pos
2-CS1 motor neg	7-CS2 motor neg
3-Ground	8-Audio in
4-Not used	9-Audio ground
5-Record out	

Joystick

1-Not used	6-Not used
2-Joystick 2	7-Joystick 1
3-Up	8-Down
4-Fire	9-Right
5-Left	

61) While talking on the phone to a technician at TI, I discovered that TI accidently shipped some prototype 248K memory cards (they look like 32K cards). These cards use a special trinary (rather than binary) circuit. To see if you have one (only fifteen were shipped), insert the Editor/Assembler cartridge, go to Edit mode, type 80 "X"s, use the Copy function to get 2 lines of "X"s (Fctn 9 C 1,1,1), then copy 2 lines giving 4 lines, 4 to 8, 8 to 16, etc. If you can get 3065 lines, then you have the 248K card (80 char x 3065 lines = 240K). If you only get 306 lines, then you have the 32K card (80 x 306 = 24K). Note that 8K is used for the Editor.

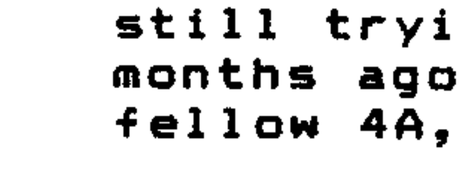
Last line 40 58

62) If you would be interested in meeting with different computer users (Apple, Atari, Commodore, IBM, etc), Madonna Nordaker would like to talk to you (her number is 262-8619). She is interested in organizing a get-together with different users and exchanging information about their computers.

63) TI has reached a preliminary agreement with March Direct Marketing to provide a means for other manufacturers of hardware and software to offer you their products by mail. Write TI at the address on page 1 to insure that MDM has your address (MDM will send you a free catalog of suppliers).

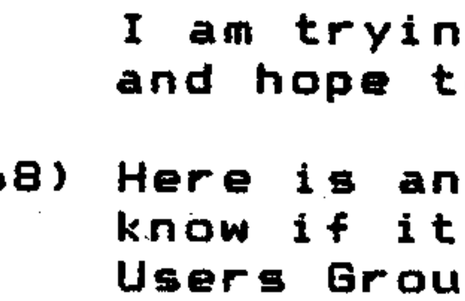
64) Here are some more pin diagrams for the 4A.

Power



1 - not used
2 - 8 volt AC
3 - 16 volt AC
4 - ground

Video



1 - +12 volt
2 - video out
3 - shield
4 - ground
5 - sound out

65) The May issue of Microcomputing had an article on how you can transfer a program from a TI 99/4A to an IBM PC. You need an RS232 card for each machine (and modem if the machines are not located together). I have a copy of the article if anybody is interested in doing this.

66) Bob Utter still has a few Forth Manuals for $8.00. The disk is in the software library and is free to copy.

67) Thanks to Steve Tetzloff for his TEII protocol manual (I am still trying to get TI to send me one - they cashed my check months ago!). Next time you are using your modem to call a fellow 4A, send this set of keystrokes -

Ctrl . G Fctn V Ctrl . (+ ! ' Ctrl .)

I am trying to put together a summary of what's in the manual and hope to present it at one of the Modem SIG meetings.

68) Here is an undocumented tip (means I haven't tried it or don't know if it even works) from Jon Zittrain of the Pittsburgh Users Group. In the immediate mode type the following -

SIZE CALL INIT CALL LOAD(-31866,33,0) SIZE

It appears that the extra 8K of the memory expansion is now at your disposal. Let me know what you experience with this.

69) Triton Marketing is now handling all 4A equipment. Call them at 1-800-227-5257 for a free catalog of what they have available (they are distributing the remainder of TI's inventory).

99 TIPS

Since nobody from our club seems to be able to come up with any ideas for this column, I will raid the other newsletters in order to fill up this page. Look for page 2 to disappear soon unless there is some action by you!

My thanks to the following user groups for these tips - Atlanta, Cincinnati, Denver, Honolulu, Pittsburgh, San Gabrial, Miller Graphics, and Tigercub.

70) In order to back up the Multiplan disk, you must use the name "TIMP" when initializing the backup disk.

71) To change the screen color while in command (immediate) mode in Extended Basic, type the following (n is the screen color you want, from 3 to 16) -

    CALL SCREEN(n) :: ACCEPT AT(1,1):A Enter Fctn 4

72) To prevent the screen from scrolling the 1st line off when you print the 24th line, use PRINT "whatever"; on the 24th line (the semi-colon prevents the screen from scrolling).

73) Pressing Fctn X ten times while using the Disk Manager to initialize a blank disk will cause the disk to be altered such that you cannot make a backup copy of it.

74) Try POKEV(784,X) (where X is 16 to 31) while in Mini Memory. This will change the background color of the space character.

75) A slick way to end your programs if you have Extended Basic and 32K is to use the following routine that will take you back to the title screen.

    CALL INIT :: CALL PEEK(2,A,B) :: CALL LOAD(-31804,A,B)

76) Or you may want to simply restart Extended Basic (including the auto LOAD). Try this routine out.

    CALL INIT :: CALL LOAD(-31962,255)

77) While editing a line you mistakenly hit Fctn 3 (erase) when you meant to hit Fctn 2 (insert) and the entire line is gone from sight. Not to worry, just hit Fctn 4 (clear), retype the line number, and hit Fctn X. Like magic your line has returned, ready to be edited.

78) CALL SCREEN(15) should be used if you have a black & white tv as your computer monitor. It disables the color generator and removes those vertical lines that you might have seen.

79) Here is the amount of memory you have present to program.

Console Basic	14536 bytes	
CB & disk	12448 "	
TE II	14024 "	
Extended Basic	13928 "	
EB, 32k, & disk	11840 "	plus 24448 bytes in the 32k

80) Here is my version of a screen dump to the TI (also Epson & Gemini) printer. It clocks in at 39 minutes and 20 seconds (note that you must use Extended Basic). It will take any character definition at all 768 screen locations and faithfully reproduce them on your printer!

```
100 OPEN #1:"PIO.CR" :: PRINT #1:CHR$(27);CHR$(65);CHR$(8) :: B$="0123456789
ABCDEF"
110 FOR R=1 TO 24 :: PRINT #1:CHR$(10);CHR$(13);CHR$(27);CHR$(75);CHR$(0);CH
R$(1) :: FOR C=1 TO 32 :: CALL GCHAR(R,C,A) :: CALL CHARPAT(MIN(MAX(A,32),14
3),H$)
120 C1,C2,C3,C4,C5,C6,C7,C8=0 :: FOR P=1 TO 15 STEP 2 :: X=POS(B$,SEG$(H$,P,
1),1)-1 :: Y=POS(B$,SEG$(H$,P+1,1),1)-1 :: Z=2^((15-P)/2)
130 C1=C1+Z*SGN(X AND 8) :: C2=C2+Z*SGN(X AND 4) :: C3=C3+Z*SGN(X AND 2) ::
C4=C4+Z*SGN(X AND 1) :: C5=C5+Z*SGN(Y AND 8) :: C6=C6+Z*SGN(Y AND 4) :: C7=C
7+Z*SGN(Y AND 2) :: C8=C8+Z*SGN(Y AND 1)
140 NEXT P :: PRINT #1:CHR$(C1);CHR$(C2);CHR$(C3);CHR$(C4);CHR$(C5);CHR$(C6)
;CHR$(C7);CHR$(C8) :: NEXT C :: NEXT R :: PRINT #1:CHR$(27);CHR$(65);CHR$(12
) :: CLOSE #1
```

Now for a contest. Let's see who can come up with a faster EB screen dump. This challenge goes out to all the user groups. Here is the program to use to time the screen dump. Start and stop the watch at the beeps. You can also try to come up with a shorter "byte" version - this uses 577 bytes.

```
1 CALL CLEAR :: CALL CHAR(32,RPT$("F",16)) :: CALL SOUND(1000,500,0)
100 "your routine"
1000 CALL SOUND(1000,500,0)
```

81) For those of you who would like to use a full screen editor to write your programs, you can now do so (the program is also useful in taking screen dumps from TEII BBS programs and converting them to "runable" programs). Use either TI Writer or Editor Assembler to create a program (ie - write the program like you would on paper ... making changes, moving lines, copying lines, etc. ... using all the great features of a full screen editor).

There are two rules to follow -
1) The first character(s) of each line must be a line number and have one space following it (this limits the length of each line to 80 characters).
2) Call your "text" version of the program you create "DSK1.TXT".

Load and run this program -

```
1 CALL CLEAR :: OPEN #1:"DSK1.TXT",INPUT :: OPEN #2:"DSK1.PGM",OUTPUT,
VARIABLE 163 :: ON ERROR 4
2 LINPUT #1:L$ :: S=POS(L$," ",1) :: N=VAL(SEG$(L$,1,S)) :: A=INT(N/256) ::
B=N-A*256 :: PRINT #2:CHR$(A)&CHR$(B)&SEG$(L$,S,80)&CHR$(0) :: IF EOF(1)=0
THEN 2
3 PRINT #2:CHR$(255)&CHR$(255) :: CLOSE #2 :: CLOSE #1 :: END
4 DISPLAY "'TXT' FILE BAD - TAKE A LOOK" :: RETURN 3
```

After the program runs, type "NEW". Then type "MERGE DSK1.PGM". Enter the 1st line number of your program, press "Fctn X" & "Fctn 1". Keep pressing "Fctn X" & "Fctn 1" until you have gone through all the line numbers. Then save the program under any name you like and run it!

82)

THESE ARE ALL OF THE PEEKS & POKES THAT I HAVE COME ACROSS FOR USE WITH EXTENDED BASIC AND 32K MEMORY EXPANSION (BE SURE TO DO A "CALL INIT"). THE P & Q VARIABLES ARE USED FOR "PEEK" - THE NUMBERS ARE FOR "POKE". IF YOU KNOW OF ANY OTHERS PLEASE LET ME KNOW AND I WILL ADD THEM IN.

ADDRESS , VALUE(S)	MEANING - IN EXTENDED BASIC
-28672 , P	P=0 SPEECH NOT ATTACHED P=96 SPEECH IS ATTACHED
-31744 , 0 TO 15	CONTINUATION OF LAST SOUND (0=LOUD AND 15=SOFT)
-31748 , 0 TO 255	CHANGE THE CURSOR FLASHING AND RESPONSE TONE RATES
-31788 , 160	BLANK OUT THE SCREEN (MUST PRESS A KEY TO ACTIVATE)
, 192	NO AUTOMATIC SPRITE MOTION OR SOUND
, 224	NORMAL OPERATION
, 225	MAGNIFIED SPRITES
, 226	DOUBLE SIZED SPRITES
, 227	MAGNIFIED & DOUBLE SIZED SPRITES
, 232	MULTICOLOR MODE (48 BY 64 SQUARES)
-31794 , P	TIMER FOR CALL SOUND (COUNTS FROM 255 TO 0)
-31804 , X , Y	RETURN TO THE TITLE SCREEN (USE "PEEK(2,X,Y)")
-31806 , 0	NORMAL OPERATION
, 16	DISABLE QUIT KEY (FCTN =)
, 32	DISABLE SOUND (USE NEG DUR FOR CONTINUOUS SOUND)
, 64	DISABLE AUTO SPRITE MOTION
, 128	DISABLE ALL THREE
-31808 , P , Q	DOUBLE RANDOM NUMBERS (0 TO 255) NEED "RANDOMIZE"
-31860 , 4	GO FROM EXTENDED BASIC TO CONSOLE BASIC (NEED "NEW")
, ~~8~~ 0	AUTO RUN OF DSK1.LOAD
-31866 , P , Q	END OF CPU PROGRAM ADDRESS (P*256+Q)
-31868 , 0	NO "RUN" OR "LIST" AFTER A "FCTN 4" IS USED
-31873 , 3 TO 30	SCREEN COLUMN TO START AT WITH A "PRINT"
-31877 , P	P&32=SPRITE COINCIDENCE P&64=FIVE SPRITES ON LINE
-31878 , P	HIGHEST NUMBERED SPRITE IN MOTION (0 STOPS ALL)
-31879 , P	TIMER FOR VDP INTERRUPTS EVERY 1/60 SEC (0 TO 255)
-31880 , P	RANDOM NUMBER (0 TO 99) NEED "RANDOMIZE"
-31884 , 0 TO 5	CHANGE KEYBOARD MODE (LIKE "CALL KEY(K,...)")
-31888 , 63 , 255	DISABLE DISK DRIVE (USE "NEW" TO FREE MEMORY)
, 55 , 215	ENABLE DISK DRIVE (USE "NEW" TO FREE DRIVE)
-31931 , 0	UNPROTECT EB PROGRAM
, 2	SET "ON WARNING NEXT" COMMAND
, 4	SET "ON WARNING STOP" COMMAND
, 16	SET "TRACE" COMMAND
, 64	SET "ON BREAK NEXT" COMMAND
, 128	PROTECT EB PROGRAM
-31962 , 32	RETURN TO THE TITLE SCREEN
, 255	RESTART EB W/DSK1.LOAD
-31974 , P , Q	END OF VDP STACK ADDRESS (P*256+Q)

83) Here is a listing of all the assembler books that are currently out on the the market. Check with members of the Assembler SIG if you have questic on a particular assembler topic or on one of the books.

Introduction to Assembly Language for the TI Home Computer
Ralph Molesworth
Steve Davis Publishing
139pp / $17

Learning TI 99/4A Home Computer Assembly Language Programming
Ira McComic
Wordware Publishing
331pp / $17

Fundamentals of TI 99/4A Assembly Language
M S Morley
Tab Books Publishing
310pp / $12

Beginning Assembly Language for the TI Home Computer
D & D Publishing
3177 Bellevue
Toledo, OH 43606
223pp / $20

Assembly Language Primer
John Dow Publishing
6560 Rosemoor St
Pittsburgh, PA 15217
130pp / $20

Assembly Language Tutorial
The Softies Publishing
7300 Gallagher Suite 229
Edina, MN 55435
???pp / $15

84) A simple way to check the catalog on your disk is to do the following - a) When you have the files you want on the disk, use the Disk Manager and catalog your disk not to the screen, but to "DSK1.CAT". This will create a file on your disk that contains the catalog you normally see. b) When you want to check what's on the disk while in BASIC, use this four liner (it will end in an error but you can ignore it).

```
1 OPEN #1:"DSK1.CAT"
2 INPUT #1:C$
3 DISPLAY C$
4 GOTO 2
```

c) When you add or delete files from the disk be sure and repeat step a.

85) From the Central Texas UG comes this tip - In the command mode type "your command" :: ACCEPT AT(1,1):A$ ENTER FCTN 4 "your command" (ex - CALL SCREEN(16) for a white screen) will execute and stay that way while you remain in the command mode (until an error occurs - then it will return to normal).

86) The Pittsburgh UG passes on this tip - ACCEPT AT(row,column):A$(0+row) will allow you to input up to 255 characters rather than the normal 28. Just make sure that you do not end up on an edge character (ASCII 31).

99 TIPS - BY JOHN HAMILTON

87) Happy Birthday to the TI 99/4. For a grand sum of $1150 you could buy it and the color monitor at a Team Electronics store in October of 1979. If you waited 4 years, you could get a 4A (no monitor) for $10 ($60 less the $50 rebate) at a Target store. Chances are you got yours for something in between (my first cost me $390 and the rf modulator was an extra $50). The tip for this month is to latch on to a 2nd (3rd, 4th, etc) console if you can since they are prone to wearing out (especially if you have kids). It doesn't hurt to have a spare in the closet.

88) For those of you who have modems and would like to upload files to any of the local boards, the instructions are out on the Mansion. Here's a short summary of what you need to do -

a) Get to the upload mode on the board and answer the questions
b) Use CTRL O and leave TEII
c) Go into BASIC from the title screen
d) Load the program you want to upload with OLD DSK1.progname
e) Type LIST "RS232"
f) Use FCTN = and leave BASIC
g) Go into TEII from the title screen
h) Type CTRL K (or whatever that board requires) to end the upload

The program for getting downloads was listed on page 3 of the 7/84 issue.

89) Reread the RS232 manual some evening. You may not realize that you can use SAVE, OLD, LIST with the RS232 (easy way to transfer information between 2 computers - for example: one of you from BASIC can type LIST "RS232" and the other in TEII will have the program "uploaded" to the screen). The same can be done using the PIO ports if you have both of the computers in the same room (note that you will need to rewire the parallel cable). Finally, you can LIST to DSK1 and get nice 80 column listings to merge into TI WRITER files. Experiment and let me know if come up with anything new or useful!

renewal
insert
goes
here

99 TIPS - BY JOHN HAMILTON

90) I have not heard any response from the other user groups concerning the the challenge to write a faster Epson screen dump program in Extended Basic. Where are the speed freaks that used to submit faster and faster programs for the TI 52 & 59 calculator contests run by Maurice Swinnen in the TI PPC NOTES? Surely they bought a 4A and will take up this challenge. See the July 1984 issue for the program and specifications.

Anyway, here is my version for the Okidata 82A screen dump. Note that it uses the block graphic commands resident at ASCII 128+ (not the bit map graphics on the PROM). This will give you an image that uses a full 8 1/2 by 11 sheet of paper (nice for us "older" folks to read).

```
100 OPEN #1:"PIO",OUTPUT,VARIABLE 132 :: PRINT #1:CHR$(29) :: FOR R=1 TO
22 STEP 3 :: FOR L=1 TO 8 :: L$(L)="  " :: NEXT L :: FOR C=1 TO 32 :: P$
=""
110 FOR B=0 TO 2 :: CALL GCHAR(R+B,C,A) :: A=MIN(MAX(A,32),143) :: CALL
CHARPAT(A,H$) :: P$=P$&H$ :: NEXT B
120 FOR L=1 TO 8 :: FOR M=0 TO 1 :: A,B=128 :: FOR N=1 TO 5 STEP 2 :: H$
=SEG$(P$,6*(L-1)+M+N,1)
130 A=A+SGN(POS("89ABCDEF",H$,1))*2^(N-1)+SGN(POS("4567CDEF",H$,1))*2^N
:: B=B+SGN(POS("2367ABEF",H$,1))*2^(N-1)+SGN(POS("13579BDF",H$,1))*2^N
140 NEXT N :: L$(L)=L$(L)&CHR$(A)&CHR$(B) :: NEXT M :: NEXT L :: NEXT C
:: FOR L=1 TO 8 :: PRINT #1:L$(L) :: NEXT L :: NEXT R :: PRINT #1:CHR$(30)
:: CLOSE #1
```

91) You can check out the club's modules for a month. We have the following ones available - Blackjack, Car Wars, Hunt the Wumpus, Invaders, Parsec and Video Games. See Ron Rutledge at the next software or general meeting. We would like to obtain more modules. The software committee is considering buying a few (if we can raise some funds). If you have any modules that you no longer use, the club would be most grateful for a donation. Thanks!

renewal
insert
goes
here

92) Here are the pinouts for the I/O GROM port (cartridges) as supplied by TI (no guarantees on correctness - proceed at your own risk).

1=reset	7=d5	13=d2	19=+5 volt	25=dbin/mi	31=grom ready
2=gnd	8=a15	14=a11	20=a8	26=a6	32=we
3=d7	9=d4	15=d1	21=grom sel	27=grom clk	33=grom gnd
4=cru clk	10=a13	16=a10	22=a7	28=a5	34=rom gnd
5=d6	11=d3	17=d0	23=a14/mo	29=-5 volt	35=gnd
6=cru in	12=a12	18=a9	24=a3	30=a4	36=gnd

93) Here are the pinouts for the I/O PERIPHERAL port (expansion box) with the same disclaimers as above.

1=+5 volt	9=dbin	17=a7	25=gnd	33=cru in	41=hold/iaq
2=spch sel	10=a3	18=a9	26=we	34=d7	42=d3
3=reset	11=a12	19=a15	27=gnd	35=d4	43=-5 volt
4=ext int	12=ready	20=a2	28=mbe	36=d6	44=speech
5=a5	13=load	21=gnd	29=a6	37=d0	
6=a10	14=a8	22=cru clk	30=a1	38=d5	
7=a4	15=a13	23=gnd	31=a0	39=d2	
8=a11	16=a14	24=0 3	32=memen	40=d1	

94) For those of who do not like the shape of the cursor, here is a nice Extended Basic routine from G-S Romano of the 99'ers UG. Changing the last eight values in line 40 will produce new shapes.

```
10 CALL CLEAR :: CALL INIT
20 CALL LOAD(8196,63,248)
30 CALL LOAD(16376,67,85,82,83,79,82,48,8)
40 CALL LOAD(12288,0,0,0,0,0,0,0,252)
50 CALL LOAD(12296,2,0,3,240,2,1,48,0,2,2,0,8,4,32,32,36,4,91)
60 CALL LINK("CURSOR")
```

95) TI has sent the club eight disks of public domain software over the last year. You may make free copies at the meetings. Included are Forth object (1), Forth source (2), Advanced Debugger (1), TI Writer and Multiplan updates (1), games, demos & songs (3).

96) There is a new publication out that focuses on the 4A. From what I have seen, it is a decent buy at $12/12 issues. Write to -
Super 99 Monthly (from Bytemaster Computer Services)
171 Mustang Street Sulphar, LA 70663

97) Local BBS's have come and gone faster than TI left the home computer market. Here is an active list as of 12/14/84 (all are free).

COLORAMA	277-6510	24 hr
COMPUTERLAND	270-8942	7 pm to 9 am
CSS (down till 1/1)	288-9128	6 pm to 7 am
DELTA 80	285-4531	24 hr
HALS	279-6769	8 pm to 8 am
MIDDLE AMERICA	961-8881	24 hr
OMNI	276-8688	24 hr

98) If you are looking for parts for your computer, call BNF Enterprises at 1-617-531-5774. Their latest catalog has the following -

Keyboard	$13.88	Internal power supply	$4.95
Modulator	$ 9.99	Cassette cables	$3.88

99) Well, 18 months and 98 tips later, I come to the end of "99 TIPS". It's been fun to write and I hope provided you with a little info on the marvelous 4A computer. I have available a 99/4 special - all 99 tips plus 4 extra (EB command summary, disk map, basic sort & assembler sort) - 22 pages with folder, mailed to you for $4.00. Send to John Hamilton. 4228 E Clinton. Des Moines. Iowa. 50317.

REVIEWS

This month I am going to review EXTENDED BASIC, both for those who don't have it and for those who have it but may not have used all of the features that it offers. Listed below are all the features that it has that are not found in regular BASIC.

MERGE - combines a program in memory with one on disk.
PROTECTED - prevents a program from being listed or saved.
RUN "device" - loads and runs a program.
DIM - arrays can now have up to 7 dimensions.
CALL CHAR & COLOR - allows for multiple assignments between ().
IF THEN ELSE - allows both statements and line numbers to be used.
CALL INIT - clears the 32k memory expansion.
CALL LOAD - loads an assembled program or pokes bytes in the 32K.
CALL LINK - runs an assembled program in the 32K.
CALL PEEK - shows bytes at addresses in the 32K.
CALL SAY - allows 300+ words to be used with the speech.
CALL SPGET - puts those words into a variable for later use.
LINPUT - inputs anything (even ,;") into a string variable.
ACCEPT AT - asks for input anywhere on the screen.
DISPLAY AT - shows output anywhere on the screen.
PRINT USING - formats your output with IMAGE.
IMAGE - allows for ####.## or ###,### type printing.
MAX - gives the larger of two numbers.
MIN - gives the smaller of two numbers.
RPT$ - repeats a string up to 255 positions.
PI - the constant 3.1415926.
AND, OR, XOR, NOT - alllows boolean logic comparisions.
  - another form for REM.
:: - allows multiple statements for each line number.
!@P- - turns of prescan feature, programs run sooner.
Redo - repeats last keyboard entry.
CALL name(parameter), SUB name(parameter), SUBEND, SUBEXIT - calls subroutines by name and passes data back and forth.
ON BREAK - stop or continue when a break occurs.
ON WARNING - stop, print, or continue when a warning occurs.
ON ERROR, RETURN - stop or branch when an error occurs.
CALL ERR - returns the error value in a variable.
REC - tells the current record in a relative file.
SIZE - reports the remaining unused memory.
CALL CHARPAT - returns the 16 values used to define a character.
CALL CHARSET - restores all redefined characters to turn on state.
CALL VERSION - tells version of EB module, currently 110.
CALL SPRITE - up to 28 characters that move by themselves.
CALL DELSPRITE - to delete a sprite.
CALL PATTERN - to change the design of a sprite.
CALL COLOR - to change the color of a sprite.
CALL LOCATE - to change the position of a sprite.
CALL MOTION - to change the velocity of a sprite.
CALL POSITION - tells where a sprite is on the screen.
CALL DISTANCE - tells how far a sprite is from a sprite/pixel.
CALL COINC - tells if a sprite is touching a sprite/pixel.
CALL MAGNIFY - makes a sprite twice as big.

If you have any questions on any of these features, please see me at the next meeting. If there are any features that I have overlooked, let me know and I'll include them in the next newsletter.

SECTOR 0

Hex	Dec	Meaning (of the 256 bytes in the sector)
00-09	0- 9	The disk name you assigned
0A-0B	10- 11	Number of sectors initialized (ex >0168 = 360)
0C	12	Number of sectors per track (ex >09 = 9)
0D-0F	13- 15	TI identifier - "DSK" or >445348
10	16	Copy protection (ex >20 = none, >50 = protected)
11	17	Number of tracks (ex >28 = 40)
12	18	Number of sides (ex >01 = single, >02 = double)
13	19	Disk density (ex >01 = single, >02 = double)
14-37	20- 55	not used
38-64	56-100	} This is a bit map of all the sectors on the disk
66-92	102-146	} Use depends on if the disk is SS, DS, SD, or DD
94-C0	148-192	} 1) Take each byte (45 bytes for 360 sectors)
C2-EE	194-238	} 2) Convert to bits (8 bits per byte) 3) Reverse the order of the 8 bits 4) If the bit is "0" then the corresponding sector (0 to 359) is free. If the bit is "1" then the sector is used.
65,93	101,147	}
C1,EF	193,239	} not used
F0-FF	240-255	}

SECTOR 1

Hex	Dec	Meaning
00-01	0- 1	Tells sector of 1st "alphabetic" file directory
02-03	2- 3	Tells sector of 2nd "alphabetic" file directory
:	:	("alphabetic" means that if the filenames were
:	:	sorted this would be the 1st, 2nd, etc filename)
FC-FD	252-253	Tells sector of the 127th "alphabetic" file dir.
FE-FF	254-255	0000 is always after the last filename (if there was only one file then 0000 would be at >02-03)

SECTORS 2-33

Hex	Dec	Meaning
00-09	0- 9	The file name that you used
0A-0B	10- 11	not used
0C	12	File Type bit 0 - 0=fixed 1=variable length bit 4 - 0=none 1=write protected bit 6 - 0=display 1=internal format bit 7 - 0=data 1=program file
0D	13	Number of records per sector (n/a for program)
0E-0F	14- 15	Number of sectors per file
10	16	End of file offset in last sector (n/a for fixed)
11	17	Record size (n/a for program)
12-13	18- 19	Number of records per file (n/a for program) note - the bytes are reversed (ex >0102 = >0201)
14-1B	20- 25	not used
1C	26	Sector where file is located } repeats as needed
1D-1E	27- 28	Number of sectors following } to use any sector note - the bytes are flipped (ex >12 = >2001>

```
**********************************************************************
*                          A  SHORT  PROGRAM                         *
**********************************************************************
```

Time to do a little assembly language programming. Listed below is the sort routine that I gave in the August issue.

```
Memory  Label  Op-Code  Operand(s)  Comment (equivilant in basic)
======  =====  =======  ==========  ============================
 7D0C          CLR      R0          This area sets up the program
 7D0E          LI       R1,>2       to receive the parameters that
 7D12          BLWP     @NR         are passed from the Basic line
 7D16          BLWP     @FP         that calls the assembler sort
 7D1A          DATA     FI          routine.  It's called by -
 7D1C          MOV      @FA,R2
 7D20          DEC      R1          CALL LINK("SORT",A(),B)

 7D22          MOV      R2,R3       1000 C=INT(B*.75)
 7D24          SLA      R3,>2
 7D26          S        R2,R3
 7D28          SRL      R3,>2

 7D2A    L1    MOV      R1,R4       1001 FOR D=1 TO B-C
 7D2C          MOV      R2,R5
 7D2E          S        R3,R5

 7D30    L2    MOV      R4,R6       1002 E=D

 7D32    L3    MOV      R6,R0       1003 IF A(E)<=A(C+E) THEN 1009
 7D34          BLWP     @NR
 7D38          LI       R7,FA
 7D3C          LI       R8,AR
 7D40          MOV      *7+,*8+
 7D42          MOV      *7+,*8+
 7D44          MOV      *7+,*8+
 7D46          MOV      *7,*8
 7D48          A        R3,R0
 7D4A          BLWP     @NR
 7D4E          BLWP     @FP
 7D52          DATA     FC
 7D54          MOVB     @ST,R0
 7D58          ANDI     R0,>4000
 7D5C          JEQ      L9

 7D5E          MOV      R6,R0       1005 A(E)=A(C+E)
 7D60          BLWP     @NA

 7D64          LI       R7,AR       1004 F=A(E)
 7D68          LI       R8,FA
 7D6C          MOV      *7+,*8+
 7D6E          MOV      *7+,*8+
 7D70          MOV      *7+,*8+
 7D72          MOV      *7,*8

 7D74          A        R3,R0       1006 A(C+E)=F
 7D76          BLWP     @NA

 7D7A          S        R3,R6       1007 E=E-C
```

```
*****************************************************************
*                        A  SHORT  PROGRAM - cont.              *
*****************************************************************

Memory  Label  Op-Code  Operand(s)   Comment (equivilant in basic)
======  =====  =======  ==========   ============================
 7D7C          JGT      L3           1008 IF E>0 THEN 1003

 7D7E   L9     INC      R4           1009 NEXT D
 7D80          DEC      R5
 7D82          JGT      L2

 7D84          SRL      R3,>1        1010 C=INT(C/2)

 7D86          JGT      L1           1011 IF C THEN 1001

 7D88          MOVB     R3,@ST       1012 RETURN
 7D8C          B        *R11

 7D8E   NR     EQU      >6044        Note - this area defines all
 7D8E   NA     EQU      >6040        of the labels that are used
 7D8E   FP     EQU      >601C        in the program (note that the
 7D8E   FI     EQU      >1200        first five would need differ-
 7D8E   FC     EQU      >0A00        ant addresses if you were to
 7D8E   FA     EQU      >834A        use the Editor Assembler or
 7D8E   AR     EQU      >835C        Extended Basic modules to run
 7D8E   ST     EQU      >837C        this program).

 7D8E          AORG     >7FE8        Note - this area defines where
 7FE8          TEXT     'SORT  '     the sorting program is located
 7FEE          DATA     >7D0C        in the Mini Memory module.
 7FF0          END                   Change for EA or EB modules.
```

Here is how the workspace registers are used for the program -

```
R0 = subscript & work area          R5  = B-C
R1 = paramenter & constant 1        R6  = E
R2 = B                              R7  = indirect from address
R3 = C                              R8  = indirect to address
R4 = D                              R11 = GPL return address
```

The program is meant to be run in the Mini Memory module. It may be assembled at any address after >7D0C (be sure and change the ref/def table). If you want to use it with the Editor Assembler or Extended Basic modules, you will need the 32K expansion (the program will go in the 8K region) and the addresses for five of the subprograms (look in the Editor Assembler manual for these).

The advantages of the assembly program are two fold. First is the increase in speed it offers over basic (4.2 vs 22.8 sec/100 #) and this can be speeded up 15 times by using the 32K expansion, since you can directly access the array in the 24K region and don't have to make time calling accesses to the VDP RAM.

The second advantage is in the amount of space each program takes. The basic version eats up 179 bytes while the assembly version needs only 130 bytes. The basic variables use 75 bytes and the assembly a mere 18. Overall count - basic @ 254 bytes and the assembly @ 148 bytes (only 58% of the basic)!