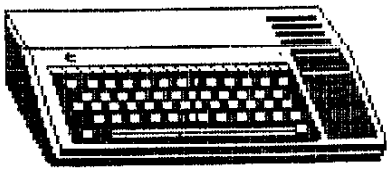
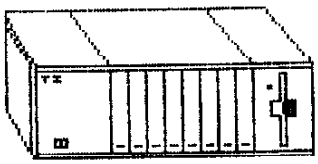
NEW JUG NEWS



NEW JERSEY USERS GROUP



Vol. 4 No. 11*Monthly Publication of the New Jersey Users Group*DECEMBER, 1985

****

MEETING

****

DECEMBER 9 MONDAY 7:00

7:00 - 8:00 BASIC SIGS WILL MEET.

8:00 GENERAL MEETING -- HARRY POTTER--Where is NEW JUG going? Where do you want it to go?

MARV SHULDMAN--What's new in hardware?

BOB COSTELLO--What's new in software?

RICH ALFONZO--******ELECTIONS******

****

The New Jersey Users Group meets on the second Monday of each month in the Metuchen Library.

****

OFFICERS

President.....	Steve Citron..	686-5619
Vice-President.....	Bob Costello..	663-4512
Secretary.....	Carol Sudol...	494-3781
Treasurer.....	Marv Shuldman.	821-8158
Newsletter Editor....	Mel Gary.....	828-5407
Technical Advisor....	Harry Potter..	625-1818
Software Review.....	Dan Ferst.....	536-4255
Assembler SIG.....	Jay Holovacs..	356-3150
Forth SIG.....	Bob Costello..	663-4512
	Harry Potter..	625-1818
Basic SIG.....	Bob Haefeli...	572-2828
	Mel Gary.....	828-5407

****

SUBSCRIPTION FREE WITH PAID MEMBERSHIP, TO USERS GROUPS AND SELECTED VENDORS

****

DECEMBER 1985

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
1	2	3	4	5	6	7
8	9 GENERAL MEETING	10	11	12	13	14
15	16	17 STEERING COMMITTEE	18	19	20	21
22	23	24	25	26	27 NEWSLETTER DEADLINE	28
29	30	31				
				*BASIC GROUP BEFORE GEN'L MEETING		

HELP COLUMBIA

I am sure that you are aware of the disasterous volcano eruption that occurred near Bogata, Columbia, killing over 20,000 persons and causing another 20,000 to be homeless. Well, one our new members, Tom Stokes, and his wife Adela are actively involved in raising funds to help the Coluabian victims. They are using their TI 99/4A and the Navarone DataBase Manager to keep track of the funds that they receive and to send thank you letters to contributors.

Any one wishing to contribute to the Columbian relief fund should make the check out to Damnificados-Columbia and send it to:

Columbian Relief Fund
c/o Adela Stokes
27 Glen Oak Dr.
Middletown, NJ 07748

PASCAL-PART II

By Ronald J. Hartranft

The manual for the p-code card contains brief descriptions of the commands available immediately after system initialization: Edit, Run, File, Compile, Link, Xecute, Assemble, Halt, Initialize, User restart, and Monitor. Any of these commands can be given by just pressing the first letter of the command. Pressing the "?" will allow you to step through all choices in case you don't remember a particular command. But you can press "F" for file even when it is not listed on the system promptline. For the remainder of this article, I want to concentrate on the file-related commands. These are similar to those available in the disk manager module, but there is a great deal of flexibility in the UCSD p-System.

Disks to be used with the UCSD p-System may be initialized with the disk manager module or by using the utility program, DFORMAT. The procedure is given at the end of this article. Note that disks used in the UCSD p-System appear to the disk manager module to contain a single file called PASCAL which completely fills the disk. Whereas the disk manager module measures file size in sectors (256 bytes per sector, 9 sectors per track, 40 sectors per side), the UCSD p-System uses blocks (512 bytes per block, 180 blocks per side). File names may be as long as 15 characters in the UCSD p-System. Only 10 are allowed in TI BASIC.

The Zero command removes the old directory entries on a disk and allows you to start fresh. The UCSD p-System can maintain two copies of the directory so that a back-up is available. If you take advantage of this feature, your first file will start at block 10. List ("L") shows the files on a PASCAL disk; ("E") is similar but gives more information. Change ("C") can be used to change either disk names of files names. Note that disk names include a colon (:) as the last character, but that it's not always displayed.

If you haven't already made yourself a back-up copy of the PASCAL disks, you can do that with Transfer command. The simplest command after pressing ("T") and getting the prompt, "Transfer?", is ("#4,#5"). This destroys the contents of the disk in drive 2 (#5) and copies the contents of the disk in drive 1 (#4) to drive 2. If you have only one drive, the appropriate command is ("#4,#4"). The system prompts you to shuffle the disks in and out when necesary. Disk names can also be used as in ("DN1:,DN2"), which destroys the contents of the disk named DN2: and copies the contents of the disk named DN1: to DN2:. You can add the contents of one disk to the contents of another by entering ("DN1:=,DN2:$"). The ("=") sign is a wild card representing any number of characters (any file name) and the ("?") sign means "use the same file names on DN2:. Drive numbers may also be used in place of DN1: and DN2:.

The Transfer command can also be used to make a copy of a file under a different name using ("DN1:OLD,DN2:NEW"). DN1 and DN2 may be the same but if they are, OLD and NEW must be different file names. Transfer can also be used to print a file to either device #6 or #8, the RS232 ports. It can also be used to send files from one computer to another.

Files can be deleted using the Remove command. Note that the prompt after typing the file name asks if the directory should be updated. The file is not deleted until the directory is updated. So if you reply "no", the file (or files) is not deleted. In fact, the file is not erased at all. Only the directory entry is deleted. If the blocks occupied by the file are not overwritten, it is possible to recreate the directory entry and retrieve the file.

REPRINTED FROM THE MARCH 1985 ISSUE OF THE LEHIGH 99'ER COMPUTER GROUP NEWSLETTER

TI-EDITOR

By Tom Kennedy
(d/l from CompuServe by ael gary)

The Text Editor is the area in which you create your document, and contains a number of commands and functions that perform various tasks on the text. It is these features that separate a Word Processor from a typewriter.

First, let's look at the command line. That's the line at

the top of the screen visible when you're in the command mode. There are seven commands shown and sixteen sub-commands that are options of the main seven. The commands are selected by typing only the letters that are that are capitalized. At this point point, you can access any of the sub-commands from the main command menu. In other words, to ShowDirectory (which is a disk catalog) you would enter the command mode, (FCTN 9), and either type "F" for files, and "SD" for ShowDirectory, or just type "SD" immediately. This feature saves a lot of time and keystrokes.

EDIT This command simply enters you into the text-edit mode in which text is created.

TABS When you hit "T", the top part of your text is shown with a scale across the top showing the current tabs and margins. Changes are made by simply typing over existing entries with the appropriate symbol (L,R,T, or I).

FILES Allows you to work with your text file as a whole. To Load, Save, Delete, Print, Purge, or ShowDirectory. "PF" for print file is not what you'll get when you print out through the text formatter, it just prints a "hard copy" of the whole file, just as you see it on the screen. It doesn't print with any of the modifications made by the format commands (more on those later). "PF" is useful for making a fast copy of a long letter, or whatever, in order to check for errors without having to scroll back and forth or up and down. Purge simply erases the file from memory to prepare for a new entry. It is similar to the "NEW" command in BASIC.

LINES This allows you to work with whole lines or groups of lines by moving them to somewhere else in the text, copying to somewhere else and leaving the original intact, to delete groups of lines, or to quickly move the cursor to some line in the text with the ShowLines option.

SEARCH Gives you the option of ether the FindString routine or the ReplaceString routine. FindString will move the cursor to the first and/or each successive use of the word string you give. ReplaceString searches the text for a given string and replaces all or one occurrence with the new string. This is great for correcting a repetitive spelling error.

RECOVEREDIT A failsafe repair in case the text buffer was purged in either the File or Quit command. It will pull back everything but the first line and restore the file. I guess the loss of the first line is the penalty paid for accidentally erasing a file, which can't be done very easily.

QUIT As the name implies, blows it all apart and leaves you with the title frame. But before it goes, all open files are closed (such as to disk or printer) so no data is lost. Fortunately, it first gives you the option of saving your file (in case you forgot to do that already) or just purging the file and going back to the edit mode. But if you really want to quit, you type "E" for Exit and it shuts down.

Now let's go over the keyboard. To accurately represent the many keystrokes, there is a file in the TI-Forum Data Library-2 called TWKEYS.HLP that list the keystrokes and their dupicates. It is recomended that you get a copy of that file to proceed.

TI-WRITER makes extensive use of the FCTN and CTRL keys and uses every possible function of the top line of keys (the numbers). There are also many functions that have duplicate methods of keystrokes to activate them. For instance, to enter the command mode, you either press FCTN 9 or CTRL C. The reason for this duplication is to allow you to choose which is easiest to use depending on where your fingers are at. The problem though, is that it can be very confusing trying to remember the fifty different key combinations that activate the thirty functions. A better method is to just pick which keys you're going to use for what function and ignore the rest. It's a good idea to use the number line keys for anything shown on the overlay strip and just memorize the few functions hidden down in the keyboard.

FUNCTION	KEY	ALSO
OOPS!	CTRL-1	/ CTRL-Z
Del Char	FCTN-1	/ CTRL-F
Reformat	CTRL-2	/ CTRL-R
Ins Char	FCTN-2	/ CTRL-G
Screen Color	CTRL-3	
Del Line	FCTN-3	/ CTRL-N
Next Paragraph	CRTL-4	/ CTRL-J
Roll Down	FCTN-4	
Dupe Line	CTRL-5	
Next Window	FCTN-5	
Last Paragraph	CTRL-6	/ CTRL-M
Roll up	FCTN-6	/ CTRL-B
Word Tab	CTRL-7	/ CTRL-W
Tab	FCTN-7	/ CTRL-I
New Paragraph	CTRL-8	
Ins Line	FCTN-8	/ CTRL-O
New Page	CTRL-9	
Command/Escape	FCTN-9	/ CTRL-C
Word Wrap	CTRL-0	
Line Numbers	FCTN-0	
Quit	FCTN-=	
Back Tab	CTRL-T	
Beginning of Line	CTRL-V	
Del. End of Line	CTRL-K	
Home Cursor	CTRL-L	
Left Mrgn Release	CTRL-Y	

Now, if you're still following along you may be quite confused with this onslaught of information. The point is, you can't learn all of this in one sitting, but after using TI-WRITER for a while you start to pick things up as you need them. Rest assured, you do spend the majority of

your time typing. The purpose of most of the functions are to manipulate the text which is already in the file.

To review, in the Command mode we can choose between Edit, Tabs, Files, Lines SearcH, RecoverEdit, or Quit. As sub-commands of those seven, we can choose Load File, Save File, Print File, Delete File, Purge, ShowDirectory, Move Lines, Copy Lines, Delete Lines, Showlines, FindString, ReplaceString, or Exit.

ASSEMBLY AND C

By Jay Holovacs

The big excitement this month of course is the new C compiler for the TI. The implementation available, of course is a bare-bones version, but the beauty of C is its extendability, and the unlimited library capabilities. (I have been working on an assembly/c bitmap graphics library and hope to have it available at the next meeting.)

One great advantage of c is that it ultimately produces true 9900 native code in a relatively friendly environment, with significant performance advantages over interpreted pseudo-code (Pascal) or threaded code (Forth). This, along with its true recursive, structured nature and speed capabilities promise to open a new door for software development by the amateur TI user.

C is a systems programming language, one that could be used for writing other compilers and interpreters, operating systems, some types of real time data acquisition and process control. While virtually any program could be written in any language if one tries hard enough, c would be less convenient than Fortran or even Basic for a scientific number crunching routine, or less than Cobol for accounting software.

How does this relate to assembly language? Well, unlike languages like Pascal which try to scrupulously avoid non structured items like assembly language, c developed as a systems programming language which would typically turn to assembly language drivers and interfaces for all the unpleasantries of I/O etc.

Small c (both in the TI and 8080/8085/Z80 implementations) has simple mechanism for direcly inserting assembly language code into c object code. This is particularly easy to use, because the intermediate output of the compiler is assembly source, and it is easy to understand exactly what is happening to your code in the program. (This feature normally doesn't exist on large scale systems because of their more advanced features, when writing code for transport be sure to write the assembly routine to emulate a standard function on other implementations.)

The following information is based on the documentation supplied and reverse engineering of c output including some compiled with portions of my assembly inserted using the #asm #endasm construction. The information appears to be correct, but I cannot guarantee total accuracy.

C uses a stack in lower RAM; with register 14 always pointing to the top of the stack. Inasmuch as 9900 does not have a designated stack pointer as such, register 15 is the start of routine which places contents in register 8 on the stack each time it is call.

The stack is heavily used by the compiler; both for values as well as addresses for many of commands. For example, the statement 'int a' simply causes the stack pointer to decrement by two, with the compiler's internal tables remembering that 'a' is stored there. During compilation, the compiler maintains a record of the location of each variable on the stack and uses this in writing code. Since the completed code then has all the address information it needs, this table is not longer needed after compilation and is discarded. This is one of the several reasons why a compiled program is faster than an interpreted one, an interpreter is constantly looking up values in internal tables.

If the statement a=45 is encountered, the compiler copies the current stack pointer into R8, increments enough to make R8 indicate the address of 'a' on the stack, and PUSHes; therefore the ADDRESS of 'a' is now at the top of the stack. Next the value of '45' is placed in R8 (here used as kind of an accumulator), the 'a' address is copied into R9, R14 is auto-incremente (effectively removing this address from the top of the stack) and the contents of R8 are moved into the addres pointed to by R9.

Why all this trouble when one could simply use LI R8,15 ? Well, that's what a human programmer (or perhaps a highly optimizing large system machine) would have done, however small c is limited to using the same procedure for all equates, whether there are variables, constants or expressions in the right hand side of the equation. Therefore it uses this generic approach because it lacks the extensive rule based knowledge (and very large size) that an advanced compiler would have.

**Note that the stack grows downward, therefore the 'top' of the stack is actually the LOWEST numerical address.

Therefore, assembly language will still have use in c, both for special routines which are not addressed by c convention and adding new features that are speed and size sensitive (floating point perhaps?).

Normally, any routine we supply in assembly language will be part of a function (not part of 'main').

Any global values (defined outside of main) can be interpreted by label (MOV @LABEL,0). Values (including array pointers) can be passed into the function in the normal manner as a local parameter. The values are placed sequentially on the stack with the last value at the 'top'; and then a return address (1 word) is placed on the very top. A function always returns whatever value is in R8 at the end of the function. This can either be discarded or used as required. Below is an example of passing a parameter and returning a value:

```
peek(adrs)
 char *adrs  /*using a char pointer so value can point
             to a byte rather than a word boundary*/
 $asm
  MOV @2(R14),R0 GET THE ADDRESS FROM STACK (2ND WORD)
  CLR R8
  MOVB *R0,R8 COPY VALUE INTO R8
  SWPB R8 GET INTO LOW BYTE
 $endasm
 /*registers 0 thru 7 are available for general use    and
do not need to be restored before returning */
```

**NOTE: I used the R convention in indicating registers (I find myself less likely to make a mistake) however since the c output does not use them, you must either remember to use the R option whenever assembling code containing your own routines or eliminate them in your source.

In all but the simplest cases (as above), it is advisable to use BLWP and a separate workspace for the assembly routine. Remember to copy R14 if you need to access the stack. Remember also to branch to the $endasm when the routine is finished.

```
  function()
 $asm
  BLWP @ROUTN GO TO THE ACTUAL ROUTINE
  B  @BACK GET TO THE PHYSICAL END OF ROUTINE TO RETURN TO
C .
ROUTN DATA WKSPC,ENTRY (POINTERS)
WKSPC BSS 32 SET ASIDE DEDICATED WORKSPACE

ENTRY MOV @28(R13),R11 COPIES VALUE OF STACK POINTER  INTO
NEW WORKSPACE
(PROGRAM. . .
  RTWP (END OF PROGRAM)

BACK NOP

 $endasm
```

After discussion with Harry Potter, we have decided to combine c, Forth, Pascal and assembler into a single Advanced Language SIG because of the overlap of interest and techniques required. In future, assembly language people will meet on the Forth night.

ELECTIONS

The following slate of nominees was adopted at the November meeting of NJUG and the election of officers for 1986 will take place at the December meeting. The constitution provides for absentee ballots for those unable to attend the meeting. Such ballots should be sent to:

Mel Gary 49 Pine Grove Ave.
Somerset, NJ 08873

Needless to say, absentee ballots must be recieved prior to the December meeting.

OFFICE	PERSON(S)	VOTE(X)
President	Steve Citron	_____
	WRITE IN_____	
Vice President(s)	John Bonito	_____
	Bob Costello	_____
	Mel Gary	_____
	Bob Guellnitz	_____
	WRITE IN_____	
	_____	
	_____	
	_____	
Secretary	Carol Sudol	_____
	WRITE IN_____	
Treasurer	Marv Shuldman	_____
	WRITE IN_____	
3 At-Large	Randy Evans	_____
	Dave Green	_____
	Harry Potter	_____
	WRITE IN_____	
	_____	
	_____	

WHAT'S ON FIRST

BY Dan Ferst

Hi! In my last article I explained how to use the mini-memory to load basic programs that are too large from disk. Now I want to talk about a fabulous program I downloaded from TEXNET which will enable you to use the disk drive with your Mini-memory. In fact it can be used to load the line by line assembler to disk. I'm placing a copy of the program in the New JUG library, but for those of you who are impatient or who cannot for one reason or another get to the meetings, I am going to list the program so that you may key it in yourself.

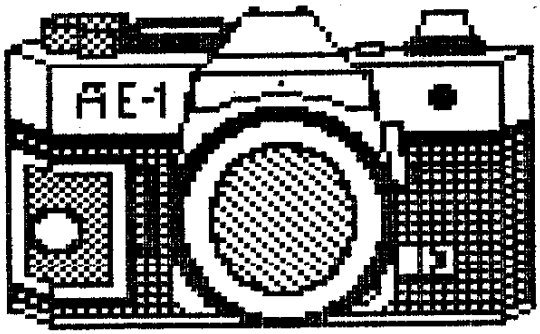
```
100 REM ********************
110 REM * Mini-Mem program *
120 REM * to linking loader*
130 REM * acceptable disk *
140 REM * file conversion *
150 REM * *
160 REM * J.  Burkett Apr 85*
170 REM ********************
180 REM
190 CALL SCREEN(8)
200 CALL CLEAR
210 AST$=" ************************"
220  PRINT AST$:"  *";TAB(25);"*":"  * Mini-Mem Program to
*"
230 PRINT "  *";TAB(25);"*";;"  * Disk File Converter *"
240 PRINT "  *";TAB(25);"*";"  * John Burkett Apr 85 *"
250 PRINT "  *";TAB(25);"*":AST$: : : : : : :
260 H$="0123456789ABCDEF"
270 D$=SEG$(H$,1,10)
280 GOSUB 1860
290 PRINT "Need instructions?"
300 CALL SOUND(150,1400,0)
310 CALL KEY(0,K,S)
320 IF S=0 THEN 310
330 IF (K<>ASC("Y"))*(K<>ASC("y"))THEN 360
340 GOSUB 2390
350 GOTO 370
360 IF (K<>ASC("N"))*(K<>ASC("n"))THEN 300
370 CALL PEEK(28700,N1,N2,N3,N4)
380 FFAMM=N16+N2
390 LFAMM=N3&+N4
400 GOTO 420
410 PRINT "Entry error!"
420 PRINT :"Enter beginning of program:"
430 INPUT " ":S$
440 IF S$="" THEN 410
450 GOSUB 1560
460 IF (DECADR<28952)+(DECADR)32750)THEN 410
470 BOP=DECADR
480 GOTO 500
490 PRINT "Entry error!"
500   PRINT   :"Enter   end   of   program:":"Null
(default)=";FFAMM-2
510 INPUT " ":S$
520 IF S$()"" THEN 550
530 DECADR=FFAMM-2
540 GOTO 560
550 GOSUB 1560
560 IF (DECADR<BOP+2)+(DECADR)32750)THEN 490
570 FFAMM=DECADR+2
580 EOP=DECADR
590 GOTO 610
600 PRINT "Entry error!"
610   PRINT :"Enter  low  end  of  REF/DEF":"table,  null
(default)=";LFAMM
620 INPUT " ":S$
630 IF S$()"" THEN 660
640 DECADR=LFAMM
650 GOTO 670
660 GOSUB 1560
670 IF (DECADR<EOP+2)+(DECADR)32766)THEN 600
680 LFAMM=DECADR
690 REM
700 PRINT :"Enter name of file to"
710 INPUT "create: ":FN$
720 OPEN #1:FN$,FIXED 80,OUTPUT
730 ADDR=28700
740 GOSUB 1370
750 DATA$="00000        9"&ADDR$
760 ADDR=FFAMM
770 GOSUB 1370
780 DATA$=DATA$&"B"&ADDR$
790 ADDR=LFAMM
800 GOSUB 1370
810 DATA$=DATA$&"B"&ADDR$
820 ADDR=BOP
830 GOSUB 1370
840 DATA$=DATA$&"9"&ADDR$
850 FOR MEM=BOP TO EOP STEP 2
860 IF LEN(DATA$)<70 THEN 880
870 GOSUB 1210
880 CALL PEEK(MEM,N,X)
890 GOSUB 1450
900 DATA$=DATA$&"B"&N$
910 N=X
920 GOSUB 1450
930 DATA$=DATA$&N$
940 NEXT MEM
950 REM
960 REM Convert REF/DEF table
970 IF LEN(DATA$)<70 THEN 1010
980 MEM=LFAMM
990 GOSUB 1210
1000 GOTO 1040
1010 ADDR=LFAMM
1020 GOSUB 1370
1030 DATA$=DATA$&"9"&ADDR$
1040 FOR MEM=LFAMM TO 32766 STEP 2
1050 IF LEN(DATA$)<70 THEN 1070
1060 GOSUB 1210
1070 CALL PEEK(MEM,N,X)
1080 GOSUB 1450
1090 DATA$=DATA$&"B"&N$
1100 N=X
```

```
1110 GOSUB 1450
1120 DATA$=DATA$&N$
1130 NEXT MEM
1140 IF LEN(DATA$)<6 THEN 1160
1150 GOSUB 1210
1160 PRINT #1:": TI5270 J.  Burkett APR 85"
1170 CLOSE #1
1180 STOP
1190 REM
1200 REM Generate checksum char & output record
1210 ADDR=0
1220 FOR X=1 TO LEN(DATA$)
1230 ADDR=ADDR+ASC(SEG$(DATA$,X,1))
1240 NEXT X
1250 ADDR=ADDR+ASC("7")
1260 ADDR=65536-ADDR
1270 GOSUB 1370
1280 DATA$=DATA$&"7"&ADDR$&"F"
1290 PRINT :DATA$
1300 PRINT #1:DATA$
1310 ADDR=MEM
1320 GOSUB 1370
1330 DATA$="9"&ADDR$
1340 RETURN
1350 REM
1360 REM Cnvrt 2 byte hex addr to ASCII,  enter  as  ADDR,
exit as ADDR$
1370 N=INT(ADDR/256)
1380 GOSUB 1450
1390 ADDR$=N$
1400 N=ADDR-N6
1410 GOSUB 1450
1420 ADDR$=ADDR$&N$
1430 RETURN
1440 REM Cnvrt a hex byte to ASCII, enter as N, exit as N$
1450 E=INT(N/16)
1460 D=N-E
1470 E=E+48
1480 IF E<58 THEN 1500
1490 E=E+7
1500 D=D+48
1510 IF D<58 THEN 1530
1520 D=D+7
1530 N$=CHR$(E)&CHR$(D)
1540 RETURN
1550 REM
1560 REM Convert address S$ to a dec numeric
1570 IF SEG$(S$,1,1)<>">" THEN 1750
1580 REM Value entered is in hex
1590 S$=SEG$(S$,2,LEN(S$)-1)
1600 IF LEN(S$)>4 THEN 1730
1610 IF LEN(S$)>3 THEN 1640
1620 S$="0"&S$
1630 GOTO 1610
1640 DECADR=0
1650 FOR N=1 TO 4
1660 DIG=POS(H$,SEG$(S$,N,1),1)
1670 IF DIG=0 THEN 1730
1680 DECADR=DECADR+((DIG-1) (4-N))
1690 NEXT N
1700 DECADR=INT(DECADR/2)*2
1710 RETURN
1720 REM Error return
1730 DECADR=0
1740 RETURN
1750 REM Value entered is in decimal
1760 FOR N=1 TO LEN(S$)
1770 IF POS(D$,SEG$(S$,N,1),1)=0 THEN 1830
1780 NEXT N
1790 DECADR=VAL(S$)
1800 DECADR=INT(DECADR/2)*2
1810 RETURN
1820 REM Error return
1830 DECADR=0
1840 RETURN
1850 REM
1860 REM Redefine lowercase characters
1870 CALL CHAR(97,"0000007008384874")
1880 CALL CHAR(98,"0040407844444478")
1890 CALL CHAR(99,"0000003844404438")
1900 CALL CHAR(100,"0004043C4444443C")
1910 CALL CHAR(101,"0000003844784038")
1920 CALL CHAR(102,"0018242070202020")
1930 CALL CHAR(103,"0000043844384078")
1940 CALL CHAR(104,"0040407844444444")
1950 CALL CHAR(105,"0010003010101038")
1960 CALL CHAR(106,"0008007808084830")
1970 CALL CHAR(107,"0040404850704844")
1980 CALL CHAR(108,"0030101010101038")
1990 CALL CHAR(109,"0000007854545454")
2000 CALL CHAR(110,"0000005824242424")
2010 CALL CHAR(111,"0000003844444438")
2020 CALL CHAR(112,"0000007844784040")
2030 CALL CHAR(113,"0000003844544834")
2040 CALL CHAR(114,"0000005864404040")
2050 CALL CHAR(115,"0000003C40380478")
2060 CALL CHAR(116,"0010381010101408")
2070 CALL CHAR(117,"0000004848484824")
2080 CALL CHAR(118,"0000004044282810")
2090 CALL CHAR(119,"0000004454545428")
2100 CALL CHAR(120,"0000004028102844")
2110 CALL CHAR(121,"0000004424181060")
2120 CALL CHAR(122,"0000007C0810207C")
2130 RETURN
2140 REM
2150 REM Instructions DATA
2160  DATA  ,,,,,MM>DISK  converts data in the,Mini-Memory
RAM into a
2170 DATA DIS/VAR 80 disk file,suitable for input by the
2180  DATA  Linking  Loader.   MM>DISK,creates  absolute
addresses
2190  DATA and data.  It will function,correctly with only
the
2200 DATA Mini-Memory on line.,,,,,,,,
2210 DATA The following 3 address,groups are saved:
2220 DATA ,1.  2 words at 28700,(>701C).  These are the
2230 DATA control values PGMM & LPMM, ' se MM  manual  pg
74).
2240 DATA ,2.  The Beginning of the,Program (BOP) thru the
End
```

```
2250 DATA of the Program (EOP).  BOP,must be entered.  EOP
may be
2260 DATA entered or defaulted to,FFAMM-2.
2270 DATA ,3.  The REF/DEF Table (RDT).,The low end may be
entered
2280 DATA or defaulted to LFAMM.  The,high end  is  always
32766,(>7FFE).,,
2290  DATA  ,When  the  EOP  and RDT low end,"are entered,
these values"
2300 DATA wil be stored as FFAMM &,LFAMM (>701C  &  >701E)
when
2310  DATA  the  program  is  reloaded.,Addresses may  be
entered
2320 DATA either as decimal or hex,values.  Preceed a  hex
value
2330  DATA  "with  the  '>'  symbol, e.g.,",",28952 is dec,
>7118 is hex."
2340 DATA All addresses are validated.,,Allowable  address
ranges:
2350 DATA ,"BOP low, high",28952 (>7118) 32750 (>7FEE)
2360 DATA "EOP low, high",BOP+2          32750 (>7FEE)
2370 DATA "RDT low, high",EOP+2          32766 (>7FFE),
2380 REM
2390 REM Display instructions
2400 FOR PAGE=1 TO 3
2410 CALL CLEAR
2420 FOR LINE=1 TO 22
2430 READ M$
2440 PRINT M$
2450 NEXT LINE
2460 GOSUB 2510
2470 NEXT PAGE
2480 RETURN
2490 REM
2500 REM *KEY-CON*
2510 PRINT " Press any key to continue"
2520 CALL SOUND(150,600,5)
2530 CALL KEY(3,K,S)
2540 IF S=0 THEN 2530
2550 RETURN
```

!!!!!1986 MEMBERSHIP DUES ARE DUE IN JANUARY!!!!!

AFTER FEBRUARY, ONLY PAID MEMBERS WILL RECEIVE NEW JUG NEWS



DECEMBER 8 - MICROCOMPUTER SHOW IS

ASPEN HOTEL MANOR - PARSIPPANY 9 - 5

TIPS FROM THE TIGERCUB

#27

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 available as a full disk of 50 programs, routines and files for just $15.00 postpaid.

>>>>>>NOW AVAILABLE<<<<<<<<<

Tips from the Tigercub VOLUME 2 The entire contents of Tips Nos. 15 through 24, with 64 routines and files, also $15.00 postpaid.

Or both for $27 postpaid.

>>>>ALSO NOW AVAILABLE<<<<<<

New Catalog #6, for $1 which is deductable from your first order. Describes 140 original programs for only $3 each (plus $1.50 per order for casette or disk, package and postage).

If you have my previous catalog, the following are now available in Extended Basic versions - Fast Addition Practice, Submarine Hunt, Rithmatik, Wawaland (also now available in Basic with Speech), Long Division Cryptograms, Miss Spell, Scrambulation, Bargraffer, Squinch, Dry Gulch, Name That Tune, Scrum, Midnight Trail, Nimbo, Kindertimes, Optical Illusion, Bazoo, Synonymy, Speeder Reader, Changeroo, Glunk, Fraction Math, Three Buckets Puzzle, Roman Numbers, Match A Patch, Kinderminus, I & E Spelling, Casting Out Nines, Haunted Graveyard, Spelling Teecher, Homonymy, Antonymy, Old -Timer Puzzle, Ten Thousand Sights, Mechanical Aptitude Test, Junior Speeder Reader, and Bars and Balls.

Due to reduced prices for disks and mailers, the PPM charge is now $1.50 for either disk or casette - BUT PLEASE BE SURE TO SPECIFY WHICH!

And my best seller - NUTS & BOLTS, a full disk of 100 (yes, I said 100) utility subprograms in MERGE format, ready for you to merge into your own programs. 13 type fonts, 14 text display routines, 9 wipes, 8 pauses, 3 programming aids, 9 data saving and reading routines, 5 graphics routines, 4 time and date, 6 music, 12 sorts and shuffles, 2 printer aids, 4 key and joystick, 4 math, 2 protection and 7 miscellaneous, plus a tutorial on subprograms. With documentation, example of using each subprogram. All for only $19.95 postpaid.

I have been receiving several requests to publicize freeware which is sometimes good but sometimes of doubtful usefulness, quality, originality or even legality! And "Freeware! Send $10 and initialized disk" is not freeware, it's somebody trying to get a free ad! So - no more freeware mentions! I am also not going to mention commercial products - after all, I'm publishing this at my own expense to promote my own software! However, I do owe a mention to Larry Hughes of Quality 99 Software, because in Tips #22 I recommended that disks with fractured files should not be copied with a quick copier. Larry informed that his trademarked QUICK-COPYer is the only program of its kind on the market that does un-fracture files. He sent along a copy to prove it, and it does just that. A very useful feature!

Now, here is the new, and final, version of the Tigercub Menu Loader.

```
100 !by A. Kludge/M. Gordon/
T. Boisseau/J. Peterson/etc.
Version #5, 9/85
110 CALL PEEK(8198,A):: IF A
<>1170 THEN CALL INIT
120 OPTION BASE 1 :: DIM P$
(127),V(127,3):: CALL LOAD(-
31806,16):: ON ERROR 130 ::
GOTO 160
130 DISPLAY AT(12,9)ERASE AL
L:"I/O ERROR" :: RUN 100
140 @,@@,A,A$,B,C,D$,FLAG,I,
J,K,KD,KK,M,M$,N$,NN,P,P$,PG
$(),PP,PP$,Q$,S,ST,T$(),TT,V
T,V(,),W$,X,X$,Y,K2,S2
150 CALL LINK :: CALL PEEK :
: CALL KEY :: CALL SCREEN ::
 CALL COLOR :: CALL CLEAR ::
 CALL VCHAR :: CALL SOUND ::
 !@P-
160 CALL CLEAR :: CALL LOAD(
8196,63,248):: CALL LOAD(163
76,67,85,82,83,79,82,48,8)
170 CALL LOAD(12288,129,195,
126,165,129,153,102,60)
180 CALL LOAD(12296,2,0,3,24
0,2,1,48,0,2,2,0,0,4,32,32,3
6,4,91):: CALL LINK("CURSOR"
)
190 CALL CLEAR :: FOR S=1 TO
 14 :: CALL COLOR(S,7,16)::
NEXT S :: CALL COLOR(0,2,16)
200 T$(1)="d/f" :: T$(2)="d/
v" :: T$(3)="i/f" :: T$(4)="
i/v" :: T$(5)="pro" :: ON WA
RNING NEXT
210 IMAGE ###
220 IMAGE ### Quit
230 IMAGE ### Delete
240 IMAGE ### Print
250 IMAGE ### Rescan
260 CALL SCREEN(5):: CALL VC
HAR(1,31,1,96):: DISPLAY AT(
1,4):"TIGERCUB MENU LOADER"
270 ! IF YOU HAVE MORE THAN
ONE DISK DRIVE, DELETE THE !
 IN LINE 280 AND THE FIRST S
TATEMENT IN 210
280 ! DISPLAY AT(12,6):"DISK
? (1-3):" :: ACCEPT AT(12,19
)SIZE(-1)VALIDATE("123"):D$
:: D$="DSK"&D$&"."
290 D$="DSK1." :: OPEN #1:D$
,INPUT ,RELATIVE,INTERNAL ::
 INPUT #1:N$,A,J,K :: DISPLA
Y AT(1,2)SIZE(27):SEG$(D$,1,
4)&" - Diskname= "&N$;
300 DISPLAY AT(2,2):"Availab
le=";K;"Used=";J-K;" Prog Fi
lename Size  Type":"---- ---
------- ---- -------" :: I,V
T=0 :: TT=J-K
310 FOR X=1 TO 127 :: IF X/2
0<>INT(X/20)THEN 340
320 DISPLAY AT(24,1):"Choice
? Enter for more 0" :: ACCEP
T AT(24,24)VALIDATE(DIGIT)SI
ZE(-3):K :: IF K=0 THEN 330
:: IF K>0 AND K<NN+1 THEN 60
0 ELSE 320
330 X=1
340 I=I+1 :: IF I>127 THEN K
=X :: GOTO 510
350 INPUT #1:P$,A,J,B :: NN=
NN+1
360 IF LEN(P$)=0 THEN 430
370 DISPLAY AT(X+4,1):USING
210:NN :: DISPLAY AT(X+4,5):
P$ :: P$(NN)=P$ :: DISPLAY
AT(X+4,16):USING 210:J :: DI
SPLAY AT(X+4,20):T$(ABS(A))
380 V(NN,1)=A :: V(NN,2)=ABS
(B):: V(NN,3)=J
390 X$="  "&STR$(B):: DISPLA
Y AT(X+4,24):SEG$(X$,LEN(X$)
-2,3):: VT=VT+J
400 IF A>0 THEN 410 :: DISPL
AY AT(X+4,20):"Y"
410 CALL KEY(0,KK,ST):: IF S
T=0 THEN 420 :: FLAG=1 :: GO
TO 430
420 NEXT X
430 DISPLAY AT(X+4,1):USING
220:NN :: DISPLAY AT(X+5,1):
USING 230:NN+1
440 IF VT=TT OR FLAG=1 THEN
460 :: DISPLAY AT(2,25)SIZE(
4):VT
450 FOR @=1 TO 10 :: DISPLAY
 AT(2,25)SIZE(1):CHR$(30)::
DISPLAY AT(2,25)SIZE(1):" "
:: CALL SOUND(-99,110,0,-4,0
):: NEXT @
460 IF FLAG=1 THEN 470 :: DI
SPLAY AT(X+4,13):USING 240:N
N+2 :: DISPLAY AT(X+5,13):US
```

```
ING 250:NN+3
470 DISPLAY AT(X+6,1):"    C
hoice?" :: ACCEPT AT(X+6,16)
SIZE(-3)VALIDATE(DIGIT):K
480 IF FLAG=1 THEN 500
490 IF K=NN+2 THEN 840 ELSE
IF K=NN+3 THEN CLOSE #1 :: N
N=0 :: GOTO 190
500 IF K<>NN AND K<>NN+1 THE
N 590
510 IF K=NN THEN CALL CLEAR
:: CLOSE #1 :: END
520 DISPLAY AT(X+5,12)SIZE(1
2):" #?" :: ACCEPT AT(X+5,15
)SIZE(2)VALIDATE(DIGIT):KD :
: IF KD<1 OR KD>NN THEN 520
530 IF V(KD,1)>0 THEN 550
540 FOR J=1 TO 10 :: DISPLAY
 AT(11,1):" ":" PROTECTED -
 CANNOT DELETE":" " :: DISPL
AY AT(12,1):" " :: NEXT J ::
 GOTO 570
550 DISPLAY AT(X+6,1)SIZE(27
)BEEP:" Verify - Delete ";PG
$(KD);"?" :: DISPLAY AT(X+6,
28)SIZE(1):"Y" :: ACCEPT AT(
X+6,28)SIZE(-1)VALIDATE("YN"
):Q$ :: IF Q$<>"Y" THEN 570
560 DELETE D$&PG$(KD)
570 CLOSE #1
580 CALL VCHAR(1,3,32,672)::
 NN=0 :: X=0 :: FLAG=0 :: GO
TO 260
590 IF K<1 OR K>127 OR LEN(P
G$(K))=0 THEN 430
600 IF ABS(V(K,1))=5 OR ABS(
V(K,1))=4 AND V(K,2)=254 THE
N 640
610 DISPLAY AT(12,1)ERASE AL
L:"Print to ? S": :"(P)printe
r?":"(S)creen?" :: ACCEPT AT
(12,12)SIZE(-1)VALIDATE("PS"
):Q$ :: IF Q$="S" THEN PP=0
:: GOTO 630
620 DISPLAY AT(12,1)ERASE AL
L:"PRINTER? PIO" :: ACCEPT A
T(12,10)SIZE(-18):P$ :: OPEN
 #3:P$ :: PP=3
630 CALL CLEAR :: CALL SCREE
N(16):: ON ABS(V(K,1))GOTO 6
80,690,750,760
640 CLOSE #1 :: IF SEG$(PG$(
K),LEN(PG$(K)),1)="=" THEN D
ISPLAY AT(12,1)ERASE ALL:"RE
TURN TO BASIC AND LOAD BY":"
TYPING OLD ";D$&PG$(K):: STO
P
650 CALL PEEK(-31952,A,B)::
CALL PEEK(A*256+B-65534,A,B)
:: C=A*256+B-65534 :: A$=D$&
PG$(K):: CALL LOAD(C,LEN(A$)
)
660 FOR I=1 TO LEN(A$):: CAL
L LOAD(C+I,ASC(SEG$(A$,I,1))
):: NEXT I :: CALL LOAD(C+I,
0)
670 CALL VCHAR(1,3,32,672)::
 CALL SCREEN(8):: FOR S=0 TO
 14 :: CALL COLOR(S,2,1):: N
EXT S :: DISPLAY AT(12,2):"L
OADING ";A$ :: GOTO 900
680 OPEN #2:D$&PG$(K),INPUT
,FIXED :: GOTO 700
690 OPEN #2:D$&PG$(K),INPUT
700 LINPUT #2:M$ :: PRINT #P
P:M$ :: IF EOF(2)THEN 730
710 CALL KEY(0,K,S):: IF S=0
 THEN 700
720 CALL KEY(0,K2,S2):: IF S
2<1 THEN 720 ELSE 700
730 CLOSE #1 :: CLOSE #2 ::
PRINT "   >>>press any key<<
<" :: IF Q$="P" THEN CLOSE #
3
740 CALL KEY(0,K,ST):: IF ST
<1 THEN 740 ELSE 580
750 OPEN #2:D$&PG$(K),INPUT
,INTERNAL,FIXED :: J=0 :: GO
TO 770
760 OPEN #2:D$&PG$(K),INPUT
,INTERNAL :: J=0
770 IF EOF(2)=1 THEN 730 ::
J=J+1 :: INPUT #2:M$ :: IF L
EN(M$)=8 THEN 790
780 PRINT #PP:M$ :: GOTO 820
790 FOR Y=1 TO 8 :: @@=ASC(S
EG$(M$,Y,1)):: IF @@<32 OR @
@>127 THEN 810
800 NEXT Y :: GOTO 780
810 RESTORE #2 :: FOR X=1 TO
 J-1 :: INPUT #2:M$ :: NEXT
X :: INPUT #2:M :: PRINT #PP
:M
820 CALL KEY(0,K,S):: IF S=0
 THEN 770
830 CALL KEY(0,K2,S2):: IF S
2<1 THEN 830 ELSE 770
840 DISPLAY AT(24,1):"PRINTE
R NAME? PIO" :: ACCEPT AT(24
,15)SIZE(-14):PP$ :: OPEN #2
:PP$ :: PRINT #2:SEG$(D$,1,4
)&" - Diskname= "&N$
850 PRINT #2:RPT$("=",28):"A
vailable=";358-VT;"Used=";VT
:RPT$("-",28)
860 PRINT #2:"FILENAME  SIZE
   TYPE":RPT$("_",28)
870 FOR P=1 TO NN-1 :: PRINT
 #2:PG$(P);TAB(15);V(P,3);TA
B(20);T$(ABS(V(P,1)));TAB(25
);V(P,2):: NEXT P :: CLOSE #
2
880 DISPLAY AT(12,3)ERASE AL
L:"(P) to print again":" (R
) to rescan":" (Q) to quit"
890 ACCEPT AT(15,4)VALIDATE(
"PQR")SIZE(-1)BEEP:Q$ :: IF
Q$="P" THEN 840 :: CLOSE #1
:: NN=0 :: IF Q$="R" THEN 19
0 ELSE END
900 RUN "DSKX.1234567890"
```

This version turns off the Quit key, restarts itself rather than crashing on an I/O error, and has pre-scan for faster start-up. It displays disk name, sectors available and sectors presumably used - it also totals up actual sectors used and sounds a warning if any sectors are not accounted for.

It lists up to 127 programs and files by number, filename, number of sectors, program or file type, file record length, and write-protection. It will stop for menu selection on any keypress or at the end of each screen, continuing on Enter. It will load and run any program that can run from Extended Basic, displaying its filename while loading. If the filename ends in an asterisk, it will warn you to return to Basic. It will delete any unprotected program or file, after first requiring verification by filename, or will inform you if the file is protected. It will read any readable file, including internal numeric, and list it to screen or printer. It will dump a catalog of the disk to your printer, and it will offer the option of quitting or rescanning the disk or another disk. And it's free, I don't even want a freeware donation - but I would appreciate if you would take a look at my catalog and see if, somewhere among those 140 programs, there might be something you would be willing to pay $3 for? The Menu Loader is included as a bonus on every disk I sell!

```
100 CALL CLEAR :: RANDOMIZE
:: DISPLAY AT(3,4):"TIGERCUB
 MATH PUZZLE"
110 DISPLAY AT(6,1):"Insert
+, -, * (multiply) OR / (div
ide) between the    digits
to equal the total": :"Type
Q to give up"
120 DISPLAY AT(12,1):"Level
1 or 2?" :: ACCEPT AT(12,15)
VALIDATE("12"):L$
130 T,X=INT(9*RND+1):: M$=ST
R$(X):: Z$=M$&" "
140 FOR J=1 TO 4 :: Y(J)=INT
(9*RND+1):: Z=INT(4*RND+1)::
 ON Z GOSUB 240,250,260,270
:: Z$=Z$&STR$(Y(J))&" " :: N
EXT J
150 IF L$="1" AND T<>INT(T)T
HEN 130 :: Z$=Z$&"="&STR$(T)
160 DISPLAY AT(12,1):Z$ :: D
ISPLAY AT(18,1):" " :: DISPL
AY AT(20,1):" " :: DISPLAY A
T(22,1):" "
170 P=2 :: FOR J=1 TO 4 :: A
CCEPT AT(12,P)VALIDATE("Q+-*
/")SIZE(1):S$
180 IF S$="Q" THEN 200 ELSE
IF S$="+" THEN X=X+Y(J)ELSE
IF S$="-" THEN X=X-Y(J)ELSE
IF S$="*" THEN X=X*Y(J)ELSE
X=X/Y(J)
190 P=P+2 :: NEXT J :: IF X=
T THEN 230 :: DISPLAY AT(18,
1):"WRONG!"
200 DISPLAY AT(20,1):"ANSWER
 IS ";M$
210 DISPLAY AT(22,1):"PRESS
ANY KEY"
220 CALL KEY(0,K,ST):: IF ST
<1 THEN 220 :: GOTO 130
230 DISPLAY AT(18,1):"RIGHT!
" :: GOTO 210
240 M$=M$&"+"&STR$(Y(J)):: T
=T+Y(J):: RETURN
250 M$=M$&"-"&STR$(Y(J)):: T
=T-Y(J):: RETURN
260 M$=M$&"*"&STR$(Y(J)):: T
=T*Y(J):: RETURN
270 M$=M$&"/"&STR$(Y(J)):: T
=T/Y(J):: RETURN
```

Enjoy!

Jim Peterson

PRESIDENT'S MESSAGE STEVE CITRON

It has been a very rewarding and difficult month for me. There is so much that is new and exciting in the T.I. world that I find it virtually impossible to keep up with all the changes that are occurring. I will report on a number of items that I find to be of most interest. The new Myarc computer is a reality, however many attending the Chicago TI-Fest were disappointed because they believed that they would see it there. The New Computer will be going int production soon and will be released in New Jersey in March at the First Annual TI-MYARC REGIONAL COMPUTER FEST.

There is much work to be done and few short months to do it in. If you are interested in helping please contact me as soon as possible. Further Details at the next meeting.

At the officers meeting in November, it was unanimously decided that the motion to increase the Board of Directors to 5 be approved and that it be brought to the membership at the December meeting. If Ratified by our membership there will be two vaccant positions on the board to be filled at that time.

Also approved by the board was a motion that NEW JUG officially sponsor The TI-MYARC FEST, there was one opposing vote on that motion.

On another note PSE&G has requested that we submit a profile of our membership so that they will have a better idea as to the nature of our group. Our Annual questionairre is being re-written with that in mind, and will be issued in next month's newsletter, and at the January Meeting. We have borrowed an Auto answer modem and should have a Bulletin Board up for testing purposes within a week. We have received a number of BBS programs which we will be evaluating.

I wish to remind the membership that January is dues renewal month. Please make every effort to pay as early as possible. Dues are $15 individual member and $20 family membership.

A year ago when I was elected President I began to institute a number of changes in the functioning of NEW JUG. A year later I feel there is still room for improvement. We need to hear suggestions, reccomendations, etc. . We also need help. Please consider joining tl Board , a committee, or a SIG. And please complete the annual questionairre, it is the best way we have of judging current interests. In our reviewing our 1981 questionairre only 1 member had a disk drive, and we were the only User Group in New Jersey.

An interesting Agenda has been planned for the December meeting.

* We will begin at 7:00 with 2 classes our Extended Basic group will meet in one room while our new combined Advanced Programming Languages group will meet in the Other. This new group will study Assembler, Forth and 'C'.
* I have appointed Rich Alfonzo to conduct the annual elections, as a past officer he is not currently running for any post.
* One of our members,Bill Reese has written a most intriguing program that can create or modifiy the screens of TI-Runner. He will demonstrate his program.
* A discussion entiteled "Where are We, and where would you like to go from here ?" will be led by Harry Potter
* Reviews and demonstrations of New Software and Hardware will be led by Bob Costello and Marv Shuldman.
* Following the meeting will be a workshop on the use of TI Writer.
* Plus a few surprises that I cannot divulge at this time.

Hope to see you all Monday December 9.

One, almost final note - Ken Gordon will be having his 15th Computer Show at the Aspen Manor in Parsippany RTE 46 on Sunday Dec. 8.1985. This is an all indoor show, Admission is $7.00. I have a number of discount tickets, which I received to late for distribution.

And finally, please try to avoid calling me the day of our monthly meeting. I average about 15 calls per meeting. It delays me considerably. If there is a reminder, how about calling ov the weekend. Thanks.

Steve Citron
201-686-5619 after 5
981 Townley Avenue
Union, NJ 07083

NEW BASIC/COMPUTER FOSTERS IMPATIENCE

There has been considerable confusion regarding the new Myarc computer. We have been hearing for the past two years about a "PHEONIX" computer with advanced capabilities. This computer was to have been designed and released by Corcomp, Inc.. Whatever happened to the 'PHEONIX' is unknown.

In October MYARC, INC. announced their development of a new T.I. compatible computer.. A number of articles appearing throughout the country seem to confuse this new computer with that of the long lost 'Pheonix'(Perhaps it burned in its own ashes). This computer is, or actually will be a reality in the VERY NEAR FUTURE.

MYARC is probably the most professional corporation to be involved with products for the TI 99/4A. Lou Phillips, president of Myarc, is one of the most knowledgeable and professional people I have yet to meet. Myarc's products are well designed, tested and reliable. Unlike other companies, MYARC continually updates their products and makes the updates available to consumers at extremely nominal costs. There has been much confusion concerning this because after products are updated a new set of instructions are released. Most of the questions I receive concerning Myarc products stem from the fact that many Users reading the original manual cannot understand why commands found in the original manuals no longer function in their updates. Lou assures me that all manuals are undergoing revision and final Updates will be released very shortly.

All electronic equipment is subject to failure, Myarc will accept repair and return defective equipment usually in the same week.

There have been many questions concerning the release date of Extended Basic Level IV - We will attempt to answer them ask best as we possibly can.

Questions about EXTENDED BASIC LEVEL IV

1...Why did it take so long to release?

THE REASON ; Myarc determined that it would be to everyones advantage to have the new Extended Basic emulate as close as possible the advanced basic language to be found in their new computer. This will allow program development now, so that software will be available for the new computer once it is release in Mid-March

Extended Basic Level IV was originally announced as an intermediate product to be followed by a series of updates as they became available, now however, it has been released as a fully operational, advanced programming language.

2...Can I use the new extended basic in my p-box with my TI or Myarc 32K card?

No! the new basic requires a minimum configuration containing the Myarc 128K card. (or the Myarc 512K card)

3...What about my Corcomp 128 or Foundation 128 K cards?

The Extended Basic Level IV has been specifically designed to operate in conjunction with the built in facilities of the Myarc 128K mem. card. (or 512 card)

Any other questions concerning the new computer or any Myarc product will be happily received. We will do our best to answer all questions as best we can. We are planning a current review sheet outline modifications that are now available and command changes in future issues.

Please address all inquires to me at the following address:

Steve Citron
981 Townley Avenue
Union, N.J. 07083
NOTE New phone number
201-686-5619

NEW JUG NEWS

NEW JERSEY USERS GROUP

DIRECTIONS

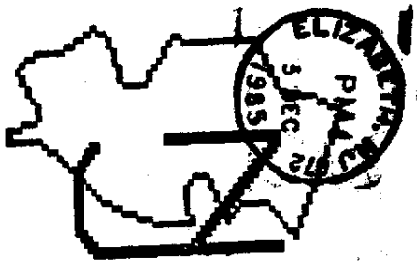
Take Garden State Parkway to exit 131, bear right toward Metuchen on Route 27 (Middlesex Ave.) until you reach the fourth traffic light (Main St.). You will have passed the Library. Turn right at light on to Main St., go one block and turn right onto Library Place, proceed half way up block; Library is on right. Park only on left side of street or on cross street (Linden Ave.). Don't use employees' parking lot.

Or from Route 287:

1. Take 287 south to Metuchen Exit;turn left off exit;
2. Bear right at fork;
3. Road will eventually bear to left;
4. At third light, turn left onto Main St.;
5. Go one block and turn right onto Library Place;
6. Library is 3/4 block down; use rear entrance.

Mel Gary
49 Pine Grove Ave.
Somerset, NJ 08873

DALLAS TI H. C. GROUP
1221 MOSSWOOD PL.
IRVING, TX 75061



T.I.-MYARC

1st ANNUAL REGIONAL COMPUTER FEST

FEATURING:

EXCLUSIVE: RELEASE OF NEW COMPUTER UNITS WILL BE ON SALE

HANDS ON OPERATION

SATURDAY MARCH 15, 1986

FOR FURTHER INFORMATION

CHAIRMAN - STEVE CITRON 201-686-5619
981 TOWNLEY AVE. UNION NJ 07083

CO-CHAIRMAN JEANETTE SHADER 201-929-0538 VOICE
201-929-8154 BBS