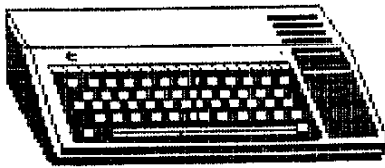


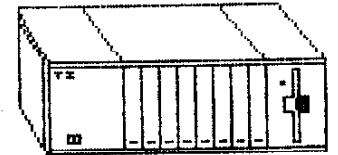
NEW

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

JANUARY 1986

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

President:
Steve Citron
981 Townley Ave.
Union, NJ 07083

Send Dues To:
Mary Shuldman
28 Tyndall Rd.
Kendall Pk., NJ 08824

Write For Application:
Carol Sudol
43 Salem Ct.
Metuchen, NJ 08840

T. I. C. O. F. F. '86

- MARCH 15, 1986 -

ROSELLE PARK HIGH SCHOOL

BE AWARE THE IDES OF MARCH

Plans for T.I.C. O.F.F.'86 are progressing at a rate that I would not have even considered. The Idea of a T.I. Festival for the East Coast is picking up momentum. I have been receiving calls from all over the country. I am going to use my column this month to give a partial idea of what has been happening.

First, the NEW MYARC EXTENDED BASIC LEVEL IV is about to be shipped. I have a copy which I have been fooling around with and will demonstrate at our next meeting. Saying that it is fantastic would be downrating it. I find it difficult to believe what our computer can really do with a highly advanced basic. Enough said, now on to the fair...

PLACE : Roselle Park High School. This school was chosen for a number of reasons. It is in a quiet suburban setting, it is convenient to mass transportation and the major auto routes in New Jersey. It is close to the Garden State Parkway, The New Jersey Turnpike, Routes 28,22,24/78 and Routes 1 and 9. It is only a few minutes from Staten Island and The Newark Airport. The facilities at the school are fantastic. The Auditorium has comfortable seating and a fine acoustic sound system.

DATE: March 15,1986

A number of User Groups from all over the country will be sponsoring What looks like might be the largest single event of its kind in history. The T.I.C. O.F.F. 1986, a day filled with activities relating to the "ORPHANED" computer the TI 99/4a.

A series of lectures will cover every aspect of computing in general and TI in particular, leaders in research, various college professors will participate in the 'Educational' events. Among the speaker scheduled to date are: Lou Phillips of Myarc, John Brown, Paul Charlton, Dr. David Jagerman (Stevens Institute), and Jay Halovacs. Unconfirmed at this time are talks by Mac McCormick and Chris Flaherty (TI-Artist)

A giant Commercial show with 20,000 square feet of exhibition area is planned as well as a Flea Market.

The New Myarc Computer will be Introduced for hands on testing and an a superb demonstration of the New Myarc Extended basic Level III.

User groups from around the world will be participating. Many develop ers who have produced products for the TI but were never released because of what was believed to be no market potential will finally be released.

We are hoping to publish a journal, which will include articles of General Nature as well as In depth Articles for both Hobbieists, Users, Teachers, programmers, developers, etc. We are giving one full page of advertising for each 3 page article submitted. This will give smaller companies, which could otherwise not afford advertising, the same advantages as larger companies.

A massive advertising campaign has been planned, to start approx. 4 weeks before the show. We will need the help of as many people as possible. Press releases are being prepared for individuals wishing to submit them to their local community newspapers etc. We have been requested to submit 5 Press passes to the New York Times. We anticipate a lot of Press coverage.

COMMITTEES

TRAVEL COORDINATOR	Monarc Travel attn: Mrs. Tillie Blahsky 1429 US Route 22 Mountainside, NJ 07092	
		1201-654-6211
PROGRAM COMMITTEE-Speakers etc.	Jay Halovacs	1201-356-3150
PUBLICITY-	Bill Dubrow	1201-463-9415
	Henry Hein	1201-385-9057
FREWARE/FAIRWARE/PUBLIC DOMAIN SOFTWARE-	Art Byers	1914-528-5402
COMMERCIAL EXHIBITS AND FLEA MARKET	Bob Costello	1201-663-4512
	Randy Evans	1201-549-5926
ADVANCED SALES(GROUP DISCOUNTS)	Jeanette Shader	1201-929-0532
LOCATION LIASON	Bob Guellnitz	1201-382-5863
AREA COORDINATORS		
NEW YORK	Art BYERS	1914-528-5402
	Carney Mimms	1914-961-5993
NEW JERSEY	Jeannette Shader	1201-929-0532
	John Simtkins	1609-939-6028
DELEWARE	Jack Shattuck	1302-764-8619
PENNSYLVANIA	Tom Burke	1215-927-4495
	BBS	1215-927-6432
MAINE	Mark Ridout	1207-797-2104
COORDINATOR	Steve Citron	1201-686-5616
	981 Townley Avenue	
	Union, N.J. 07083	

**BE AWARE THE IDES OF
MARCH**

NOTE WELL!

1) **T.I.C.O.F.F.** - anyone with a list of community (neighborhood) newspapers should contact Steve Citron (201-868-5619) or Jeannette Shader (201-929-0532).

2) **SOFTWARE LIBRARY** - would the person that has the disks from our software library please return them to one of the club officers. Without them we have no club software library!

WHAT'S ON FIRST

By Dan Fersht

The world seems to have gone data base crazy. It appears that unless you own at least one or perhaps all of the existing data base programs then you are not a true "computerist", if such a word exists.

Most of us own a program that contains a real, honest to goodness data base. It is not cumbersome and it is not difficult to set up.

I'm talking about our old friend TI Writer, which is perhaps the most efficient and useful amongst all of the software that TI produced. TI Writer while being an excellent word processing package, (particularly when you consider the display limitations of the TI99/4a), can double as a powerful, easy to use data base for keeping track of information.

The key to TI Writers value as a data base manager lies in its Find String feature. By using it, you can move to any specific keyword hidden in the text. You just have to enter FS at command level and respond with the desired string at the prompt (i.e., /string/). You'll be surprised at how quickly the cursor seems to appear on the first occurrence of that string, regardless of its placement in the file. It's so fast that it almost appears to be magical in its response. If the string is not the one you are specifically searching for, you merely have to return to the command level and reissue the search command. As you know, it is not necessary to reenter the string. The cursor will move to the next location where that string is found. The search begins wherever the cursor is located at the time the command level is entered.

The following is a sample application which keeps track of magazine articles of interest using TI Writer's Tab Function to set up a format similar to the one that follows:

MAGAZINE

ISSUE

ARTICLE TITLE

SUBJECT

Under subject, enter appropriate key words such as GRAPHICS, EXTENDED BASIC, GAMES, DISK DRIVES, etc. The others are self-explanatory. When you want to locate a particular subject or article, (by title), search the file using titles or key words.

The file capacity is not large, (about 27,000 to 30,000 characters). However, the ease with which files can be created and accessed make this approach attractive, and more important, convenient for tracking information. If you already own TI Writer, you don't need to purchase anything else to begin using this data base. This method is a lot faster than writing your own data base manager. It's also a lot cheaper than going out and buying one that may be less sophisticated and more difficult to use than our old friend TI Writer. Before logging off for this first month of 1986, I would like to thank Glenn Davis for his valuable contributions to this, as well as other articles I have submitted for your approval.

TI-FORMATTER

By Tom Kennedy

(d/l from CompuServe by ael gary)

The Text Formatter is a program which prints out the document. Most importantly, the special symbols, called Format Commands, that the formatter uses to alter the print-out of the document, which are installed in the Text Editor. In other words, you put these commands into the text when you write it and as the formatter comes across them it changes the text accordingly but doesn't actually print the symbols. There are six groups of formatter commands that are all applied in a similar manner. All commands must be in caps and must be on a line that starts with a period. The use of these commands in your text is what separates the word processor from a typewriter. They allow you to get the most out of your printer.

So, now you've written your document, and inserted all the format commands, now how do you print it out? First, save the document and exit the Text Editor. At the title menu, select Text formatter, (make sure the

program disk is in the drive) and the screen will blank with the prompt "ENTER INPUT FILENAME". Enter the name of the file you just saved, (ex. DSK1.MYFILE) and hit enter.

Next, the prompt "ENTER PRINT DEVICENAME" appears after the file is loaded. If you use a serial printer, the device name would be RS232.BA=xxx with xxx being the baud rate. If you're using a parallel printer, the device name is PIO. Also, you must add either .CR or .LF to the end of the device name. This tells TI-Writer whether your printer will handle the carriage return or the line feed. Check your printer manual and the TI-Writer manual in detail to find out which you use.

The next prompt is "USE MAILING LIST". If you aren't printing "form letters" just hit enter to accept the default of N (NO).

Next is "WHAT PAGE(S)? <ALL>". If you want to print the whole document, accept the default for all pages. Otherwise, you can print any of the pages or groups of pages.

The prompt "NUMBER OF COPIES: 1" tells how many copies of each page are to be printed.

The last prompt is "PAUSE AT END OF PAGE? N". The main purpose of this function is if you are using separate sheets of paper it will stop and wait for you to align the next sheet. Another use is to save a little paper. TI-Writer has an annoying habit of scrolling one whole blank page up before starting to print, which is not that big of a deal since what's one piece of paper worth considering how much you go through normally. But if you're just running test samples of type styles, or the like, you end up with a lot of white paper at your feet. To prevent this, type "Y" and turn off your printer. Now hit enter and turn the printer on, you should see "PRESS ENTER TO CONTINUE" (the software thinks one page has been printed). If not, turn the printer on and off again. Now you align the paper to the top of the page and hit enter and the printing begins. But if it's a long letter, you'll have to sit there and hit enter after each page so usually it's better to select the default when using continuous feed paper.

Now, about the Mailing List Option. Let's say you've written a form letter to send out to various individuals, maybe a resume. You write the letter like normal, but when you come to a name or address or something that will change with each letter, you put in its place a variable in the form of #n#, where n is a number to identify the order. So instead of starting off with: "Dear Mr. Smith" you would have "Dear Mr. 1" and so on. when you're all through with your letter, save it and purge the memory. Now you must create what is called a Value File, which is your mailing list where TI-Writer will draw the variables from. A value file consists of a list values to be inserted into the letter, listed one to

a line, preceded by the number of the variable and ending with a carriage return symbol. Groups of values must be separated by a line with just an asterisk and a carriage return. For example:

```
1 John Smith
2 123 STREET
3 Seattle, WA
*
1 Jane Doe
2 456 STREET
3 Seattle, WA
```

At the top of your letter you insert the .ML f command where f equals the filename of your value file. After selecting the mailing list option the computer will use this command to fill in the variables. If there is no .ML command in the letter then when you are prompted for "MAILING LIST NAME:" you supply the filename. This allows you to call on a number of files for different groups.

Another way to insert values is to use the Define Prompt command. With this command you do not insert a .ML command calling a value file and instead you insert lines containing the format: .DP n:t - where n is the number of the variable and t is the prompt text. Now, when you come to the prompt "USE MAILING LIST?" you select "N" for NO and as the document is printed when a variable is encountered the printing stops and the text you chose appears on the screen asking you for the appropriate value. If you don't include a ".DP n:t" command in your text, the computer responds with "ENTER DATA FOR VARIABLE #n#" and it can get confusing trying to remember which item you're on. This method is handy for letters which you only want to print one copy at different times to different people.

ASSEMBLY PROGRAMMING

by Jay Holovacs

Sometime back (time flies) I promised a guided tour of actually using the Debugger to work on an actual assembly program; essentially to overcome the "how do I start?" feeling that most people face when beginning assembly language programming.

The first point (and this makes a good lesson) is that there are some errors in the Sillywriter listing in the Oct. NJUG news; so a corrected listing is included at the end of this article. It is almost a guaranteed occurrence that both syntax (typically typos) and logical errors will exist in any new Assembly program.

After completing the source listing in the editor, save it using option 3. I recommend a standard suffix for source, object and list files (I use /S, /O, /L etc.) which will keep them recognizable.

Load the assembler using option 2 of the main menu. When prompted for source file use the name you just saved. Also you must designate files for object code (actual executable code) and list file (essential for debugging). Though a printer can be specified for list, I recommend listing to disk. The process runs faster, and the listing can eventually be printed out from the editor when the entire program finally assembles without error messages (this typically can take several tries). 'Options' will be required next, specify RL, meaning that the assembler will recognize the R register abbreviation and that a listing file will be generated.

As the assembler executes, any errors it finds will be listed to the screen. Write down the line numbers for correction later (the numbers will also be in the list file if you forget). Don't take the error description wording too literally, for instance 'invalid register' could be really referring to a location where you forgot to place the 'R' before a label, fooling the assembler into looking for a register with that name. Any fixes must be done on the SOURCE file in the editor and the assembler then re-run. Once complete, load the listing back into the editor (don't worry about the control characters warning you get) and 'save' it to printer using option 3 (this is faster than the print option).

Now look at the listing. It contains 3 columns of hexadecimal numbers, plus your original source code. The first column references line numbers in the original code. The second lists memory addresses (relative to the beginning if you used relocatable code like this example; actual if you assembled with the absolute code option). The third column is the actual contents of these addresses.

Note that instructions (lines of source code) can occupy one, two or three hexadecimal words. Note also that STATUS and NEWWKY did not generate addresses, the EQUATE instruction merely substitutes the value of these wherever they were referenced. This is different from the DATA and BYTE directives, which place the value you specify in a memory location and place the ADDRESS of the memory location wherever they are referenced.

Go back to the E/A main menu and load your object code first; then DSK1.DEBUG. Once this is loaded, hit enter, and type in DEBUG as the program name. You will now be running DEBUG, and will execute your program from this environment. The first thing you will see is a '.' on the lower left of your screen. This is the DEBUG prompt. All commands are single letters, followed by appropriate arguments; backspace and delete don't work. (To abort a command that you have started to enter hit a . which is an invalid termination.)

Since the sample program is relocatable code, and since it was loaded on a 'clean slate' it should start at A000 (all addresses will be given in hex) plus the

address. The first executable address is at the LMPI instruction (A022 if typed in exactly as listed). To verify this, enter 'M' at the prompt then A022. The correct value, 02E0 should appear. Using the space bar to increment, '-' to decrement, move around the memory and compare the values to those on your listing. Note that the values marked with an apostrophe in the list are different because these are offsets from the beginning of the program. Since the assembler does not know in advance where a relocatable code program will be loaded, affected values are added and inserted by the loader while loading.

Now to test part of our program. Pick a breakpoint (not too far away). Since the program almost immediately jumps to the CLRSCR subroutine, if we are confident in the first few lines we can insert the breakpoint there. A good spot is CLR2 (breakpoints must be at the beginning of an instruction, but do not have to be at label locations). To set, type B A074 at the prompt (some consoles will give the 'breakpoint uses two words' message; this is no problem). The program stops just BEFORE executing the instruction indicated in the breakpoint.

There are certain rules of thumb for choosing breakpoint locations. These are ones that I use:

- 1) Never go more than a few instructions in untested code.
- 2) Look at the registers after each breakpoint and confirm that they contain what you expect. Also check any memory locations that were accessed in the stretch.
- 3) Always break at a conditional jump the first time to check the status register and see if the conditions are what you expect. Test each set of conditions upon reaching the jump.
- 4) Restart from a common point where possible rather than starting from where the breakpoint left off.
- 5) Place a breakpoint at the beginning of a called subroutine the first time to make sure that entry conditions are as expected.
- 6) Place a safety net breakpoint further down in code in case of an unanticipated jump, or other reason the breakpoint you are working with was not executed.

To start executing in the correct place, we need to set the program counter in the 9900, do this with the R command. The first value is the workspace pointer, since our first instruction will load this, it need not be loaded now. Hit space bar to advance to the program counter and enter A022. Hitting the space bar again will give us access to the status register, but that is a more advanced feature. Hit enter to set up these values. They can be reviewed at any time with the R command.

Hold your breath and enter the Q command. If all went well, a fraction of a second later 3 addresses will appear at the bottom of your screen; stopping address, workspace at time of stop, and status register at time of stop. Hit W [enter], and the contents of the current workspace will be displayed. With all working well, R0 will contain 2FF and R1 will have 2000. Now try going a little further. Place a breakpoint at A07C (the RTWP instruction) and reset the program counter to the beginning. This time the complete screen clearing should work, examination of the workspace will should show that R0 has decremented to zero.

(DEBUG has many other commands, some, like hexadecimal arithmetic, X, Y, and Z user preset values, data movement and value comparison are very useful. Others, astonishingly, like single step and grow address change are either not operational or useless on the 99/4A. Read and try the functions to get a feel for the usefulness of the debugger.)

This should provide a taste of the debugging procedure. Next time we will cover some more techniques and hints, but in the meantime, EXPERIMENT. The worst that can happen is a crash, an likely you will discover that working in assembler is not so difficult after all.

#SILLYWRITER V0.2 (INCOMPLETE)

REF VSBW,KSCAN (MORE REF'S WILL BE ADDED AS PROGRAM EXPANDS

*IT IS EASIER TO FOLLOW CODE IF WE USE NAMES FOR VALUES:

STATUS EQU >837C
NEWKY EQU >2000 TEST FOR CONDITION BIT

```

WKSP    BSS    32
        LNPI   WKSP
** MAIN LOOP ****
*ROUTINE TO READ THE KEYBOARD
        BLWP   @CLRSCR CLEAR THE SCREEN
        CLR    RO      THIS WILL ADDRESS BYTE TO
SCREEN MEMORY
        MOVB   RO,>8374 EQUV TO THE 0 IN CALL
KEY(O,K,S)
        KYCHK  BLWP   @KSCAN
        MOV    @STATUS,R2
        ANDI   R2,NEWKY      SEE E/A MANUAL P
250, 251
        JEQ    KYCHK      NO KEY PRESSED, TRY
AGAIN
        MOVB   @>8375,R1
        BLWP   @VSBW WRITE THE VALUE TO SCREEN
MEMORY
        INC    RO      ADVANCE TO NEXT POSITION
        JMP    KYCHK    READY ALL OVER AGAIN
*CALL THIS ROUTINE WITH A BLWP @CLRSCR
*CLEARS SCREEN, NO PARAMETERS PASSED TO OR FROM THIS
ROUTINE
        CLRSCR DATA CLRWKS,CLR1 POINTERS TO WORKSPACE
AND ENTRY
        CLRWKS BSS    32      WORKSPACE FOR CLEAR
SCREEN ROUTINE
        CLR1   LI     RO,>2FF  START AT END OF SCREEN
IMAGE TABLE
        LI     R1,>2000  PUT BLANK IN HYBYTE
        CLR2  BLWP   @VSBW  WRITE TO VSB
        DEC   RO
        JNE   CLR2
        RTWP                      RETURN TO CALLING
PROGRAM
while loading.

```

HA HA IBM
OLDENS IS OLD
FAREAST
all lower case
NEW FROM CS&D

NEW COMPUTER!!!!

As everyone is aware, Myarc is planning to introduce a new computer which is rumored to be based on the design of the ill-fated TI 99/8. In fact, Myarc even had a 99/8 to play with before it was cancelled in just two months before TI left the home computer market. The truth about the 99/8 was that it was largely incompatible with the 99/4A. Thus, when Myarc decided to design a new computer they had to make major changes to the design of the 99/8 and the result of this work is a computer originally named "Noah" (from the "arc" in Myarc . . .) and now in search of a number for a name.

It was widely expected that Myarc would show this computer at the TI Faire in Chicago on November 2. But no dice. They brought along an empty shell of what the machine would look like and a mother board that they claimed was the machine. You may well ask then, why didn't they show it in operation. The answer is simple, although Myarc wouldn't admit it straight out. They blew a chip on the board when they were working on it the day before the show and were unable to replace it in time. But Lou Phillips, president of Myarc still gave a very clear picture of what this new, unnamed machine is all about. First the basic information.

It is expected to be released in the first quarter of '86 and sell for \$499. The machine has an IBM key board complete with a slash key where the left shift key should be. There are 10 function keys but instead of being mounted on the left of the keyboard as on the IBM keyboard they are mounted across the top of the unit horizontally. There is also a numeric keypad like on the IBM, but instead of an oversized plus (+) key there is a large enter key to facilitate numeric entry. The cartridge port has been moved to the upper left hand part of the machine above the first few function keys.

It will come initially with 256K of CPU memory (expandable to a full 2 megs), 64K of VDP memory, 64K of ROM, a parallel output, and an RS232 I/O port, two internal expansion slots, and a port to hook up a mouse. The mouse Phillips mentioned was the MS (Microsoft) Mouse which brings up the issue of IBM compatibility (more later). The internal ROM includes 48K of library routines, 8K of GPL interpreter and 8K (seems like a lot to me) of mouse support. When the machine powers up 16K of RAM is used for various internal tasks so that you are left with about 240K of space for your programs. And remember that all routines, screen and graphics tables are kept in the 64K of VDP memory, so that you really have quite a lot of memory to work with. If you choose to expand the RAM of the system, it will have to be done externally using 3 off board RAM expansion banks. The current Myarc memory cards such as their 128 and 512K cards will work as memory expansion.

The machine is built around the TMS9995 microprocessor which is a more advanced version of the TMS9900 inside your TI-99/4A. The 9995 is 2.3 times faster and comparable in speed to a Motorola 68000 that drives Apple's Macintosh. According to Mack McCormick the 9995 can run as fast as 12 MHz but it looks like it will only be running at an incredible 10.7 MHz due to some technical considerations. The 9995 uses 16 bit parallel memory on the main board which allows it to go even faster than the 9900 which was a 16 bit processor doomed to forever run on an 8 bit bus thus working at only half speed (roughly...).

The machine will be able to run nearly all programs written for the 99/4A through a bit on the gate array which when set will make the machine look nearly identical to a 99/4A. Thus all your software is still good. Almost. Myarc says 99% compatibility. The exceptions they've found are programs that use non-standard methods to scan the keyboard. This is only two programs so far. No big deal. The reason for the problem is that the 99/4A has 48 keys and the new machine has 84 so that a different KSCAN routine obviously had to be used. The programs that don't work use their own KSCAN routine and thus will not work.

A few more comments on compatibility. There will probably not be immediate support for speech. The machine can support it but there will be no port for you to plug it into the side of the machine. Myarc is planning to develop something like the Triple Tech card from CorComp to allow you to put the speech synthesizer inside the PE Box. There is worse news though for those of you with a P-Code card. McCormick said that that card is a technical nightmare and that the increased development time and costs to allow it to work wouldn't be worth it. Besides, he added, P-Code is essentially dead as even its creator has abandoned it. Now here's the bad news for everyone. You can use your current PEB but you will have to buy a card from Myarc to be able to do it. The reason is that the flex cable and card that connect your console to your PEB doesn't have the intelligence or connectors to allow the new machine to access the expanded memory in the PEB on a 16 bit bus or using the new PAB format (more later). However having to buy this new card isn't all bad. It won't have as bulky a cable as the TI card so you can move the console around freely and it will have a time and date function built in so that you don't need a clock card. It is an added expense however.

The communications chip is the same 9901 that is used in the 99/4A running at the same speeds. The graphics chip inside the machine is perhaps the single most impressive component. Myarc is using the 9938, a chip TI developed and then abandoned (like all good things). It has 64 pins and is now being produced by the Japanese (who else?). It is fully compatible with 9918A inside the 99/4A but supports extra modes and features.

Where the 9918A has 8 control registers for graphics characteristics, the 9938 has 32 which allows for an incredible amount of flexibility and power. The 9938 has two text modes. The first is identical to the text mode of the 9918A except that you can choose the foreground and background colors from a set of 512 colors instead of 16. Text mode two is 80 by 24 or 80 by 26 (which allows for a status line at the bottom like on the IBM) with 6 x 8 characters and a choice of two colors from the same 512.

Multicolor mode is still there as well as graphics mode I. Graphics mode II allows definition of 768 different patterns and a choice of 16 colors from the 512. Graphics mode III is the same as mode 2 except that instead of only being able to have four sprites on a horizontal line at a time you can have up to ten on a horizontal row. Graphics mode IV is similar but has 256 x 212 resolution and graphics V can support up to 512 x 424 using interlacing but this mode can only be displayed on an RGB monitor. Graphics mode VI has 512 x 212 resolution and 16 colors. Each pixel can have its color individually defined. This mode requires the full 64K of VDP memory for storing the screen. Graphics mode VII has the same resolution but uses a full byte of memory to define the color for each pixel which means that each pixel can be one of 256 colors! This mode requires additional VDP memory to use and Myarc has made provisions for up to 196K of VDP RAM to be put in the console. One of the control bits on the 9938 allows for what Phillips calls "animation tricks." He says that it can do screen swapping which essentially provides for automatic animation controlled by the 9938.

The machine will support the old PAB (Peripheral Access Blocks) format iVDP memory so that, in theory, all peripherals manufactured to TI specifications will work. There is some question as to whether or not the CorComp disk controller will work but Myarc seemed to imply that it would. A new PAB format will also be supported. It will be identical to that developed for the 9978 and will reside in CPU memory for faster speed. It will also allow for logical record lengths of up to 4096 characters instead of the 256 on the 9974 and will have a full byte reserved for error codes which means there can be 256 error codes instead of 8 as in the old PAB format. Including support for both the new and old PAB formats is one of the major changes from TI's 9978. TI was planning to abandon the old PAB format which would have made your PEB 100% useless. Myarc has made provisions so that you don't have to buy a whole new system.

Phillips said that the first peripherals that would be released would be the new PEB interface (described above) and a new disk controller card that will fit in the internal expansion slot for people who don't have (and don't need to buy) the PEB. This disk controller will support quad density disks which means almost a full megabyte of storage on a single floppy.

Phillips said that they already have a version of this controller working and will probably release a version of it for the 9974A as well. After those two cards are complete Phillips says that the next thing he plans to work on is a card that will allow for IBM compatibility. He commented that the reason for choosing the keyboard that they are using was so that it could be made into a PC compatible computer easily. He also said that 3.5 inch drives were a definite possibility in the not too distant future.

The computer will come with Extended BASIC built in. But not TI Extended BASIC. Instead it will use an advanced version of Myarc's Extended BASIC II. Phillips said that XB II is very similar to GW Basic from Microsoft and is somewhere between 2 and 4 times faster than TI Extended BASIC. A complete description of XB II, which is now available for use on the 9974A when using Myarc's 128/512K memory expansion card, will be given elsewhere as it is too long to fit here. The additions to XB II that will be included in the new computer include full mouse support, advanced event driven control keys (which means that you can set your program to automatically branch to a certain line number when a certain key is pressed) and support for the new PAB format.

Phillips has promised to release a reference manual for the machine similar to the one released by IBM for the PC. In other words, the machine will have an open architecture and no hidden secrets like TI kept with GPL. This should help enormously in getting new software written and hardware built for the machine by third party companies which can fully utilize the incredible power of Myarc's new machine.

Phillips has promised to release the machine and claims that Myarc has sufficient capital to allow it to bring the computer to market. He did however admit that they are expecting a "hard, up-hill battle" for the first year. When asked about other languages, Phillips said that Pascal would probably not be next but that C would be. His reasoning is that C is what is really in vogue now and it would make new software development easier.

Listening to Phillips talk about this new machine made a few things very clear. First, that Myarc really has a machine nearly ready to release. Second, that the machine is state of the art and really something that could compete in the current market. Third, that Myarc is thinking long term and has big plans. Now whether or not a small engineering company from New Jersey working with a computer developed by TI that lost TI millions, can actually succeed is another question. I think that if anyone can, Myarc will. But there is no way to find out except to wait.

A few notes concerning this file: This file was written on November 4 1985 by J. Peter Hoddie, co-director of the Boston Computer Society TI 9974A user

group. It is based on several pages of notes I took at the TI Faire in Chicago on November 2 during a talk given by Lou Phillips of Myarc. This file is not complete in that I have several pages of notes I took at the TI Faire in Chicago on November 2 during a talk given by Lou Phillips of Myarc. This file is not complete in that I have lots more information on the product and many more editorial comments to make. . A complete article along with a full description of the faire, the products, people, and talks will be completed in time for the November 20 BCS meeting. It should be well over 10 pages in length. If you want a copy come to the meeting or send \$1 to the address below. This file is a rough draft. You may distribute it but please include the author's name as well as information as to where the final version can be obtained. Thanks. Boston Comp. Society TI 99/4A User Group One Center Plaza Boston, MA 02018 (617)-353-7369 (author's phone) (reconstructed and uploaded by Barry Boone (76354,16371))

FAIRWARE

(d/l from CompuServe by mel gary)

This is an incomplete list of the Fairware programs available for the TI99/4A Home Computer, compiled by Terrie MASTERS, President of LA Users Group and founder of User Group Network. As a Fairware author, you can be added to this listing by leaving a message to T. MASTERS (72157,1372) or calling (213)271-6930. UGN is dedicated to promoting FAIRWARE and communications between TI users/groups. Ask for details! TEXTFILE upload.

From User Group Network, the latest listing of Fairware. Compiled by Terrie Masters, President, LA 99 Users Group. SUPPORT FAIRWARE!

[1] **DISKMAN** Bruce Caron ... 25 Ottawa St. Arnprior, Ontario, Canada K7S 1W7 A marvelous disk-based Disk Manager which rivals CorComps manager.

[2] **DISCOPY** Steve Lawless 2514 Maple Avenue, Wilmington, Delaware 19808 EXCELLENT disk cloner! features ability to copy to 2 drives at once and uses the Foundation 128K card to copy a disk in ONE PASS!

[3] **DISASSEM** Fred Hawkins 1020 North 6th Street, Allentown, PA. 18102 An XB disassembler with many unique features and terrific documentation for those that PAY!

[4] **SUPER DISK DUPLICATOR** Tom Knight 7266 Bunion Drive, Jacksonville, FL. 32222 Allows inputting start and stop sector number for copying disks.

[5] **TK WRITER** Tom Knight (See Above) Loads TI WRITER from XB or E/A. No cartridge needed!

[6] **NEATLIST** Danny Michaels Route 9, Box 460 Florence, AL. 35630 XB utility to list multi-statement lines to printer or disk for easy reading and references program variables to line number used.

[7] **SCREENDUMP** Danny Michaels (See Above) Screen dump to EPSON compatible printer with double or single size and vertical or horizontal page printout.

[8] **The DIRECTOR** Ron Rutledge 1020 3rd Street Waukegan, IA 50363 XB program database that allows cataloging disk-based programs.

[9] **FAST TERM** Paul Charlton 1110 Pinehurst Court Charlottesville, VA 22901 Simply, THE BEST TERMINAL EMULATOR IN THE WORLD!

[10] **SPRITE BUILDER** John Taylor 2170 Estaline Drive Florence, AL. 35630 XB graphics generating program with assembly language routines for speed at crucial places. Includes a full disk of preformed graphics.

[11] **PILDT 99** Thomas Weithofer 1000 Harbury Drive Cincinnati, OH. 45220 An ENTIRE language for the TI that is the simplest programming language known to us (or anyone else!). Must send TWO PHYSICAL DISKS! and postage/mailer.

[12] **TE4TH** Ken Carruthers 3537 Faberge Way, Sacramento, Ca. 95826. A Terminal Emulator Program written in Forth Language. A must for persons interested in Forth Language. \$5.00 plus diskette and stamped mailer.

[13] **EASYSPRITE** Tom Freeman 515 Alca Real Dr., Pacific Palisades, CA 90272 An extremely fast XB program with assembly routines to create graphics sprites with easy cursor control saving for program insertion.

[14] **DISASSEMBLER** Marty Kroil 218 Kaplan Avenue Pittsburg, P. 15227 Super-fast disassembler, 100% assembly and full featured.

[15] **TECHIE BBS** Monty Schmidt 121 N. Blair, Madison, WI. 53703 Freeware BBS system for the 99/4A.

[16] **COMPACTOR** Monty Schmidt (see above) Assembly language program that takes an uncompressed D/F80 AL program and will compress to about 2/3 the disk space and yield faster load times.

[17] **UNCOMPACTOR** Monty Schmidt (see above) Opposite of above.

[18] **PRO 99er BBS** Mark Hoogendoorne 21 Long Street, Burlington, MA. 01803 TI BBS system with TRUE TE2 transfer capabilities.

[19] **DISK MANAGER** Todd Kaplan, 5802 N. Western Apt.

38, Chicago, IL. 60659 INCREDIBLE Disk Manager on disk; forget TI'S DM2.

[20] ASSAULT THE CITY, John Behnke 5755 W. Grace, Chicago, IL. 60634 An original Tunnels of Doom Game.

[21] HOMEPRINT, Bob Lawson 16223 Hill Point Dr., Houston, Tx. 77059. A utility to print Household Budget Management Module Files.

[22] DISK FILE CATALOGER, Jim Williams, 5217 122 PL. SE, Bellevue, Wa. 67115. A disk cataloger, additional information later.

[23] TRIVIA, Robert Wessler
*****WITHDRAWN BY
AUTHOR*****

[24] 2D GRAPHICS, Jean-Pierre Morin Ottawa U.G. 25 Arrnprior, Ontario, Canada K7S 1W7. An incredible graphics drawing program written in Forth, with an

outstanding demo, and documentation.

[25] CATLIB, Marty Kroll, 210 Kaplan Ave. Pittsburgh, Pa. 15227. A cataloging library program, capacity 123 disks, 900 files with many good features.

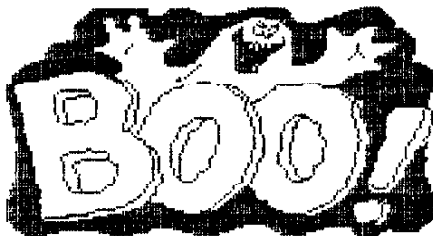
[26] MASS-TRANSFER, Stuart Olson, 25327 W. Wayside Place, Lake Villa, IL. 60046. Assembly language Terminal Emulator, menu driven, x-modem transfers, capable of multiple xmd transfers at once. \$10.00 plus disk and stamped mailer.

[27] CHECKBOOK and BUDGET MANAGER, John Taylor, 2170 Estaline Drive, Florence, Al. 35630. An extremely efficient program for both check book and budget maintenance. 10.00 plus disk and stamped mailer.

[28] SUPERBUG II, Edgar L. Dohmann, Route 5, Box 84, Alvin, Tx. 77511. Extensively redone version of the SUPERBUG program released to Public Domain by Texas Instruments.



GRAPHICS-USED AND TI-ARTIST



GRAPHICS-USED AND TI-ARTIST

NEW JERSEY USER'S GROUP - NEW JUG

MEMBERSHIP QUESTIONNAIRE

Please print all information neatly

PLEASE ALSO INDICATE
A) OCCUPATION ; B) EDUCATIONAL BACKGROUND

Last name _____ First name _____

Home address _____

Town _____

Zip code _____

State _____

Home phone _____

Bus. phone () - _____ ext _____

SEX AND AGES OF ALL THOSE USING
COMPUTER IN YOUR HOUSEHOLD.

(M/F) AGE COMMENTS

(M/F)	AGE	COMMENTS
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

EQUIPMENT OWNED:

HOW MANY

- _____ Console
- _____ Cassette cables
- _____ Cassette recorder
- _____ P-box
- _____ 32 K
- _____ Disk controller
- _____ Disk drive # _____ ds/dd ? _____
- _____ RS232
- _____ Modem
- _____ Printer
- _____ Dot/Daisy/TP _____
- _____ parallel/serial/both _____
- _____ Brand ? _____
- _____ P-code card
- _____ Others please specify ((128K 80col) _____
_____ ([cpm) etc. _____

MODULES OWNED:

- _____ EXTENDED BASIC
- _____ MINI-MEMORY
- _____ EDITOR ASSEMBLER
- _____ LOGO (I / II)
- _____ PERSONAL RECORD KEEPING
- _____ PERSONAL REPORT GENERATOR
- _____ STATISTICS
- _____ T.I. WRITER
- _____ MULTIPLAN
- _____ Others please specify: _____

WHICH OF THE FOLLOWING SPECIAL INTEREST
GROUPS OR WORKSHOPS WOULD YOU BE
INTERESTED IN ATTENDING ? please indicate
your proficiency (1-10) 10 the highest.
(y/n) (1- 10) comments

- | | | | |
|-------|-----------------------------------|-------|-------|
| _____ | CONSOLE BASIC | _____ | _____ |
| _____ | EXTENDED BASIC | _____ | _____ |
| _____ | ASSEMBLER | _____ | _____ |
| _____ | FORTH | _____ | _____ |
| _____ | PASCAL | _____ | _____ |
| _____ | LOGO | _____ | _____ |
| _____ | MULTIPLAN | _____ | _____ |
| _____ | GRAPHICS | _____ | _____ |
| _____ | MUSIC | _____ | _____ |
| _____ | TI WRITER | _____ | _____ |
| _____ | PERSONAL REC.KPB | _____ | _____ |
| _____ | PRINTER GRAPHICS | _____ | _____ |
| _____ | INTERFACING and
using PRINTERS | _____ | _____ |
| _____ | TELECOMMUNICATIONS | _____ | _____ |
| _____ | Compuserve/Source | _____ | _____ |
| _____ | bulletin boards | _____ | _____ |
| _____ | Computer speech | _____ | _____ |

COMMITTEES - Please check off those
committees which you are interested
in joining or leading. The club can
be no better than its membership.

- _____ software library
- _____ publicity/membership
- _____ telecommunications
- _____ program committee
- _____ board of directors
- _____ liason to other User's Groups
- _____ newsletter
- _____ budget committee
- _____ fund raising
- _____ yearly journal (magazine)
- _____ others we may have overlooked

Please use reverse side to indicate
what you would like to get out of
membership and what you can offer
the club.

CIRCLE DAYS CONVENIENT FOR MEETINGS
Su Mo Tu We Th Fr Sa times _____
COMPUSERVE No. _____
SOURCE No. _____

NEW JERSEY 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
47 Pine Grove Ave.
Somerset, NJ 08873



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

THE GREAT AMERICAN

T. I. C. O. F. F.

- 1 9 8 6 -

Texas Instrument Computer Owners Fan Festival
- 1986 -

March 19, 1986
Roselle Park High School
Roselle Park, New Jersey

FEATURING: New HP/PC Computer

Educational Program Hardware and Software Demos
Flea Market 20,000 square feet vendor area
Lectures, class and workshops Cafeteria with seating
User Group Meetings and Exchanges comfortable auditorium
S.I.G. Meetings - Special Interest Groups sound system
Fairware, Freeware, Public Domain Libraries
Simple parking, convenient to major highways
Roselle Park Railroad Station 1 block away

BE SURE THE DES OF MEET

For Further Information Contact:

Steve Citron Coordinator
981 Toomey Avenue Jeanette Shaden
Union, New Jersey 07083 1 201 929-0952
1 201 686-0626 1 201 929-8161 8169

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 disastrous volcano eruption that occurred near Bogota, 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 Columbian 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, Asseable, 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 now 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 necessary. Disk names can also be used as in ("DN1:,DN2:"), which destroys the contents of the disk named DN1: 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 either 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 duplicates. It is recommended 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 CTRL-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-D
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 extensibility, 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 ideas 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 directly 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-incremented (effectively removing this address from the top of the stack) and the contents of R8 are moved into the address 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 B$=SEG$(H$,1,10)
280 GOSUB 1860
290 PRINT "Need instructions?"
300 CALL SOUND(150,1400,0)
310 CALL KEY(0,K,S)
320 IF S=0 THEN 310
330 IF (K<>ASC("Y"))*(K<>ASC("y"))THEN 360
340 GOSUB 2390
350 GOTO 370
360 IF (K<>ASC("N"))*(K<>ASC("n"))THEN 300
370 CALL PEEK(28700,N1,N2,N3,N4)
380 FFAMM=N16+N2
390 LFAMM=N36+N4
400 GOTO 420
410 PRINT "Entry error!"
420 PRINT "Enter beginning of program:"
430 INPUT " ":S$
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$)<6 THEN 1160
1150 GOSUB 1210
1160 PRINT #1:" T15270 J. Burkett APR 85"
1170 CLOSE #1
1180 STOP
1190 REM
1200 REM Generate checksum char & output record
1210 ADDR=0
1220 FOR X=1 TO LEN(DATA$)
1230 ADDR=ADDR+ASC(LEFT$(DATA$,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<59 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,"00000038447C403C")
1920 CALL CHAR(102,"0018242070202020")
1930 CALL CHAR(103,"000004384438047C")
1940 CALL CHAR(104,"0040407B44444444")
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,"0000007B447B4040")
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,"0000004434545428")
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 PCMM & LFCMM, see MM manual pg
74).
2240 DATA ,2. The Beginning of the,Program (BOP) thru the
End

```

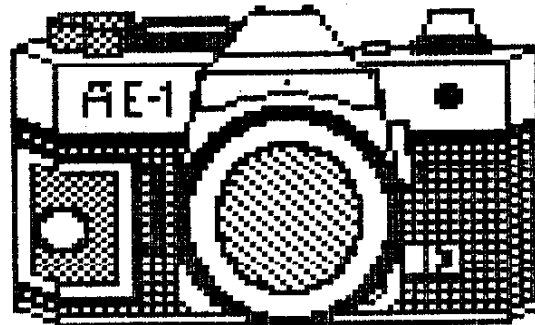
2250 DATA of the Program (EDP). BOP,must be entered. EOP may be
 2260 DATA entered or defaulted to,FFAMM-2.
 2270 DATA ,J. The REF/DEF Table (RDT).,The low end may be entered
 2280 DATA or defaulted to LFAMM. The,high end is always 32766,(>7FFE).,,
 2290 DATA ,When the EDP and RDT low end,"are entered, 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,value. 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
 2370 DATA "RDT low, high",EDP+2
 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

32750 (>7FEE)
 32766 (>7FFE),

!!!!!!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 - '85

TIPS FROM THE TIGERCUB

#27

Copyright 1985

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes...

The entire contents of Tips from the Tigercub Nos. 1 through 14, with more added, are available as a full disk of 50 programs...

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.

Also Now Available! New Catalog #6, for \$1 which is deductible from your first order. Describes 140 original programs for only \$3 each...

If you have my previous catalog, the following are now available in Extended Basic versions - Fast Addition Practice, Submarine Hunt, Rithmatik, Mawaland (also now available in Basic with Speech), Long Division Cryptograms, Miss Spell, Scrambulation, Bargraffer, Squinch, Dry Gulch, Name That Tune, Scrub, Midnight Trail, Niabo, Kindertimes, Optical Illusion, Bazoo, Synonymy, Speeder Reader, Changeroo, Glunk, Fraction Math, Three Buckets Puzzle, Roman Numbers, Match A

Patch, Kindertimes, I & E Spelling, Casting Out Nines, Haunted Graveyard, Spalling Teacher, 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 cassette - 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 <>170 THEN CALL INIT
120 OPTION BASE 1 :: DIM P% (127),V(127,3):: CALL LOAD(-31006,16):: ON ERROR 130 :: GOTO 160
130 DISPLAY AT(12,9)ERASE ALL:"I/O ERROR" :: RUN 100
140 @,@,A,A+,B,C,D+,FLAG,I,J,K,KD,KN,M,N%,N%,NN,P,P%,P%\$(),PP,PP%,Q%,S,ST,T\$(),TT,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(8196,63,240):: CALL LOAD(16376,67,85,82,83,79,82,48,8)
170 CALL LOAD(12288,129,195,126,165,129,153,102,40)
180 CALL LOAD(12296,2,0,3,240,2,1,48,0,2,2,0,0,4,32,32,36,4,91):: 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 200 AND THE FIRST STATEMENT IN 210
280 ! DISPLAY AT(12,6):"DISK ? (1-3):" :: ACCEPT AT(12,19)
SIZE(-1)VALIDATE("123"):0% :: 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 F ilename 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)S IZE(-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+4,28):"Y"
410 CALL KEY(0,KN,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(12,25)SIZE(4):VT
450 FOR @=1 TO 10 :: DISPLAY AT(12,25)SIZE(1):CHR\$(30):: DISPLAY AT(12,25)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$(KD)
570 CLOSE #1
580 CALL VCHAR(1,3,32,672)::
NN=0 :: X=0 :: FLAG=0 :: GO
TO 260
590 IF K<1 OR K>127 OR LEN(P
6$(K))=0 THEN 430
600 IF ABS(V(K,1))=5 OR ABS(
V(K,1))=4 AND V(K,2)=254 THE
M 640
610 DISPLAY AT(12,1)ERASE AL
L:"Print to ? S": "(P)rinte
r?": "(S)creen?" :: ACCEPT AT
(12,12)SIZE(-1)VALIDATE("PS"
):Q$ :: IF Q$="S" THEN PP=0
:: GOTO 630
620 DISPLAY AT(12,1)ERASE AL
L:"PRINTER? PIQ" :: ACCEPT A
T(12,10)SIZE(-18):P$ :: 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$(K)),1)="3" THEN D
ISPLAY AT(12,1)ERASE ALL:"RE
TURN TO BASIC AND LOAD BY:"
TYPING OLD ";D$&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$

```

```

P6$(K):: CALL LOAD(C,LEN(A$)
)
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$(K),INPUT
,INTERNAL,FIXED :: J=0 :: 50
TO 770
760 OPEN #2:D$&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 :: @0=ASC(S
EG$(M$,Y,1)): IF @0<32 OR @
0>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? PIQ" :: ACCEPT AT(24
,15)SIZE(-14):PP$ :: OPEN #2
:PP$ :: PRINT #2:SEG$(D$,1,4
)%" - Diskname="&M$
850 PRINT #2:RPT$(" ",28):"A
vailable=";350-VT;"Used=";VT
:RPT$(" ",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 dup 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
0 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$=M$%2
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(28,1):" " :: DISPLAY A
T(22,1):" "
170 P=2 :: FOR J=1 TO 4 :: A
CCEPT AT(12,P)VALIDATE("0+*
/")SIZE(1):S$
180 IF S$="0" THEN 200 ELSE
IF S$="+" 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(28,1):"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

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 into 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 vacant 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 questionnaire 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, recommendations, etc. . We also need help. Please consider joining the Board, a committee, or a SIG. And please complete the annual questionnaire, it is the best way we have of judging current interests. In our reviewing our 1981 questionnaire 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 modify the screens of TI-Runner. He will demonstrate his program.

* A discussion entitled "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 on 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

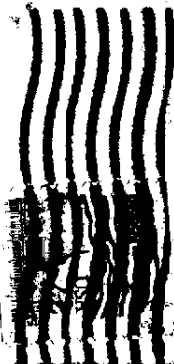
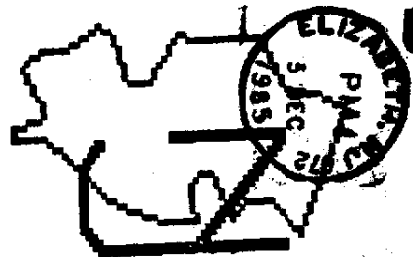
Take Garden State Parkway to exit 131, bear right toward Metuchen on Route 27 (Middlesex Ave.) until you reach the fourth traffic light (Main St.). You will have passed the Library. Turn right at light on to Main St., go one block and turn right onto Library Place, proceed half way up block; Library 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.

Nel Gery
49 Pine Grove Ave.
Somerset, NJ 08873

DALLAS T. 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, 1985

FOR FURTHER INFORMATION

CHAIRMAN - STEVE CITRON 201-686-5619
981 TOXLEY AVE. UNION NJ 07033

CO-CHAIRMAN JEANNETTE SHADER 201-929-0522
201-929-3121 886