

THE NATIONAL NINETY-NINER

VOL II - NO. 6 - JUNE, 1985

COPYRIGHT 1985 BY

\$1.50

THE 99ER'S ASSOCIATION
3535 SO.H ST., #93
BAKERSFIELD, CALIF. 93304
(805) 397-4361
DON VEITH - EDITOR/PRESIDENT

CREATED FOR TI 99/4A HOME COMPUTER OWNERS

TABLE OF CONTENTS

<u>SECTION/ARTICLE</u>	<u>AUTHOR</u>	<u>PAGE</u>
ANNOUNCEMENTS		1
9900 MES REAL TIME CLOCK SCHEMATIC	DON VEITH	1
TI-99/4 SPARE PARTS	DON VEITH	1
M + T UTILITYWARE CHANGES NAME	DON THOMSON III	2
FROM THE MAILBOX		2
YOUR PERSONAL COMPUTER STORE		2
PERIPHERAL EXPANSION BOX INTERCONNECT		3
REFERENCE GUIDE FOR TI-WRITER		3
ARTICLES		3
CP/M - PART IV	LEONARD LANIGAN	3
PASCAL NOTES	EDGAR DOHMANN	4
THE TI FORTH DIMENSION	JEFF STANFORD	5
TIGERCUB TIPS # 21	JIM PETERSON	7
REVIEWS		12
FINANCIAL ANALYSIS ON TI COMPUTERS	DARRELL INGOLD	10
MINIWRITER I WORD PROCESSOR	DARRELL INGOLD	10
BITMAC	ED DOHMANN	11
SCHEMATIC		
9900 MICRO-EXPANSION SYSTEM TIME CLOCK	JOHN CLULOW/RON GRIES	12

ANNOUNCEMENTS

9900 MICRO-EXPANSION SYSTEM REAL TIME CLOCK SCHEMATIC

By Don Veith

My apologies to John Clulow for our failure to print the schematic when the article was presented in our January, 1985 issue. The schematic may be found on the Page 12 of this issue. The schematic parts list is reprinted below to assist in construction of the clock. Printouts of the original article are available at our address on the cover page. To obtain a copy of the article, forward a self-addressed, stamped-envelope (SASE) with a note requesting a copy of "A Real Time Clock For The CorComp 9900 Micro-Expansion System" by John Clulow and Ron Gries. Please allow three (3) weeks for the article to return after you mail your request.

Price	Supplier	Cat. No.	Description
\$.49	Radio Shack	270-398	"AAA" Cell Battery Holder (2)
\$.39	"	270-404	"AAA" Cell Battery Holder (1)
\$ 1.95	"	276-158	Grid Board
\$.79	"	272-1437	22 MFD Tantalum Capacitor
\$.99	"	267-1123	Germanium Diodes (pk of 10)
\$.19	"	271-023	1K Ohm Resistor (pk of 2)
\$.19	"	271-049	220K Ohm Resistor (pk of 2)
\$ 2.39	"	278-501	Wire-Wrap wire (spool)
\$ 4.49 ea	"	23-127	NiCad Batteries (need two pks)

For other components...

\$ 9.60	Digi-Key	MMS8167AN	MMS8167 Real Time Clock
\$ 2.70	"	KF386	32.768 KHz Min. Crystal
\$ 4.20	"	C5-22 H4	Ser Edgebd Connector 22/44
\$.54	"	74LS04N	Hex Inverter
\$.45	"	74LS30N	8-input NAND gate
\$ ----	(don't carry)	74LS245N	Bidirectional Octal Bus Drivr
\$.15	"	C8914	14 pin sockets (2 required)
\$.23	"	C8920	20 pin socket
\$.28	"	C8924	24 pin socket

\$An alternate source...

\$12.95	JDR Microdevices	MMS8167	MMS8167 Read Time Clock
\$ 1.95	"	KF386	32.768 KHz Min. Crystal
\$ ----	(don't carry)	-----	H4 Ser Edgebd Connector 22/44
\$.24	"	74LS04N	Hex Inverter
\$.25	"	74LS30N	8-input NAND gate
\$ 1.49	"	74LS245N	Bidirectional Octal Bus Drive
\$.15	"	C8914	14 pin sockets (2 required)
\$.29	"	C8920	20 pin socket
\$.30	"	C8924	24 pin socket

Digi-Key 1-800-344-4539
Highway 32 South
P.O. Box 677
Thief River Falls, MN 56701

JDR Microdevices 1-800-538-5000
1224 S. Bascom Avenue
San Jose, CA 95128
CA Res: 1-800-662-6279

TI-99/4 SPARE PARTS

Lolir Electronics has purchased a large stock of power supplies, wall transformers, keyboards, and video modulators from Texas Instruments. We have dealt with this firm representing Tex-Bug, our local Bakersfield Users Group. The firm's attitude is very positive with a willingness to assist the customer, a rare occurrence in business today. All orders are shipped by UPS on a C.O.D. basis. Orders are processed in a prompt manner. Tex-Bug's orders arrived in exactly five working days as promised. We were especially impressed with the care taken to package the electronic products.

We found the prices to be excellent. A substantial discount was given when ten (10) items of the same type were ordered over the individual unit price. Please contact the firm by telephone to find out what items are still available and their prices. I hope that you have an opportunity to speak with Nancy. She took the time to answer questions and went out of her way to provide assistance beyond what is required.

Contact this fine firm for details on items available and their prices at:

Lolir Electronics Inc.
13933 N. Central Xway
Keystone Center, Ste. 212
Dallas, Tx 75243
(214) 234-8032

M + T UTILITYWARE CHANGES NAME AND MARKETING STRATEGY

By Donald Thomson III

Due to the ever increasing problems of software piracy, software firms either have to drop out of the picture or change their marketing techniques. I have reached the point where I must decide whether to "BAIL OUT" or change my marketing concepts. I have chosen the latter.

Effective July 1, 1985 M + T UTILITYWARE will be changing its name to: THOMSON SOFTWARE (D.I.P.)

The name change is to make it known that I own this company lock, stock, and barrel. There are no partners. The (D.I.P.) you see in the name stands for "DISTRIBUTION INCENTIVE PROGRAM". DIP is a modified form of SHAREWARE, NOT FREEMWARE, which is currently being utilized with the IBM PC. For those not familiar with SHAREWARE, read on. Shareware is a form of distribution where the author of a program encourages people to distribute his programs. Lets say person (A) has legally purchased the program DISK+AID from me. This purchaser is known as a registered owner of the program. The registration number was assigned by me. If person (A) passes the program to person (B), (B) is encouraged to forward a \$20.00 registration fee to me. If that happens, (B) will be assigned a registration number. Person (A), because he has worked to get (B) to register his copy, will receive a \$5.00 FINDERS AWARD. For every program that (A) passes around and gets registered, he will get \$5.00. There is no maximum number of programs (A) can get people to register. Here is where the modification comes in to SHAREWARE. I have gone one step further and have instituted a 2 level FINDERS AWARD. If person (B) passes the program to person (C), and (C) then registers his program, (B) will get his \$5.00 FINDERS AWARD. In addition, (A) will also receive another \$5.00 FINDERS AWARD for this second level. Again there is no limit to the number of programs a person can distribute and get registered. AT THIS TIME, (D.I.P.) ONLY APPLIES TO PROGRAMS OWNED AND COPYRIGHTED BY DONALD M. THOMSON. (D.I.P) DOES NOT APPLY TO PROGRAMS THAT I AM MARKETING FOR OTHER COMPANYS, UNLESS THEY HAVE AGREED TO ALLOW ME MARKET IT UNDER DIP. Right now that applies only to GRAPHX. DO NOT DISTRIBUTE GRAPHX!!!

There will be limited documentation on the disks that leave THOMSON SOFTWARE (D.I.P.) after JULY 1, 1985. The source code will no longer be placed on the same disk as the object code. The documentation will include a DIP registration form outlining how this new concept operates, a sublicense agreement form, and some limited notes on how to use the program. ORIGINAL manuals will be available from me for a cost of \$5.00. This is to cover copying and mailing costs.

I am going one step further and offering something that, as far as I can tell, is not offered by any other company in the United States. I will be offering a sublicense policy of documented source code for persons who wish to put their programming talents to work. This license fee is \$30.00 and covers all the documented source code CURRENTLY held by me. The whole idea is fairly complex. Basically it comes down to people submitting upgrades within the agreement guidelines will be eligible for author royalties. More information the agreement will be included on disks leaving THOMSON SOFTWARE (D.I.P.) after July 1, 1985 or send a self addresses stamped envelope for a copy in the mail. If someone submits a modification and they are not a licensed holder of the source code, their modification will not be considered until the \$30.00 license fee is paid.

The money back warranty that I offer will remain in effect for people who send the \$20.00 registration fee directly to me and I supply them with the disk. I cannot warrant programs that are passed from one person to another because I am unable to determine if the program has been modified or altered. The 30 day warranty on the disk applies only to those people who send their registration fee directly to me and I supply them the disk. For people wanting to get the program directly from me please add an extra \$2.00 to cover the cost of shipping and handling.

This new marketing concept was decided upon because of the incentives it will give people to make money. You will probably not get rich passing the programs around, but if you can get just four people to register their copies, yours will have been totally free. Everything else is YOUR SPENDING MONEY. Imagine, me paying you to be eligible for upgrades and problem support. For those people who perfect modifications to the programs, this concept has potential that could be very profitable.

Remember, the registration fee is what entitles you to receive your FINDERS AWARDS for every program you get registered. An example, you provide a copy of the program to another person. You failed to pay the registration fee to my company. The person who received the program decides to register their copy with me. As a result of your failure to pay the registration fee, you ARE NOT entitled to receive the \$5.00. You are effectively giving me \$5.00 that rightfully would be yours if the registration fee were paid. The current registration fee is \$20.00. This fee applies to programs written in any of the languages available for the TI-99/4A. D.I.P. will take effect formally on JULY 1, 1985. Everyone who has legally purchased a program and is in good standing with THOMSON SOFTWARE (D.I.P.), will be contacted and issued their registration numbers. These people will be getting one registration number for each program they purchased.

I feel this new concept will become a widely used idea in the future. I will be the first to offer it to the TI-99/4A community. I certainly hope people will take advantage of this unusual opportunity. It is one way we are assured of keeping this UNBELIEVABLE computer alive and well. Good luck to all and HAPPY COMPUTING!!

FROM THE MAILBOX

Items appearing in this section were received in the mail unsolicited. We print the information here as a courtesy to our readers. The listing of a firm's offer here does not constitute an endorsement by this publication or THE 99'ERS ASSOCIATION.

YOUR PERSONAL COMPUTER STORE

This firm, located in Galveston, Texas, provides a variety of computer supplies and hardware to Users Groups in the Houston-Galveston area. The letter contained information on prices for Xidex disks, TI RAM memory chips, and printer ribbons.

The firm offers Xidex disks to Users Groups at a cost of \$10.00 for 10 single-sided, single-density for 10 disks. Ten double-sided, double-density diskettes may be purchased for \$12.00. In both cases, sleeves and labels for the diskettes are included.

A plastic library case is included for an additional cost of \$2.30/box. The firm purchases their disks in lots of 1000 each direct from the Kindex company. The diskettes have a lifetime warranty. The firm offers FREE SHIPPING on all orders of 50 diskettes or more from a Users Group. If you wish a reference on these diskettes, the firm suggests checking a rating on the diskette in a BYTE Magazine article in Vol. 9, No. 10, p. 142.

TI RAM memory chips, nine (9) 64K chips per set with a response time of 150 ns, are offered for \$15.00 per set. If four (4) or more sets are ordered, shipping will be prepaid by the firm. For less than four sets of chips, add a \$2.00 shipping charge per order.

Printer ribbons are offered in the colors of red, blue, and purple plus the traditional color black. All ribbons are available at the same cost. The T-416 series ribbons for the Okidata Microline printers are priced at \$2.00 each. These ribbons also fit the Gemini Series of printers. The T/437 ribbon for the Prowriter, C. Itoh 8510, and NEC 8023A-01 printers are \$6.00 each. An order for \$50.00 of any grouping of ribbons is shipped free of charge. ANY COMBINATION OF PRODUCTS WHOSE TOTAL ORDER VALUE EXCEEDS \$50.00 WILL BE SHIPPED FREE. The items may be ordered at the address below:

Your Personal Computer Store
2724-6 61st. Street
Galveston, Tx 77551-1845
(409) 740-3223 or 740-3220

PERIPHERAL EXPANSION ROI INTERCONNECT

The TEN-X Interconnect System alleviates the problems associated with the Expansion System's heavy cable and interface connection. The device plugs into the 99/4A's I/O port and the existing cable plugs in from behind the console. An additional bonus, the keyboard is tilted slightly for comfortable typing. The Interconnect is available in three (3) models. For use with the speech synthesizer (1), without the synthesizer (2), and for anyone interested in the added benefit of a tilted keyboard (3). The cost for models (1) and (2) is \$43.85. Model (3) is listed at \$14.95. You are requested to add \$3.00 for shipping and \$2.00 for C.O.D. Additional items the firm has available for the 99/4 are: parallel printer cable - \$19.95, monitor cable - \$11.85, and an RS-232 modem cable - \$19.95. Contact the firm on the SOURCE, Act # TI6656 or at the address below:

Tex-X Precision
P.O. Box 163
Concord, Ca 94522

REFERENCE GUIDE AND TUTORIAL FOR TI-WRITER

A complete manual is available on disk for TI-Writer. The new manual includes a step-by-step tutorial on using the Text Editor, Text Formatter, and instructions for using Define Prompt and Mail Merge. The manual is written in easy to understand terms and is a reference for the beginner or experienced user. The manual costs \$15.00. It is neither copyrighted or protected. To order the new Reference Guide and Tutorial for T-Writer, send the requested funds to:

Larry Robinson
503 Third Ave.
Bonaire, Ga 31005

ARTICLES

CP/M - PART IV

By Leonard Lanigan

In this report, I will examine some of the more exotic applications that are made available to us by the Morning Star Software (MSS) CP/M card.

The software literature available for the CP/M operating system is as varied as it is large. If you are interested in using such languages as FORTRAN, COBOL and Pascal, all of these compiled languages are available in versions for CP/M machines. Have you given up on ever using the UCSD p-System because you can't find a p-Code card? Take heart! The UCSD p-System is available to run under CP/M (one operating system running under another-how about that), and while some of the superficial details of operation will be different from the TI implementation, the basic features will be the same, and programs should be interchangeable if they are entirely in p-Code.

I must admit that I have done little programming with my TI computers since shortly after I bought my first one (an old-style 99/4) in early 1980. The only language I had available was BASIC, and it ran so slowly that I soon grew discouraged. MSS has given us the key to many new, fast and powerful programming languages. As an example of the power of Pascal, the Personal Pearl database program is written largely in Pascal, with some 8080 assembly code. While UCSD Pascal is not as fast as a true compiled version, it should run 4 to 10 times faster than TI-BASIC.

Ah-but you have spent months mastering BASIC and don't want to start again? There are several versions of BASIC available for the MSS card, from CBASIC (a compiled form of BASIC) to the "standard" of the lot: Microsoft BASIC.

In addition to increased speed, these are the languages of choice for much of the commercial business-oriented software. For example, Business Master II is written in CBASIC.

Last month I mentioned a couple of reviews of the MSS card published in HOME COMPUTER MAGAZINE, and in MICROPENDIUM. While both of these articles are accurate in most respects, they both contain errors, which should be pointed out (I expect I will commit a few errors, and anyone is welcome to point them out, especially to me).

Let's take the review in MICROPENDIUM, as it seems rather more critical than the other. One complaint noted by the reviewer is the lack of software available in the Osborne 1 format. While it may be that most computer stores do not stock programs in this format, some do. In my area there is a professional systems house which carries Osborne 1 software (they also carried the computers and are concerned with customer service). Any software shop should be able to order the software you need in such a standard format, even if they don't keep it on hand. Also, I spoke with Mr. Roch of Elliam Associates, and all their commercial and public domain software is available in the SS/SD format. It is necessary to be sure that software is written in 8080/8085 code (as most is), and that the Z80 CPU is not specifically required. The descriptive literature should clearly state if the Z80 is needed, as it is not a standard CP/M requirement.

Patricia Swift's review in HCM is particularly interesting in the light of her review of the Microsoft CP/M card for the Apple computer (in the same issue of HCM). It appears that Patricia is well versed in CP/M and 8080 assembly programming, and her report fairly glows in its praise. I noted only two errors in her article, both of which may stem from the documentation she was provided. First, she has the old price listed (\$595), when the card is now available for about \$445 through your user's group. Second, the correct key sequence to escape CP/M and return to the title screen is to press <ctrl><ctn><enter> all simultaneously. This has worked properly for me every time, and is the procedure noted in my user guide. Other than these small items, Patricia's review seems quite exhaustive and accurate. It appears that she did not have the SETUP program available, while it is included in the material I have. I use it primarily to change screen color (I prefer white-on-black) for my monochrome monitor, but there are several other options available, including the ability to rename various I/O ports for printers, modems, etc., and the ability to modify both 8085 and 9900 memory contents. I have not yet made use of these features beyond setting up the RS-232/1 port for use with my printer, and noting the presence of the others, but the fact that they are included in the system speaks well for the design and execution of the MSS card.

I have finally made Personal Pearl crash. This should not be surprising, as I have been intentionally tweaking its nose for over a month trying to make it fail. I have found two things that give it indigestion, though only one results in a crash. In the first case I directed the program to search for an item that was not in the database. The program went through all the motions of finding the requested item, including 3 disk swaps and several minutes of thinking time, then brought up a completely blank screen! I would have expected such a sophisticated program to know immediately that I had sent it on a futile errand, or at the very least, to give a message that my instructions had been carried out, and that the item was not there. As it was, only the absolute blankness of the screen (no prompts, nothing), and the total cessation of disk activity let me know that Pearl had finally given up the chase. The "crash" was caused by using a write protected data disk. Instead of an error message from Pearl such as "DATA DISK WRITE PROTECTED", Pearl quit and left me the delightful CP/M error message "BDOS ERROR ON B". From this point I had to reload Pearl, and reenter my data.

This really is not such a bad performance. I had been doing such things as inserting the wrong disks, or using the wrong drive, and each time the program gently warned me that something was amiss. It really is a well-written program, and quite powerful.

If there are specific questions that you need answered, or particular topics you feel should be examined in this series, please let me know. You can reach me at the NATIONAL NINTY-NINER address, but your best bet is to write directly to:

Leonard Lanigan
P.O. Box 358
Browns Valley, CA 95918

Please include an SASE if you require a personal response, otherwise stay tuned to the NATIONAL NINTY-NINER.

PASCAL NOTES

By Edgar Doheann - JSC Users Group (JUG)

Last month we looked at the Insrt, Xch, and Quit commands of the P-System Editor. This month we will complete our series on the Editor by looking at the rest of the commands that are available. I feel the commands discussed last month are the most useful. This month I will discuss the remainder of the commands in alphabetic order.

As with most software tools (compilers, editors, etc.), I recommend that you not try to master all there is to know about the P-System Editor at one time. Learn just enough to get started creating and editing files, then as you get comfortable with the modes you have learned, start expanding your familiarity by trying some of the additional commands presented here.

ADJUST -- This command lets you adjust the indentation of lines on the display. This is a useful feature if you are using the Editor as a general word processor. If you are using it to enter source files I think it is a waste of time. I prefer to use a "Nester" type of program to automatically adjust source code files into a consistent format.

Copy -- This is a very useful command which lets you copy text from the copy buffer or another file into the file you are editing. Position the cursor to the point where you want text inserted then type C to enter the Copy mode. Once in the Copy mode, you can type B for Buffer or F for File.

The copy buffer is a little tricky to use. Information is placed into this buffer from the Dlet, Insrt, and Zap commands. In Dlet, the buffer receives any text that is marked for deletion, regardless of whether or not it is actually deleted. With Insrt, the buffer receives only the text that is actually inserted. With Zap, the buffer receives the deleted text.

The File command lets you copy all or part of another text file into the file currently being edited. If you only want to copy part of the file, the copy file must have been previously edited and markers placed in desired segments using the Set command. Then by using the Marker option of the F copy command, only a portion of the file will be copied.

Dlet -- This command lets you delete text from the file being edited. This is a useful feature if you are starting with an existing program and need to make wholesale changes to it. If you only need to delete or change a few lines or characters, I prefer to use the Insrt or Xch modes.

Find -- This is a handy feature which locates specified words or characters within the file. Unfortunately, this feature has two confusing modes of operation, Literal and Token. Literal mode searches for strings that may be part of larger strings or contain spaces. Token mode searches for strings that are preceded and followed by a space or punctuation mark. Since you have to enclose your string with delimiters (characters that are not letters or numbers and not in the string itself), I fail to see why two modes of searching are even necessary in the first place.

The other confusing thing is that the current mode is implied, not stated. If the Find prompt line shows L(it), then you are in Token mode and must press L if you wish to change to Literal mode. Similarly, if the prompt line shows T(ok) then you are in Literal mode and must press T to switch to Token mode.

Jump -- This command lets you move the cursor to a specified point in the file to make changes more easily. This is especially handy if you are working with large files that take a long time to move through just using the arrow keys. The Jump command lets you move the cursor to the beginning or end of the file or to a specific spot marked with a Marker.

Kol -- This command lets you move text left or right one column at a time. Like the ADJUST command, I only find this one useful in general word processing applications.

Margin -- Same comment as Kol and ADJUST. This one lets you adjust text to fit within specified margins.

Page -- This command displays the next page of the file. Like Jump, this is handy for moving to a different area of a large file. By using a repeat factor, you can skip to any page in the file. Paging can go forward or back by using Set to set a global direction indicator.

Rplc -- This command works similar to the Find command. However, in addition to merely searching and locating a string, this command will let you specify a new string to replace the old one with.

Set -- This is a useful command which lets you change some of the characteristics of the Editor. Mainly it lets you set Environment options and insert Markers into a file. The Environment mode displays a list of current options and lets you change them if desired.

Auto indent specifies whether each line of text will be automatically indented the same as the previous line (TRUE) or if the cursor is to return to the first column of the line (FALSE). Filling specifies whether word-wrap is to occur (TRUE) or if a carriage return is required to end the line (FALSE). The Left, Right, and Para Margin options let you set the default text margins. The Command ch option specifies the character which indicates that margins should not be adjusted on specific lines of text. Tabstops lets you set the tab positions. Token Def specifies whether Token (TRUE) or Literal (FALSE) will be the default for the Rplc and Find commands.

The Marker command lets you set markers in your file to make it easier to copy information from a file or to Jump to a specific area of a file. To set a marker, be sure the cursor was at the desired spot before entering the Set command. Once in the set command, type M to select Marker, then type a 1 to 8 character name as th marker to be inserted at the spot where the cursor was. Each file can have a maximum of 20 markers. I recommend that you select a consistent marker naming convention and use it for all files to make it easy to remember marker names. I use the numbers 1 to 20, what can be easier?

Zap -- This command deletes all text between the start of any information that has been found, replaced, or inserted and the current cursor position. Before using this command, you must first use the Find, Rplc, or Insrt commands as the immediately preceding command. The deleted text is stored in the copy buffer so you can recover it if you make a mistake. If you need to recover it, be sure to use the Copy command right away before you change the contents of the copy buffer with some other command.

In summary, I find the Insrt and Xch commands to be the most useful (along with Quit). The next group of useful commands are Copy, Find, Page, and Jump. I never use ADJUST, Kol, or Margin but if you use this Editor as a general word processor, you might find them useful. I rarely use Dlet or Zap but that is probably a reflection on the type of editing I do and personal preference more than anything else. I also do not use Set very much, I find the default options to be satisfactory for most of my work and I do not use markers too often.

My general recommendation for learning the Editor is to use it. Start with the basic Insrt, Xch, and Quit commands. Once you are comfortable with them, then practice some of the other commands. Do not try to memorize everything about the Editor, as you use it more and more you will begin to remember more and more of the details without having to use the manual as a reference. The main thing to do is become familiar with a set of commands you like to use and develop techniques of creating, editing, and saving files that you are comfortable with.

The II Forth Discussion

By Jeff Stanford

Now that I have covered the fundamentals of getting Forth into your II-99 and how the two editors function, I can now start explaining how Forth works. In this part of my tutorial, I will cover the different types of numbers available in Forth and how to convert equations from the "STANDARD" form into the form Forth uses. When Forth boots up, it provides several types of words to manipulate integer numbers and some words to accept and display single characters (or a string of characters with a little work). This may seem a bit crude and primitive and you are right, but they form the basic tools needed for writing I/O routines to fit your personal needs.

The three basic types of integers Forth uses are BYTE (8 bits), WORDS (2 bytes / 16 bits), and DOUBLE-WORDS (4 bytes / 32 bits). When the -FLOAT load option is used, you obtain access to the floating point routines in the console's read only memory (ROM) which provides the same accuracy as BASIC or up to sixteen (16) decimal places. A floating point number takes up eight bytes of memory (64 bits!). Each type of number has its own range of numbers it can express. The following table give the ranges of numbers available for each type of integer and for floating point numbers.

type	ranges of values	
	as a signed number	as a unsigned number
BYTE	-128 --> 127	0 --> 255
WORD	-32768 --> 32767	0 --> 65535
DOUBLE-WORD	-2147483648 --> 2147483647	0 --> 4294967295
FLOATING POINT	-9.999999999999999 E 127 --> -1. E -128, 0.0, 1. E -128 --> 9.999999999999999 E 127	N/A

As the table shows, increasing the number of bits for an integer also increases the range of numbers you can express. So, it would seem that a floating point number is the best choice for general use. I believe after trying floating point numbers, you will find Forth runs almost as slow as similar programs written in Basic. This is due to the time needed to manipulate the eight byte numbers. Basic on the TI-99 is slow for this reason when compared to other computers. All calculations are done with floating point numbers. Ask what your friends get for the square root of four. You will learn that the TI wins for being very accurate. Another problem with using floating point numbers arises when you want to do bit-map graphics. The floating point calculations for transcendental functions (SIN, COS, LOG, ...) are done within the video display processor (VDP) memory (the 16K of memory in the console) which happens to be in the middle of the table to define colors in the bit-map mode. The TI-Forth manual explains this problem in Chapter Seven (Page Four (4)) and how to work around it if you must use floating point numbers and bit-map graphics. Now, just because I do not recommend the use of floating point numbers, it does not mean that fractional numbers should be forgotten. There is a method call "fixed point" notation which gives you fractional numbers and still keeps the speed of integer calculations. While a number is in the computer, it is kept biased by some number. When it needs to be displayed, you just insert the decimal point in the proper location. For example, say I am interested in working with money and want to represent five dollars and thirty five cents (\$5.35). The simplest way is to use a bias of 100 and express the value as 535. When you are ready to print the number, remember to insert the decimal point two places from the right. Another way of looking at this type of number is to write it as an integer multiplied by 10(-2). In my example, it would be written 535 * 10(-2). Every book written about Forth has something to say about fixed point notation. I suggest you consult one of these books in your personal collection for more information regarding the use of fixed point numbers.

In the math world there are three ways to write algebraic equations. The most popular is called "infix notation". The operator (+ - * / ...) is located between the two operands. It is used by most people and computer languages to express equations. An example of a infix equation is 'X = 5 + 2'. The other two methods are "prefix notation" and "postfix notation." NOTE, the name implies where the operator goes in relation to the two operands. Prefix means the operator is written in front of the two operands (X = + 5 2) Postfix means the operator is written after the two operators (X = 5 2 +). Another name for these two methods are 'Polish Notation' and 'Reverse Polish Notation' (RPN for HP fans) named for the nationality of the mathematician, Jan Lukaciewicz, who first proposed it. Forth uses postfix notation or RPN to do its calculations. RPN is easier for a computer to manipulate allowing Forth's compiler to be compact and simple. In fact, equations written in almost any computer language are converted into RPN when a program is compiled. It is more efficient for the computer to handle equations written in this manner. Writing equations in Forth is almost as simple as solving the equation by hand with pencil and paper. For simple equations, you simply switch the positions of the operator and the second operand. For example: 5 + 7 becomes 5 7 +. For more complex equations which contain parentheses to direct the order in which the calculations takes place, you simply follow the rule for solving them by hand: 'start with the innermost set of parentheses and work your way outward', for example:

$$(((3 + 7) * 9 + 7) - (5 * 13)) / 4 \text{ becomes}$$

$$3 7 + 9 * 7 + 5 13 * - 4 /$$

Reading the RPN version of the equation from left to right, you can solve the equation. Every time an operator is encountered, you already know what the two operands are. A step by step evaluation is shown below:

- 1: 3 7 + 9 * 7 + 5 13 * - 4 /
- 2: 10 9 * 7 + 5 13 * - 4 /
- 3: 90 7 + 5 13 * - 4 /
- 4: 97 5 13 * - 4 /
- 5: 97 65 - 4 / (note that math operators work only on the two numbers preceding it)
- 6: 32 4 /
- 7: 8

Please note there is no need for parentheses in the RPN version of the equation. Their use is not allowed in Forth equations. Parenthesis are used to place comments into a Forth program. Another convention you should be familiar with is how Forth uses a stack for it's calculations. A stack works like a spring loaded plate dispenser found in cafeterias. As more plates are placed on the plates already in the dispenser, the whole stack of plates slides downward. As plates are removed from the dispenser, the plates left in the dispenser slide upward. The topmost plate in the dispenser is always easily reached by customers. Another name for a stack is a LIFO, which is an acronym for 'Last In - First Out'. This applies to the plate dispenser. The last plate placed in the dispenser is the first plate to be used by a customer. The stack in Forth works in a similar manner. You 'push' numbers onto the stack, and then retrieve them from the stack by 'popping' each number, starting with the last one you pushed onto the stack and working your way backwards until the first number entered is recovered from the stack. For a quick example of how a stack works, try this example on your computer. With Forth running, type:

1 2 3 4 (enter) - Forth should respond with 'ok'.

What you just did was to push the numbers '1 2 3 4' onto the stack. '.S' is a handy little Forth word which is added when you used the '-DUMP' menu option. It displays the contents of the stack without changing the stack. So when you load in the '-DUMP' option and type '.S (enter)', the entire contents of the stack is displayed. The last four numbers in the list should

be the ones you typed in. To recover the numbers you pushed onto the stack, use the Forth word '.' (DOT) which takes the number from the top of the stack and displays it on the screen. When you use '.' (enter), you should get the last number typed or a four (4). If you continue using '.', each number entered will be displayed in the reverse order of their entry into the computer (remember the meaning of LIFO). If you use '.' when there are no numbers in the stack, an 'empty stack' error message will appear. Almost every Forth word is defined in three parts: 1) what inputs are needed on the stack, 2) what processing is done, and 3) what outputs are left on the stack. For example the Forth word '+' expects two numbers to be on the stack which are added together with their sum left on the stack as an output. The TI-Forth manual and most books about Forth use the following method to illustrate how the stack is affected by a word. For example, the effect on the stack for the word '+' (plus) is shown as '(n1 n2 --- n3)'. It implies that two numbers are used from the stack as inputs (n1 n2) and the resulting output (n3) is left in the inputs place on the stack. Along with this input/output diagram, a brief statement explains how the inputs are use and/or how outputs are generated (this is the process section). This same input/output diagram is used to show the effects on the stack for most math operators. The only difference is in the process section ('-' which computes the difference of the two inputs, ...). Now, let us look at my example differently and see how the stack varies as the equation is evaluated. The equation in RPN form was '3 7 + 9 * 7 + 5 13 * - 4 /' and the effects on the stack word by word are:

WORD	STACK CONTENTS	NOTES
		(START)
3	3	PUSH 3 ONTO STACK
7	3 7	PUSH 7 ONTO STACK
+	10	ADD TOP TWO NUMBERS AND RETURN SUM TO STACK
9	10 9	PUSH 9 ONTO STACK
*	90	MULTIPLY TOP TWO NUMBERS AND RETURN PRODUCT TO STACK
7	90 7	PUSH 7 ONTO STACK
+	97	ADD TOP TWO NUMBERS AND RETURN SUM TO STACK
5	97 5	PUSH 5 ONTO STACK
13	97 5 13	PUSH 13 ONTO STACK
*	97 65	MULTIPLY TOP TWO NUMBERS AND RETURN SUM TO STACK
-	32	SUBTRACT TOP TWO NUMBERS AND RETURN DIFFERENCE TO STACK
4	32 4	PUSH 4 UNTO STACK
/	8	DIVIDE TOP TWO NUMBERS AND RETURN QUOTIENT TO STACK

As you can see, the final result of the equation remains on the stack. To print the results, you type '.' (enter) and the results will be displayed on the screen. Here is a brief list of commands which manipulate Forth's stacks:

WORD	INPUTS/OUTPUTS	DESCRIPTIONS
*	(n1 n2 --- n3)	compute the product of n1 and n2 (n3 = n1*n2)
+	(n1 n2 --- n3)	compute the sum of n1 and n2 (n3 = n1+n2)
-	(n1 n2 --- n3)	compute the difference of n1 and n2 (n3 = n1-n2)
/	(n1 n2 --- n3)	compute the integer quotient of n1 and n2 (n3 = n1/n2)
DUP	(n1 --- n1 n1)	make a copy of the number on top of the stack
SWAP	(n1 n2 --- n2 n1)	exchange the top two numbers on the stack
OVER	(n1 n2 --- n1 n2 n1)	make a copy of the second number on the stack
ROT	(n1 n2 n3 --- n2 n3 n1)	rotate the third number to the top of the stack

All of these words work within a program or in the command mode direct from the console keyboard. You can use your TI like a calculator. Simply type in the equation and press enter (remember to use '.' to print your results). For example, if you want to compute the area of a rectangle 2 feet by 3 feet in square inches, tackle the problem in two ways; 1) calculate the area in square feet, then convert square feet to square inches, or 2) convert each side of the rectangle from feet to inches and then calculate the area in square inches.

Here are both methods explained so you can type them into the 99/4 using Forth:

1) 2 3 * 144 *

2) 2 12 * 3 12 * *

(Remember 12 inches = 1 foot and 144 square inches = 1 square foot)

Next time I will show how to work with variables in Forth.

TIPS FROM THE TIGERCUB #21

Copyright 1985
TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software. The entire contents of Tips from the Tigercub Nos. 1 through 14, with more added, are now available as a full disk of 50 programs, routines and files for just \$15.00 postpaid! Nuts Bolts is a disk full of 100 (that's right, 100!) XBasic utility subprograms in MERGE format, ready for you to merge into your own programs. Contents include 13 type fonts, 14 text display routines, 12 sorts and shuffles, 9 data saving and reading routines, 9 wipes, 8 pauses, 6 music, 2 protection, etc., and now also a tutorial on using subprograms, all for just \$19.95 postpaid! And I have about 140 other absolutely original programs in Basic and XBasic at only \$3.00 each! (plus \$1.50 per order for cassette, packing and postage, or \$3.00 for diskette, PPM) Some users groups charge their members that much for public domain programs! I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

I thought that my 28-Column Converter, as published in Tips #18, was finally foolproof, but someone found a way to print a program incorrectly with it! I'm sure you know that characters 127-143, and on up to 159 in Basic, can be redefined and used in graphics. You probably also know that these redefined characters can be put into PRINT or DISPLAY AT statements, by holding down the CTRL key as you type them. If you load a program containing such redefined characters and LIST it, they will appear as blanks. If you RUN the program, so that they are redefined by the CALL CHAR statements, and then LIST it again, they will show up in their redefined form - but if you print out the program on your printer, they will still appear as blanks. So, before you publish a program, it's a good idea to RUN it and LIST it, and look for any of those gremlins. If you do want to publish such a program, this fix will take care of it by underlining all characters that must be typed with CTRL down (except that lower case v is typed with FCTN down). It's slow, so only use it when you need to.

```

190 IF Q$="E" THEN 195 :: PR
INT #2:".TL 126:94;" :: PRIN
T #2:".TL 123:64;" :: PRINT
#2:".TL 125:38;" :: PRINT #2
:".TL 124:42;" :: PRINT #2:"
.

195 PRINT "Does the program
contain":"redefined characte
rs above":"ASCII 126? (Y/N)"
196 ACCEPT AT(24,1)VALIDATE(
"YN"):Q$
282 IF Q$="N" THEN 290
283 FOR J=1 TO LEN(L$)
284 A=ASC(SE6$(L$,J,1)):: IF
A<127 THEN L2$=L2$&CHR$(A):
: GOTO 288
285 IF A=127 THEN A=118 ELSE
IF A=128 THEN A=44 ELSE IF
6 THEN A=59 ELSE IF A=157 TH
EN A=61 ELSE IF A=158 THEN A
=56 ELSE IF A=159 THEN A=57
ELSE A=A-64
286 L2$=L2$&CHR$(27)&CHR$(45
)&CHR$(1)&CHR$(A)&CHR$(27)&C
HR$(45)&CHR$(0)
288 NEXT J :: L$=L2$ :: L2$=
""

```

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

TIGERCUB CHALLENGE

```

100!The Unprintable Unkeyabl
e Program!
110!To shuffle the numbers 1
to 255 into a random sequen
ce without duplication
120!The strings contain the
ASCII characters 1 to 127 an
d 128 to 255
130!Most of the ASCII charac
ters below 32 or above 159 c
annot be input from the keyb
oard
140!So how was this program
programmed?
150 M$=""
!""$$%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKL
MNOPQRSTUVWXYZ[\]^_`abcdefg
hijklmnopqrstuvwxyz{~} "
160 M2$=""
170 M$=M$&M2$
180 L=LEN(M$):: RANDOMIZE ::
x=int(L&RND+1):: N=ASC(SE6$
(M$,X,1)):: M$=SE6$(M$,1,X-1
)&SE6$(M$,X+1,LEN(M$))
190 PRINT N:: IF LEN(M$)=0
THEN STOP ELSE 130

```

GROCERY SHOPPING LIST

Are you desperate for some way to convince your wife that your computer and PEB and printer and all are not just a too-expensive plaything? Maybe this will do the job. The first thing to do is to prepare a file of the grocery items she might want to buy. It will be especially useful if you can list the items in the sequence in which she will come to them in the aisles of her favorite store. This little program will set up the file. Type END when you are finished.

```

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

```

If you have TI-Writer, you can also use that to create the file, edit it and add to it - but BE SURE to delete all the carriage return symbols and any blank lines at the end. Save it under the filename BUYLIST. Next, this program will hopefully get your wife to actually sit down at the keyboard and try out your computer. It will go through the list and ask if she wants to buy. If she types in any quantity other than 0, it will output the item name and quantity to the printer. At the end, she will be given the opportunity to add any other items.

```

100 CALL CLEAR
110 OPEN #1:"DSK1.BUYLIST",I
NPUT
120 OPEN #2:"PIO"
130 LINPUT #1:A$
140 IF EOF(1)THEN 210
150 DISPLAY AT(12,1):A$
160 DISPLAY AT(12,LEN(A$)+2)
:"0"
170 ACCEPT AT(12,LEN(A$)+2)S
IZE(-4):Q
180 IF Q=0 THEN 130
190 PRINT #2:A$&" "&STR$(Q)&
" "&CHR$(175)
200 GOTO 130
210 DISPLAY AT(12,1):"ADDITI
ONAL? Y"
220 ACCEPT AT(12,13)VALIDATE
("YN")SIZE(-1):Q$
230 IF Q$="N" THEN 300
240 DISPLAY AT(12,1):"ITEM?"
250 ACCEPT AT(12,7):A$
260 DISPLAY AT(14,1):"QUANTI
TY?"
270 ACCEPT AT(14,11):Q
280 PRINT #2:A$&" "&STR$(Q)&
" "&CHR$(175)
290 GOTO 210
300 CLOSE #1
310 CLOSE #2

```

The list will be in enlarged print, so that no one in the store will see her putting on her reading spectacles. And after each item and quantity is a blank square to be checked off when she picks up the item. You might also point out that she could use the checkoff blocks to mark the items she has coupons for, and she could jot down prices on it to be sure she isn't cheated at the checkout counter, or to shop for better bargains elsewhere.

The program is set up for the Gemini printer. You may need to change the "PIO" to the name of your printer, and other printers may not have the open block character CHR\$(175) available. Of course, you can also use this program for more important things, such as shopping for computer software....! If you type the period key while holding down the CTRL key, the printer interprets the resulting blank space as CHR\$(27), even though the computer knows it is really CHR\$(155). Since CHR\$(27) is the ESC or "escape code" which tells the printer to interpret the following characters as function command codes, you can for instance set up the printer for emphasized double-struck double-width underlined italics by OPEN #1:"PIO" :: PRINT #1:" E G W"&CHR\$(1)&" -"&CHR\$(1)&" 4 ", using CTRL . in the blanks. I have been overlooking another very useful feature, the skip-over perforation. PRINT #1:" N"&CHR\$(6), again with CTRL to advance to the top of the next page when there are only 6 lines left at the bottom of the page (providing that you started at the top, of course). This makes it possible to LIST "PIO" a program, or PF PIO from TI-Writer Editor, without printing right across the perforations.

Ghosts! Did you ever read data from a file, and find that you were getting data from a file that was no longer on the disk? It can happen, at least if you are reading from a RELATIVE file in the UPDATE mode. When you delete a file, only its address is actually deleted - the data remains on the disk until it is overwritten by a new file. If the new file is shorter than the old one, and you try to read beyond the end of the file, you may awaken the ghost! Are you making use of those special characters that are available on your Gemini printer? You didn't know about them? Try this.

```
100 OPEN #1:"PIO" :: 110          and then hold down the FCTN
PRINT #1:" (hold down the      key and type <>/0;BHJKLMBQY
CTRL key and type 1234567/    ) "
```

Type RUN. Surprised? Some of those can be very useful, such as

the true division sign that you get with FCTN H. There are many more of these that you can access by CHR\$. For a complete list of them and their CHR\$ codes, run this -

```
100 OPEN #1:"PIO" :: FOR
CH=160 TO 254 :: PRINT
#1:CH;CHR$(CH);: NEXT CH ::
CLOSE #1
```

Unfortunately, these can't be used out of TI-Writer. Here's a handy little routine to practice on your typing.

```
100 CALL CLEAR
110 CALL CHAR(94,"3C4299A1A1
99423C")
120 CALL SCREEN(5)
130 CALL VCHAR(1,31,1,96)
140 CALL COLOR(1,8,16)
150 FOR SET=2 TO 12
160 CALL COLOR(SET,2,16)
170 NEXT SET
180 PRINT TAB(10);"TIGERCUB"
: TAB(8);"TOUCH-TYPING": : T
AB(11);"TUTOR": : TAB(9);" T
iger cub Software": :
190 REM by Jim Peterson
200 PRINT " Watch the scree
n, not the": " keyboard!": :
Letters and numbers will"
210 PRINT " appear on the sc
reen grid": " in position cor
responding": " to their keybo
ard position.": : " Type the
m and they will"
220 PRINT " disappear.": : :
" Press any key"
230 CALL KEY(O,K,ST)
240 IF ST=0 THEN 230
250 CALL CLEAR
260 CALL CHAR(32,"FF80808080
80808")
270 CALL VCHAR(1,30,1,192)
280 CALL HCHAR(14,1,1,384)
290 CALL VCHAR(1,4,1,14):: C
ALL VCHAR(5,6,1,11):: CALL V
CHAR(8,7,1,6):: CALL VCHAR(1
1,8,1,3):: CALL VCHAR(8,29,1
,6)
300 CALL VCHAR(11,28,1,3)
310 CALL CHAR(48,"003A44C54
644488")
320 KEY$="1234567890=QWERTYU
IOP/ASDFGHJKL;"&CHR$(13)&"ZX
CVBNM,."
330 RANDOMIZE
340 K=ASC(SEG$(KEY$,INT(42#R
ND+1),1))
350 GOSUB 370
360 GOTO 420
370 X=POS(KEY$,CHR$(K),1)
380 Y=ABS(X>11)+ABS(X>22)+AB
S(X>33)+1
390 R=Y#3
400 C=((X-ABS(Y>1))*(Y-1))
*(2)+4*Y
410 RETURN
420 CALL HCHAR(R,C,K)
430 CALL KEY(3,K,ST)
440 IF ST=0 THEN 430
450 GOSUB 370
460 CALL BCHAR(R,C,6)
470 IF B<>32 THEN 500
480 CALL SOUND(-100,110,0,-4
,0)
490 GOTO 340
500 CALL HCHAR(R,C,32)
510 CALL SOUND(-100,1000,0,1
005,0)
```

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

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

MEMORY FULL -- Jim Peterson

REVIEWS

FINANCIAL ANALYSIS ON TI COMPUTERS

Written by Joseph & Susan Berk

Review by Darrell Ingold

Does 'Perpetuity Analysis' or 'Multiple Linear Regression Analysis' or perhaps 'Pro Forma Financial Forecasting Analysis' make your heart go pitter patter? If so I think I've got just the book for you; if NOT, stick around...you might just learn something (like I did)!!

As the title of this book and some of the sample programs listed above might well indicate, this book is for the serious user of the Texas Instruments computer. This is further evidenced in the fact that all the programs listed are also (with the inclusion of only one additional line) usable on the TI Professional computer. The Berks approach financial matters in a no-nonsense manner that make each program concise and readily understood by not only those who already know analysis, but by anyone who has a reasonable math background and even a limited understanding of business. The question that may then pop into your head is 'Why do I need this book?'. The simple answer is, you MAY not...BUT do you ever buy stocks or plan to invest any money toward your retirement? How about plans to start your own little business? Do you own rental property? Open a savings account? Purchase an IRA?

As you can see everyone can use a little financial analysis...whether they think they need it or not. We all do financial analysis to one degree or another every time we make a decision on how to put away a little cash for the proverbial rainy day. Have you heard the statement, 'interest compounded quarterly' or 'interest compounded daily'? Just what is the difference in terms we can understand, namely \$\$? If you want to have \$100,000 in the bank when you retire how can you effectively determine how much money at what interest for how long it will take to reach your goal? The answer simply is FINANCIAL ANALYSIS!

Have I sold you on your need for financial analysis? Good! Now let's get down to the specifics of just how good this book actually is.

I am indeed amazed at the way Joseph and Susan Berk managed to present what many people feel are complex financial analysis, in such a concise, understandable way. Each chapter is treated separately and explained at the beginning so that the reader knows what the goal of the analysis is and what it is used for before anything else is mentioned. There are, of course, some terms used that may be foreign to the non-business person but by completely reading the explanation most people will still be able to perceive the concept. If there are several concepts involved, each is treated separately and then tied to the whole. Following the description comes a step-by-step presentation of what actually appears on the computer screen along with some sample input and output based on the preceding narrative. Sometimes there are several examples of specific applications and these are followed by a full listing of all the formulas used in the program. This allows you to isolate the individual formulas for closer study (if you wish) or helps the more experienced person grasp the method used. Lastly in each chapter you will find the actual program listing. These listings are printed as a single column, in very clear, easy to read print with common line sequencing of 10. These programs are remarkably short with length ranging typically from 50 to 150 lines. There are a couple that approach 200 but some are less; the lines are single statements and are generally brief. The overall length could be shortened considerably by using multiple line statements in Extended Basic but they are presented in TI Basic so as to be available to a broader spectrum of users.

After a conversation with the authors, it is expected that in the very near future, all the programs will be available on disk and cassette. They will come as a package deal directly from the Berks. The disk/cassette will also be available to the person who already has their book too. Watch for details in the Announcements section of the NATIONAL NINETY-NINER!

MINIWRITER I WORD PROCESSOR From DataBiotics

By Darrell Ingold

MINIWRITER I is a very usable word processor for the less-than-full system TI owner. It requires only a cassette recorder with proper connector, and the MiniMemory (which I have noticed in some of the recent catalogs is less than half a C-note), and this cassette program. That means that for well under \$100 you can proceed to processing words.

The features on this program are many and a lot of them are the same ones found on TI-Writer. It is possible to insert or delete letter by letter or even line by line. The Move feature and Copy feature work well (and quickly) and are easy to operate. The directional arrow keys (Function S,E,D,X) allow direct movement to any place on the document and the functions 4 & 6 will scroll the screen up or down 24 lines at a time for quickly reaching another section of the document. Function 5 windows from left to right across the page with overlapping windows for continuity of reading. A lot of this sounds rather familiar to you TI-Writers users, right?

MINIWRITER I also has a tab function and a cancel last command (undo) feature. Ready to go back to the very beginning to proof-read your letter? Easily done with the 'home' command key. All through now and ready to start another letter? Function 'erase' empties the buffer for a fresh start. There is also a 'find string' command for searching the document for a name or whatever. The 'save file' and 'load file' will work with either cassette or disk.

If you are already familiar with the use of the MiniMemory you will slide easily into this program but if you are like me and have not used a cassette or MiniMemory for a long time you will have to read the instructions. Pages 5-7 of the 28 page instruction booklet lead you (and me) by the hand on loading and starting the program. Nothing to it (if you follow directions!).

One of the interesting things I noticed different about this program from my TI-Writer was that in TI-Writer as my cursor reached the right hand edge of the screen, the screen (and therefore the text) would jump to my left with the cursor now appearing at about column 12 much as the windowing (function 5) works on this program. This characteristic is very unnerving to many people. With MINIWRITER I however, as the cursor reaches the righthand edge of the screen it moves one letter at a

time to the left, making it much easier to keep your eyes focused on it. There is a word wrap feature with it also but it works differently than what I am accustomed to. When the extreme right margin is reached, the cursor goes directly to the beginning of the next line with the next letter, even if it is in the middle of a 3 letter word. To prevent this fragmentation, there is an audible beep 6 spaces before the end of the line, much like a typewriter bell, to warn the user to hit the enter key after the next word or hyphenate. If you use a monitor with no sound, this will be a real problem because it will get very frustrating constantly going back to correct these fractured words. As long as you have the audible beep it is as natural as using a typewriter.

On page 26 there is a list of printer commands (control key commands) that can be placed right into the text to control the printing on the printer such as carriage return, line feed etc. These characters appear on the screen but do not appear in the printed document. It is necessary to check these commands out with your printer manual to make sure you get what you want in the final product on the printer.

MINIWRITER I may not have all the features of some of the other disk based word processors but it has most of them at a fraction of the cost.

It is important to realize that in order to get a printed copy of your letter you must have a compatible printer which also requires some kind of interface (RS232). These are available as stand alones (without the Expansion Box), or as cards to fit inside the Expansion Box. These interfaces are available through most TI retailers and catalog outlets as are various printers. Printer selection is a rather complex matter and should be discussed with your friends at your local users group or someone else who has had the experience ALREADY. The Miniwriter I retails for \$19.95 .

BITMAC

By Edgar Bohann

At the March and April JUG meetings I demonstrated a pre-release version of BITMAC, a new program from DataBioTics. I am happy to say that it has been finalized and is now in production. The list price is \$39.95 but I understand that both of our group's dealers are discounting it about 20% which makes it highly affordable.

It is difficult to describe the power of BITMAC to anyone who hasn't seen it or its ancestor (MACDRAW on the Apple Macintosh) or one of its cousins (on a brand X machine). When I first saw MACDRAW, I liked everything about it except the silly mouse (I set traps for them at home) and the lack of color. BITMAC has fixed all that and best of all it runs on the TI-99/4A which I like much better than any Apple!

BITMAC's main menu screen features a verticle row of ICONs which are symbols that indicate a menu selection. You move an arrow cursor to the desired ICON with your joystick and press the fire key to make your selection. The program is written entirely in TMS9900 assembly language so it is very fast. The ICON concept also makes it very easy to use.

The purpose of BITMAC is to enable you to make high quality graphics pictures as easily as possible. You can save the drawings on disk or print them on a dot matrix printer. If you save them on disk, you can recall them later for a Slide Show presentation with BITMAC, edit them further at a later date with BITMAC, or include them in other programs that you write.

The main ICON choices (which put you into one of BITMAC's submodes) are: Slide Show, Draw, Enhance, Copy, Lines, Circle, Rectangle, Vault, Scrub Brush, Text, and Color. Draw Mode allows you to make freehand sketches. Enhance Mode has several choices for computer enhancement to your graphics similar to the way NASA retouches satellite photos. Copy Mode allows you to mark a portion of your graphics for duplication at another spot on the screen. Lines, Circle, and Rectangle Mode let you mark a spot for one of these type of elements then let the computer draw it. You can even "drag" an elastic element until it is placed just where you want it then release the fire button to place it. Vault mode lets you save your drawing to disk or a printer. Scrub Brush Mode lets you erase all or part of your graphics. Text Mode lets you add text characters to your graphics. Color Mode lets you add color to the picture for the final touch.

The best part of BITMAC is its ease of use and speed. It is the best graphics drawing program I have seen yet for our computer. It makes heavy use of the joystick (or a trackball). Text mode of course is mostly keyboard oriented but the other modes require very little keyboard operation. Draw Mode is the lone exception. In this mode a number of special options can be selected from the keyboard. You can choose from a choice of 9 brush sizes, 5 cursor speeds, 2 cursor shapes, 16 cursor colors, single or multiple cursors (multiple cursors are handy for placing circles and rectangles), and several other options.

The manual which comes with BITMAC is excellent. It is printed in red ink on light gray parchment paper which provides high contrast and makes it easy to read even in dim light. The manual starts with a general operation description and an explanation of joystick operation. This is enough to get you started "playing" with the program which is great because I dislike manuals that I have to read cover to cover just to get started. The BITMAC manual then has a section devoted to each ICON on the main menu which makes the manual an excellent reference document. For a finishing touch, some of the more complex modes (such as Vault and Color) include operation hints to help you get the most out of the program.

When I reviewed Advanced Diagnostics, I rated it among the top 3 TI-99/4A programs ever written. The two previous top-rated programs were Extended Basic and TI-Writer. Now there is a fourth. I would be hard pressed to rank these 4 programs 1 to 4 because each is of a different type and has a different purpose. However, I believe that Extended Basic, TI-Writer, BITMAC, and Advanced Diagnostics are the 4 most extensive, complete, useful, and innovative programs for our computer. What makes these programs even more remarkable is that I do not believe you can find a Basic interpreter, word processor, graphics generation, and diagnostic/disk fixer program for any other home computer that are comparable to these.

As a side note, if I were to throw in a game program for a "Top 5" TI-99/4A program set I think it would be TI-RUNNER. For now though, BUY BITMAC! It is a truly unique and amazing program.

***** END OF ISSUE *****

THE 99'ERS ASSOCIATION
ATTN: LUCI VEITH
3535 SO. H ST., #93
BAKERSFIELD, CALIF. 93304

PLEASE MAIL CK./M.O. FOR SUBSCRIPTION CHOICE SELECTED ABOVE TO:

SUBSCRIPTION TYPE	AMOUNT	CHOICE
THIRD CLASS - BULK RATE	\$12.00	-----
FIRST CLASS - US AND CANADA	\$17.00	-----
FIRST CLASS - OVERSEAS	\$22.00	-----

NAME: _____
 ADDRESS: _____
 CITY: _____ STATE: _____ ZIP: _____
 DATE: _____

SUBSCRIPTION FORM
FOR "THE NATIONAL NINETY-NINER"
PUBLISHED BY THE 99'ERS ASSOCIATION

THE 99'ERS ASSOCIATION
3535 SO. H ST. #93
BAKERSFIELD, CA. 93304



first class

MR. BILL PECHNIK
1467 CARMICHAEL DR.
PENTICTON BC CANADA V2A 4R9