

ROCKY MOUNTAIN ~~97ers~~
TIC TALK

VOL III, NO 5

DENVER, COLORADO USA

JAN 1985

Non-member Subscription Rate - \$7.50 Annually

Single Copy Price - 75 cents

FROM THE EDITOR

Well, another month has gone by, and I'm running late with the newsletter again. I have been trying to write a program to print a calendar similar to the one we used in the newsletter last year. So far, I'm not having too much luck getting it to work. Hopefully, by next month, I'll have it going and in use on the front page.

I want to remind everyone that the meeting in February will be on the first (1st) Tuesday instead of the usual second (2nd). I will note this in the February issue of the newsletter, but I figure it can't hurt to say it here too.

I want to wish everyone the happiest of holiday seasons, and hope to see every member at the January meeting.

BEST WISHES!

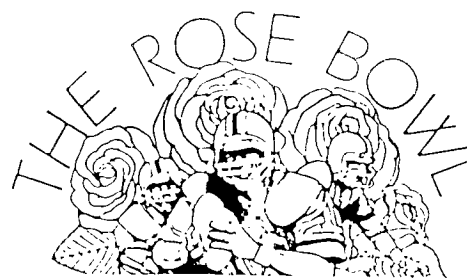
JANUARY MEETING

JANUARY 8

Jefferson County Fairgrounds

Auditorium 7:00 PM

6th Ave. West to Indiana Ave.



JANUARY						
S	M	T	W	T	F	S
			2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

New Programs Reviewed

by Mike Holmes

One of the best services that we as a club can supply to our members is to try to filter out programs which may sound like what you want but really do not perform as advertised. With this in mind the group is beginning to invest in some software for evaluation purposes. If we find something unsuitable or very useable we will attempt to bring it to your attention here.

This month I am going to review two new pieces of software which were ordered for this purpose.

1. 9900BASIC - Which is advertised as an assembly language development tool. and
2. SUPER COPY - Advertised as a general purpose copy program.

9900BASIC is a series of subroutines written in assembly language which are intended to simulate the syntax of extended basic and make it easier for an experienced extended basic programmer to develop programs in assembly language. The routines included seem well thought out and of good utility. The user should know something about assembly language on the TI since all the functions of Extended Basic are not included in the package. The code is included as source code which is included in your assembly program by using the assembler's copy directive. All in all this is an excellent package and seems to be well documented and thought out. The single problem with the documentation is the last page. This page contains a list of names which are not explained. Upon looking into the source code I found that these are the names of labels which are reserved for the use of 9900BASIC only. If you use any of these names in your assembly program you will

find that there are duplicate definitions.

SUPER COPY is also a product of CHALLENGER SOFTWARE, who marketed 9900BASIC. After looking at the previous program I was prepared to use a well thought out responsive copying routine. I was very disappointed. Aside from being able to copy protected files of some types I found this package to be a waste of money. The program offers three options on it's menu. 1. Catalog a disk. 2. Copy a file. And 3. Exit the program. There is no provision for using the expansion memory or for backing up a disk in it's entirety.

This program turned out to be an extended basic disk catalog program which does get around TI file protection but there are better programs on the market for this purpose. They also work faster.

```
*****
*                               *
* W A N T   A D S               *
*                               *
*****
```

FOR SALE

1 TI 99/4A CONSOLE (NEW)	\$65
1 P.E. BOX (NEW)	
-32K MEMORY CARD	
-DISK CONTROLLER	
-DISK DRIVE	\$375
1 RS-232 CARD	\$80
1 TI-WRITER	\$65

OSCAR SMITH 794-8448 (HOME)
977-9524 (WORK)

MOONLIGHTER

ARTICLE

NOW THAT YOU'VE BEEN USING YOUR COMPUTER FOR AWHILE ISN'T IT ABOUT TIME THAT YOU SET IT UP PROPERLY. SITTING AND STARING AT A SCREEN FOR A FEW HOURS CAN BECOME A LITERAL PAIN IN THE NECK. YOU SHOULD SET UP YOUR COMPUTER IN AN ENVIRONMENT THAT MAKES COMPUTING PRODUCTIVE AND LESS OF A STRAIN ON YOUR BODY. HERE ARE SOME STEPS TO HELP YOU COMPUTE PAIN-FREE:

+ YOUR SCREEN SHOULD BE ARM'S LENGTH FROM YOUR FACE.

+ IF YOU SIT AND STARE STRAIGHT AHEAD, YOU SHOULD BE LOOKING AT THE VERY TOP OF THE COMPUTER SCREEN: IN OTHER WORDS, WHEN YOU LOOK AT THE CENTER OF THE SCREEN, YOU SHOULD BE LOOKING DOWN SLIGHTLY.

+ HERE'S HOW TO POSITION YOUR KEYBOARD: LET YOUR ARMS HANG LOOSE AND BEND YOUR FOREARMS UP TO FORM A RIGHT ANGLE AT THE ELBOWS-- THAT'S A GOOD HEIGHT FOR THE KEYBOARD.

+ YOUR CHAIR HEIGHT SHOULD ALLOW YOUR FEET TO JUST TOUCH THE FLOOR.

+ YOU SHOULD HAVE SOME LOWER-BACK SUPPORT. IF YOU DON'T HAVE A GOOD BACK-REST, PLACE A PILLOW BEHIND YOUR BACK AT THE BASE OF YOUR SPINE.

+ THE TOP OF YOUR PRINTER SHOULD BE WAIST HIGH AS YOU SIT, AND CLOSE ENOUGH THAT YOU CAN REACH IT EASILY WHILE SITTING AT YOUR TERMINAL.

+ KEEP A SMALL FOOT-STOOL OFF FRAME BOOK UNDER THE DESK. PROP ONE OR BOTH FEET ON THE STOOL TO HELP RELIEVE PRESSURE ON YOUR LOWER BACK.

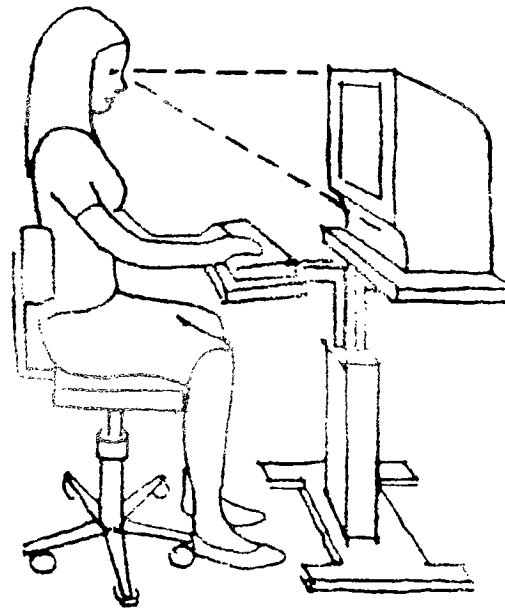
+ TO GET ON BOTH FEET AT THE SAME TIME IN ORDER TO DISTRESS YOUR LOWER BACK...

+ CHANGE YOUR SITTING POSITION WHENEVER POSSIBLE. SQUIRMING AROUND CAN PREVENT MUSCLE SPASMS AND HELP KEEP YOUR BLOOD CIRCULATING.

+ FACE YOUR WORK DIRECTLY, RATHER THAN TWISTING YOUR BODY.

+ AVOID SITTING ON A WALLET OR OTHER BULKY OBJECT. IT CAN PRESS ON YOUR SCIATIC NERVE (WHICH RUNS DOWN THE LENGTH OF YOUR LEGS), RESULTING IN SHOOTING PAINS.

+ WHEN YOU START TO FEEL STIFF OR TENSE, GET UP AND WALK AROUND FOR FIVE TO TEN MINUTES. IT WILL DO A WORLD OF GOOD.



ADEQUATE SUPPORT FOR MIDDLE AND LOWER BACK.

SPLIT-LEVEL TABLE GIVES MAXIMUM FLEXIBILITY IN SCREEN AND KEYBOARD HEIGHT AND ANGLE.

BOTTOM OF CHAIR SHOULD CURVE DOWN SLIGHTLY.

USER SHOULD LOOK DOWN SLIGHTLY TO SEE SCREEN.

CHAIR HEIGHT SHOULD BE ADJUSTABLE.

FORTH DISK FIXER

by Joel Gerdeen
from MSP 99 NEWSLETTER

The October ASSEMBLER SUBGROUP meeting covered a review of the TI99 disk format and the use of the language FORTH as a disk fixer. This article continues that discussion and also reviews the operation of the NAVARONE INDUSTRIES' DISK FIXER command module.

The DISK FIXER module allows you to access floppy disks by sector rather than by file. You can display, print or change any byte (character) or series of bytes anywhere on the diskette. This allows you to UNDELETE files that have been inadvertently deleted, CLOSE files that were improperly closed and recover data that is otherwise inaccessible. Note, details of these operations will not be covered in this article. Only a review of the capabilities of DISK FIXER and how FORTH can be used to accomplish the same is covered. In summary, anyone who is inclined to fix disks should learn FORTH and can come to the ASSEMBLER SUBGROUP for further assistance.

First, a definition of some terms. The TI99, single-sided, single-density disk is divided into 360 sectors, a minimum of two of which are always used for diskette directory information. DISK FIXER works with sectors while FORTH works with SCREENS or BLOCKS. Each block is made up of four sectors, so there are only 90 BLOCKS on a 385D diskette.

FORTH contains two different full-screen editors that edit a complete screen or block all at once. Each screen is made up of 16 lines of 64 characters each for a total of 1024 characters or bytes. One of the editors displays only 40 characters wide while the other displays the full 64 characters. FORTH allows you to change any

character on the screen as long as it is displayable. Remember that only 95 of the 256 ASCII codes are displayable. The non-displayable codes must be changed in a different manner that will be described later. Note that DISK FIXER does not allow any characters to be edited in this full-screen editor fashion.

To load disk data into memory, the DISK FIXER has a command called READ SECTOR with the syntax "R sss,d", where sss is the sector number and d is the disk drive device code. Both sss and d can be omitted to allow sequential reading of the same diskette. Only one sector is in memory at any time.

FORTH loads a block from diskette in two ways. Typing "b BLOCK" will load block numbered b if it is not already in memory. Typing "b EDIT" will both load the block and place you in the editor mode for changes. As many as five blocks of data can be in memory at once.

To store a sector of data back to disk, DISK FIXER uses the WRITE SECTOR command with the syntax "W sss,d" where variables are the same as before. Care must be taken because the sector number sss is incremented by other commands and other disk sectors could be overwritten.

FORTH will store any block in memory to disk with the simple command "FLUSH". Note that all changed blocks in memory will be flushed. You can erase all blocks in memory with the "EMPTY-BUFFERS" command.

To display a sector, DISK FIXER uses the DISPLAY BUFFER command with the syntax "D". The current sector in memory will be displayed. The display will show both the hexadecimal and ASCII forms of the data along with a memory location required for further changes.

FORTH will display any block in memory or on disk with the command

"b BLOCK n DUMP" where again b is the block number and n is the number of bytes to be dumped. The display is similar to that of DISK FIXER.

To change data in the sector buffer, DISK FIXER uses the ALTER DATA command with the syntax "A oooo" where oooo is the address as displayed by the D command. New hexadecimal data is typed in. A command INSPECT/CHANGE with the syntax "M oooo" can be used to change any memory location including VDP memory. This command has the same syntax as the M command in the TI-DEBUGGER.

FORTH uses one of two store commands to alter memory. For complete 16-bit words of data the syntax "n addr. !" will store the data word n at address addr. For bytes of data, the syntax "n addr C!" will store a byte n at address addr. In addition, the FORTH command EXPECT can be used to read characters directly from the keyboard to memory. Of course for text, it is easier to use one of the editors described above.

The last command that DISK FIXER has is the PRINT SECTOR command with the syntax "sss,d,n" where n is the number of sectors to print. The format of the output is similar to that of the display command except that it is twice as wide as the screen output. Note that multiple sectors are read from the diskette and printed automatically to any TI99 supported device.

FORTH uses a variation of the DUMP command to accomplish the above, making use of the SWCH command to switch display output to the printer. The syntax is "SWCH b BLOCK n DUMP UNSWCH". Note that if n is 1024 the complete block or four sectors will be printed. If greater than 1024 you will start printing other memory locations outside your block. Note also that the FORTH TRIADS command will print any number of blocks directly from diskette but

only for blocks that are ASCII displayable.

DISK FIXER also has a FIND STRING command. While FORTH does not, a FIND command procedure to do such is easily written.

In summary, I am an advocate of FORTH which has extended my interests further into the power of the TI99/4A. While the DISK FIXER module is all it is advertised to be, I put my money into FORTH and its more powerful features.



TI-WRITER TIF

(Adapted from an article
by Norman Rakke in the
Pittsburgh users Group's
P.U.G. Peripheral)

Reprinted from MSP 99 NEWSLETTER

Have you ever wished that your printer has some special character or symbol, such as Greek letters like pi or alpha, or a division symbol, or Cyrillic characters, or something of that sort? Well, if your printer has dot graphics capability, and allows you to mix text and graphics on the same line, then you can use TI-Writer to create characters which you define, and print them in documents you create. Before getting into details, you should note that the information in this article specifically relates to the TI99/4 printer; however, the control codes used are standard Epson codes, and the technique should work with most Epson compatible printers. An attempt was made to verify this technique on the Star Micronics Gemini 10, but the attempt failed because the Gemini 10 will apparently not allow graphics and text mixed on a line. If you have some other printer which has the above mentioned capabilities, you can still do what is described below; however, you should read your printer manual to determine how your printer does each of the things mentioned. If anything is done in a different manner, you will have to take that into account.

First of all, you must make sure that your printer is ready to receive graphics data. You may have to remove the cover from your printer and change the position of a DIP switch so that the printer can receive 8 data bits. Check your printer manual to make sure that the graphics mode works properly. Finally, if your printer is connected to the serial port of the 80322 interface, you will need to include `.Dfns` in your file description; every time you use the printer,

Printer graphics consist of one or more columns of dots. There are a total of 480 such columns across a page. Each column is 8 positions high, and a dot can appear in any one of the 8 positions. Each position has a data value associated with it as shown in the figure below. The data representing a particular column is simply the sum of the data values for all positions where a dot is to be printed.

```

+---+
|128|
+---+
| 64|
+---+
| 32|
+---+
| 16|
+---+
|  8|
+---+
|  4|
+---+
|  2|
+---+
|  1|
+---+

```

For example, to print a column where only the top dot is printed would require a data value of 128. A column in which both the top and the bottom dots were printed would require a data value of $128+1$, or 129. A column which had all 8 dots printed would require a data value of $128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$, or 255.

Now let's create our own graphics character and see how we can incorporate it in a document prepared with TI-Writer. Let's make an arrow pointing up as our special character. The normal characters built into the printer are as wide as 6 columns of graphics (480 columns / 80 characters), so let's make our character the same size. It's helpful to draw the character on graph paper, so let's do that.

```

+---+---+---+---+---+---+
|   |   |   |   |   |   |
+---+---+---+---+---+---+
|   |   | * |   |   |   |
+---+---+---+---+---+---+
|   | * | * | * |   |   |
+---+---+---+---+---+---+
| * |   | * |   | * |   |
+---+---+---+---+---+---+
|   |   | * |   |   |   |
+---+---+---+---+---+---+
|   |   | * |   |   |   |
+---+---+---+---+---+---+
|   |   |   |   |   |   |
+---+---+---+---+---+---+

```

The data for the columns from left to right is 16,32,126 (64+ 32 + 16 + 8 + 4 + 2), 32, and 16. (You might find it useful to know that the normal text characters of the printer do not use the column on the right (to prevent characters from running into each other) or the bottom row (except in lowercase characters with descenders). To send the graphics data to the printer we first need to send a control code. For normal graphics mode this is in the form of ASCII codes 27,75,n1,n2. The codes n1 and n2 define the number of bytes of graphic data which will be transmitted. This represents the number of columns of dots which will be printed. If N is the number of columns of graphics to be printed, then n2 is the integer result of $N/256$, and n1 is $N \text{ MOD } 256$, or the remainder of $N/256$. For our special character, n2 is 0 and n1 is 6. The graphics data follows immediately after n2. The complete string of ASCII values needed to print our special character is therefore 27, 75, 6, 0, 16, 32, 126, 32, 16, 0. We will create the special character by using the TI-Writer "transliterate" command. We will use some character we won't be using, such as ~, and transliterate it to the string of data we defined above.

We can now use our special character in a document. If we were writing directions for using a program where pressing the E key moved something up on the screen, we could write the following:

To move up ~ Press E

If we now print this short file using TI-Writer's Text Formatter, we will get the following:

To move up ↑ Press E

Using special characters which are six graphics columns wide allows you to still be able to use the .AD and .CE even if special characters are present in the text. If you are not going to use either of these commands, you can make your characters of different widths than 6.

CAUTION! Using the method just described, you can design almost any character that you might desire. Unfortunately, the values 8, 12 and 13 cause problems which disallow their use for graphics data using this method. You may have to modify your character to avoid these values.



ADDING A NUMERIC KEYPAD TO THE TI99/4A.

by Norman Riger
from the
MSP 99 NEWSLETTER

Proficiency with a numeric keypad is a valuable skill. Employment ads in the newspaper often mention requirements of ten thousand keystrokes per hour for keypad operators. Considerable time and practice is required to achieve such speed and the necessary accuracy. Your home computer can provide you with the opportunity to learn this valuable skill. This article shows how to connect a numeric keypad to the TI99/4A. This method will not work with the TI99/4, which is wired differently and has a different keyboard.

My procedure involves selecting a surplus keypad that has ten separate spring type switches, each with two terminals. Both terminals of each switch must be accessible and not permanently connected to the terminals of a different switch. There are four rows of switches on the standard keypad. The top row (left to right) has keys for seven, eight and nine. Below are keys for four, five and six. The third row has keys for one, two and three. The bottom row has a single key (at the left) for zero. There is no ground connection on the keyboard of the TI99/4A and there should be no ground on the keypad selected for this project. Only these ten keys are required and any others are not needed and may be ignored as long as they aren't connected to the required keys.

Turn the computer upside down and remove the seven recessed Phillips screws used to fasten the bottom. Carefully pull out the on/off switch until it comes off and then remove the bottom. A fifteen conductor ribbon cable connected the keyboard to the processor printed circuit board (covered by a metal shield). Only seven wires are needed to connect the keypad and there are several methods possible.

A clamp type connector can be attached to the ribbon cable (use an ohm-meter to make sure that adjacent conductors don't become shorted together). It is possible to disconnect the ribbon cable connector from the processor printed circuit board and place an additional connector between them which contains the seven wires required. These methods have the advantage of requiring no solder connections on either the keyboard or the processor printed circuit board. Another possible method is to scrape some of the insulation off of the required conductors in the ribbon cable and solder directly to them.

My method is to solder the seven wires directly to the printed circuit board in order to save the cost and trouble of finding additional connectors or run the risk of damaging the ribbon cable. The end of the ribbon cable closest to the joystick connector is pin one and the end closest to the I/O port connector is pin fifteen. My method involves soldering each wire and running it through one of the ventilation slots in the bottom of the computer. Bread ties may be used to provide strain relief for the wires.

Pin two of the keyboard connector should be connected to one terminal of the switches on the keypad for the 6, 7, 8, 9, and 0 keys. Pin seven should be connected to one terminal of the switches for the 1, 2, 3, 4 and 5 keys. At this point, each of the ten switches on the keypad should have a

connection to one terminal with the second terminal still unconnected.

Pin eight on the keyboard connector should be connected to the one and zero keys on the keypad. Pin nine is connected to the five and six keys. Pin 13 is wired to the 2 and 9 keys. Pin 14 goes to the 3 and 8 keys. The last connection is from pin 15 to the 4 and 7 keys. Check the keypad to be sure that all twenty terminals are connected correctly.

Be careful to avoid cold solder joints and solder bridges between adjacent connectors. A grounded (three wire cord) soldering iron is recommended for the protection of the sensitive computer chips. Replace the bottom of the computer along with the screws and the on/off switch. If it's necessary to remind you that the computer should be turned off and unplugged during the modification procedure, it's recommended that you have the job done by a qualified electronic technician. The modification described in this article has been performed successfully and requires no additional hardware or software of any type.

```

70 REM *****
75 REM *   LOAD   *
80 REM * A PROGRAM TO *
85 REM * CATALOGUE DISKS*
90 REM * BY FRANK UNLIK *
95 REM *****
100 CALL CLEAR
110 DIM R$(60)
120 DIM X$(254)
130 DIM TYPE$(5)
140 TYPE$(1)="DIS/FIX"
150 TYPE$(2)="DIS/VAR"
160 TYPE$(3)="INT/FIX"
170 TYPE$(4)="INT/VAR"
180 TYPE$(5)="PROGRAM"
190 A=1
200 OPEN #1:"DSK"&STR$(A)&".
",INPUT,RELATIVE,INTERNAL
210 INPUT #1:A$,J,J,K
220 DISPLAY AT(1,1):"DSK";ST
R$(A);" - DISKNAM=" ;A$;"AV
AVAILABLE=";K;"USED=";J-K
230 DISPLAY AT(22,1):"*****
*****"
240 DISPLAY AT(4,1):" No FILE
NAME Sz TYPE P": --
-----";
250 Q=1
260 FOR I=001 TO 127
270 CALL KEY(3,2,3)
280 IF S=1 THEN 290 ELSE 330
290 FOR A=1 TO 100
300 CALL KEY(3,2,3)
310 CALL KEY(3,2,3)
320 IF S=0 THEN 310 ELSE 330
330 (CALL)
340 IF Q=21 THEN 350 ELSE 36

```

```

0
350 Q=6
360 INPUT #1:A$,A,J,K
370 R$(N+1)=A$
380 IF LEN(A$)=0 THEN 340
390 N=N+1
400 IF N=16 THEN 420
410 IF N=32 THEN 420 ELSE 470
420 DISPLAY AT(23,1):"TO SEL
ECT FROM ABOVE PRESS S10 CON
TINUE PRESS SPACE BAR "
430 CALL KEY(3,K,S)
440 IF S=0 THEN 430
450 IF K=83 THEN 550
460 DISPLAY AT(23,1):"
"
470 IF N<15 THEN 480 ELSE 470
480 DISPLAY AT(Q,1):"
"
490 DISPLAY AT(Q,1):N;TAB(15)
;A$;TAB(15);J;TAB(19);TYPE$(
ABS(A));
500 IF ABS(A)=9 THEN 510
510 IF A=0 THEN 530
520 DISPLAY AT(Q,28):"Y";
530 NEXT LOOP
540 CLOSE #1
550 DISPLAY AT(23,1):"
"
560 DISPLAY AT(24,1):"SELECT
No:"
570 ACCEPT AT(24,11):N
580 RR$="RUN "

```

```

590 RS$="DSK1."
600 RZ$=RR$&SEG$(1,1)ARS
&R$(M)&SEG$(1,1)
610 CALL CLEAR
620 DISPLAY AT(5,1):"
TYPE IN-"
630 DISPLAY AT(11,1):"
640 DISPLAY AT(12,1):"
"
650 DISPLAY AT(15,1):"
660 DISPLAY AT(13,5):RZ$;TAB
(27);""
670 DISPLAY AT(20,1):"
AND PRESS ENTER"
680 END

```

CATALOGUING DISKS
from BUG-BYTES

If you're like me, you want to run a program, and you can't remember its file name, so you have to get out your disk manager module to catalogue the disk. With this programme you won't need to do that any more. Put the programme on to each of your disks, and if you use the filename LOAD for it, it will load automatically as you select Extended Basic, then you just have to enter the filename.

FILE FILE TRANSFER WITHOUT THE TEL.

by Larry Frikockis

Reprinted from CLEVELAND AFEA 99/4A

When transferring files over the phone, the easiest and most reliable method is to use the Terminal Emulator. The price for this ease of use and error checking is that the process is very slow. Fortunately, there is a way to transfer Basic and Extended Basic programs at five times the speed of the TE.

First, both the sender and receiver must select TI Basic. Then, load the program to be sent into memory and type "SAVE RS 232", but do not press enter yet. The receiver should type "OLD RS232". When you are ready, switch over to the modems and press enter. The receiver should wait about five seconds and then press enter also. Both users should see a number, corresponding to the number of blocks of the program, at the top of the screen. This number will slowly count down to zero at which point the transfer is complete. Since the program is loaded directly into memory, the receiver must remember to save the program to disk or cassette.

Obviously, this method is limited to programs which can run in Basic or Extended Basic. The procedure for transferring assembly programs or text files is a little more complicated, but still faster than the TE.

To use this method, both users must have an editor assembler module. First, select the editor assembler and press "1" for the edit section. The receiver should do the same. Then select option 4 for print. When you are asked for a filename, enter "RS232.BA=300.CRLF". The receiver selects option 1 for load (make sure Disk A is in the drive) and uses "RS232.BA=300" as a filename. When both sides are ready, switch over to the modems and press enter. Again, the receiver should wait five seconds before hitting enter. With this method, there is no way of telling how much more of the file is left to be sent, so just sit back and wait until it is finished. The file can be saved to disk by using option 3.

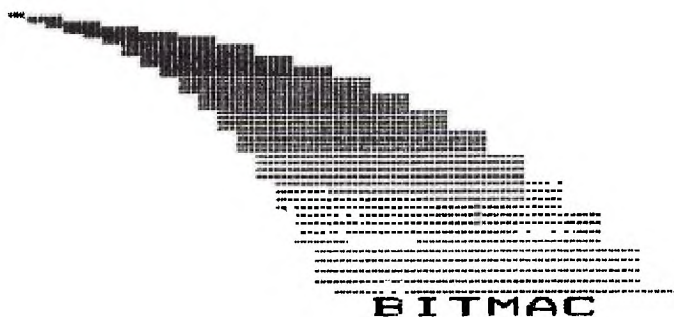
<<<< DISPLAY ADS >>>>

10 in X 7.5 in - \$15.00	ALL DISPLAY ADDS must be camera ready
RATES: 5.5 in X 7.5 in - \$8.00	and must be received before the 15th
3 in X 7.5 in - \$4.50	of the month and accompanied by a
	check made out to the ROCKY MOUNTAIN 99ers P.O. Box

3400, Littleton, CO 80161. Since the Club is a non-profit organization all money collected for advertizing goes toward the publishing costs of this newsletter.

<<<< WANT AD RATES >>>>

MEMBERS - FREE (25 word max) We must have your add by the 15th of the month to assure insertion in the next issue. Call 458-7315 or mail to BOX 3400 Littleton, CO 80161. NON-MEMBERS must use DISPLAY ADS!



Finally! Bitmapped graphics that BOTH you and your TI can understand!

The power of this program has never been offered to the TI user from any source, including Texas Instruments. BITMAC offers the ability to create the finely detailed and elaborate graphics that, until now, could only be done "on other computers".

BITMAC offers 16 colors, point to point lines, rectangles, circles, copy area, shrink area, enlarge area, "life" graphics enhancement, extensive cursor control options, invert bits, extensive erase and redo functions, text on graphics, text on text, multiple color text characters, ICON command entry, single key entry, "real time" processing and much more!

If you want to SIGN your name on the screen, spend hours "doodling", have a computer blackboard or create computer art, this is THE program. BITMAC was designed with novice computer users in mind. Even if you are a complete beginner, you should find it easy to make graphics that "old hands" will envy.

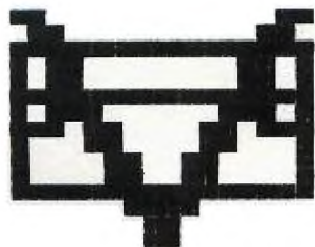
BITMAC comes with extensive documentation and warranty. There are even provisions to update your program whenever a new version is to be released! There's a three year warranty, and even "insurance" against damage to your disk! ("oops who spilled the coffee")

The price? \$35.00 (\$2.00 backup disk)

TO ORDER:

Vaughn Software is a mail order firm. In metro Denver phone orders are welcome. FREE delivery available in most of metro Denver. All other orders should be submitted by mail. E-MAIL orders for DAVE VAUGHN at the STAR BOARD 455-3113.

Include \$5.00 postage and handling for all orders by mail. Please write for advance copy of the warranty, please include a self addressed and stamped envelope. Dealer inquiries not invited.



VAUGHN SOFTWARE
FOR THE TI-99/4A

Rocky Mountain 99ers

TIC TALK

This publication is printed monthly for the benefit of the membership of the Rocky Mountain 99ers Computer Club. The Club and the paper are not for the benefit nor backed by any commercial enterprise. Both are non-profit in nature and are for the sole purpose of computer education. Any fees collected are used to defray any cost to maintain the organization. Neither the paper nor the Club have any affiliation with Texas Instruments. Any statements published in this paper are not necessarily the opinion of the membership.

OFFICERS and CHAIRMEN

PRESIDENT.....TED MICHELSEN.....986-3513
VICE PRESIDENT.....MIKE HOLMES.....751-7945
SECRETARY.....MARTHA WEEG.....320-5589
TREASURER.....KEN MONSON.....233-1788
EDITOR.....DAVID OWEN.....458-7315
LIBRARIAN.....PETE CROWELL.....750-5949
MEMBERSHIP.....MARTHA WEEG.....320-5589
PROGRAM CHAIRMAN.....MIKE HOLMES.....751-7945
EDITOR/ASSEMBLER..SIG.....MIKE HOLMES.....751-7945
TI FORTH.....SIG.....PETE CROWELL.....750-5949
MULTIPLAN.....SIG.....BEN KRAMER.....237-1056
THE STAR BOARD....BBS.....455-3113

* * ROCKY MOUNTAIN 99ers * *
P.O. Box 3400
Littleton, CO 80161

FIRST
CLASS

* Do you see stars on the label *
* this means your membership is *
* now due. Send in your renew- *
* al today so you don't miss a *
* single issue of TIC-TALK!!! *
