



# SOONER 99ERS

This newsletter is the official publication of the SOONER 99ers  
POB 61061  
Oklahoma City, OK 73146

Sooner 99ers BBS:  
(405) 672-8270 300/1200 8N1

Club Officers:  
Mark Mitchell

Dave Lewis  
Garth Potts  
Dick Farrar  
Jerry Robertson  
Barry Peterson

President  
Vice President  
Secretary  
Treasurer  
Librarian  
Newsletter Editor  
(405) 721-5930

February, 1990

Greetings, fellow club members, if you were not aware of recent club activities, look at the top of the page again. We have held elections and have new officers!!!! (Well, not so much new as in different positions) Please join me in congratulating our new Prez, Mark Mitchell.

Due to popular demand/request, I have reprinted an article on Funnelweb configuration. We now have version 4.21 of that GREAT software package in the club library and many of you will need to set it up. The article I have included was originally from Jim Swedlow of the User's Group of Orange County (California) and comes to us via the Tidewater UG newsletter and the QB-Monitor of the Queensborough (N.Y.) User's Group. Thanks for all of those groups and I hope it fills your needs. (It talks about versions 4.10-4.12 but should also apply to the new 4.21 version)

Opinions expressed in this newsletter are those of the editor or authors. Articles, unless otherwise noted, are written by the editor.

I have also included a reprint of TI-Writer Toolbox #8 by Ed Machonis of the QB 99ers. It looks great and contains useful information pertaining to that software package we all use.

I have also included another 'crib sheet' from Garth Potts. This one, for TI-Artist Plus puts everything at your fingertips. (maybe not everything, but pretty darn close) Garth does a great job of simplifying what you need to have for so many software packages.

Also in this issue is an article or commentary on Computer Shopper's decision to drop support for the TI-99 and other Classic Computers. I have dropped my subscription but will also send a note to tell them why. (As if they care)

If you're reading this newsletter at the Feb. 10th meeting, you already know we are going to modify consoles to eliminate the alpha lock/joystick conflict. Otherwise, you missed a useful fix for your machine. Till next month, bye...BP

Advertising related to the TI-99 or Myarc 9640 will be accepted on a 'space-available' basis for individuals or companies with no payment required.

## STILL CLASSIC AFTER ALL THESE YEARS

Ten years ago a major semi-conductor manufacturer had a nifty 16 bit microprocessor chip, "all dressed up with no place to go", and decided that it would be a great idea if they went into computer business. The result, of course, was our TI99.

Texas Instruments' colossal blunders in marketing need not be repeated here. Yet millions of TI99's were sold, both before and after TI succumbed to its self-inflicted wounds in October 1983 and the TI99 has refused to die. These quirky machines still run, users groups are still active, and software is still written for a computer that, according to the "experts", should have died long ago.

We TI owners are true computer hobbyists. We take our computers apart, modify them, adopt bits and pieces from other machines, and write and use software that doesn't seem to be limited by the "inadequate" memory of the TI.

Since the beginning of our turbulent relationship with the TI, we've had an interesting ally in the Computer Shopper. If we needed a chip, or a connector, or a disk drive at a decent price, the odds were we could find it in the Shopper. Computer Shopper repaid us for our loyalty by running columns devoted to the various orphans. We provided a market for power supplies and 300K half height drives, they gave us two or three pages every month, and everybody was happy. It was a nice hobbyist type arrangement and as long as the Shopper was run by hobbyists TI and Times and Adam weren't bad words.

Unfortunately the hobbyists at the Shopper are gone. The new boys at Park Avenue, the ones in the BMW's and high dollar suits, view computers as an appliance to be marketed and sold. Appliances are constantly upgraded and today's appliance makes yesterday's appliance obsolete.

We're told that if you don't own a 486 or 586 or 986 or whatever else is new this week, there's something wrong with you. Your breath smells, you don't have any sex appeal, and you probably can't count to ten without lots of help.

Now that the appliance salesmen have taken over from the hobbyists, we're told that we should "move up" to a better machine. We were informed in patronizing tones that we were obviously living in the past and somehow morally and mentally deficient because we chose to regard our computers as a hobby.

My XT Clone - another "obsolete" machine by the way - is a pretty sophisticated typewriter and adding machine that lets me bring work home. I like my XT pretty much in the same way that I like my food processor or radial arm saw. It's a tool and it does a lot of useful work but my clone isn't my hobby.

There's something to be said for obsolescence. I've got a '55 Ford pickup in my driveway and a muzzle loading rifle hanging over my fireplace and there's a real joy in making something old work well. Playing with an old TI is a challenge. We TI'ers as a group probably know our way around our computers much better than any of the marketing experts in the fancy suits. We make cables, they buy them. We get along with twenty five cent floppies, they gripe because 80 meg hard drives are too small. We write programs, they throw memory and money at their problems.

Next week the appliance salesmen will be selling used cars or aluminum siding and our TI's will still be running. We'll still be looking for cheap disks and soldering up cables and having fun. And fun is something that all the money and all the memory chips in the world won't buy.

Dave Lewis  
Sooner 99ers

## NEW MUSIC FROM HARRISON SOFTWARE

Dolores P. Werths — Harrison Software, 5705 40th Place, Hyattsville, MD 20781

Supplement to the *Banach Computer Society* 71-80/84, November - June 85.

Many people have done music programs on the TI-99/4A Home Computer, and many have done well at that, writing in Extended Basic. The story is about a new way of doing music programs in Assembly Language, and then making that music play as if it were Extended Basic. What we've done at Harrison Software has been a team effort between our resident musician and our resident assembly programmer. Neither could create this music alone.

The effort began a couple of years ago, when our musician (Dolores P. Werths) wanted to hear some music by Johann Christian Bach. That music, which is a set of six piano sonatas, is very rarely played, and was not available on records. She asked our resident assembly programmer (Bruce Harrison) for some help in getting the TI to play the music, so she could listen to it and study the composition. After making some attempts in Extended Basic, which were really not satisfactory, the effort to do it all in assembly was started.

The team effort was absolutely necessary because the assembly programmer cannot read music, and the musician cannot deal with all the intricacies of assembly, including the unfriendly hexadecimal number system. The programmer then made up a file of nothing but equates called the *notes* file. This file allows the musician to create music in her own quasi-musical notation, while the assembler takes care of converting the musician's input data into that mysterious hexadecimal stuff. The notes file allows the musician to enter such statements as:

```
DATA D32
BYTE V52
```

The musician means by this that a third-octave D is to be played by generator 2, and that volume level 5 is assigned to that generator. This concept, where all the available notes and volumes are entered as mnemonics, is still in use in all our music efforts, but we had another hurdle to overcome. That hurdle is the matter of timing the durations of the notes when playing. That was important because in the Baroque and Galante music, there are numerous musical ornaments, and these require very fast and precise timing to sound right.

In the TI Editor/Assembler manual, there is a whole chapter devoted to sound, and the method recommended by TI was to create "sound lists" in main memory and to put these bytes in VDP RAM, then let the VDP produce the sound through the

generators. Unfortunately, the timing of sound durations by the VDP is done using its vertical interval as the clock. This means that the timing is all done in 60ths of a second, and that proved to be a killing limitation for the ornaments in the music. In other words, one cannot properly execute 64th note triplets when the smallest duration for either a note or a rest is 1/60th of a second.

Fortunately, there was another method of doing things in Assembly, which was mentioned in passing by TI in the E/A manual. That is the method of passing sound bytes directly to the sound generator chip at address > 8400. TI of course pointed out that when you do this, you must use your own timing loop to control the duration of the sound. This turned out to be the main ingredient of our success. We soon found out that our timing loop, even with purposeful time-wasting operations built in, would have to execute over 200 times for a 64th note at a *Presto* pace. This gave us the flexibility to execute all the ornaments flawlessly.

To make all that accessible to the musician, we created the concept of the *TEMPO* file. This too, like *NOTES*, was all equates, and required only one numeric entry to define the duration of the 64th note, then let the Assembler derive the durations for 32nd, 16th, 8th, and so on, from that initial entry. Even the durations for triplet notes are entered as simple math expressions, which the Assembler computes for us. Again, as with the notes, the durations were translated into easy mnemonics, so that the musician enters *DATA Q* for a quarter note, *DATA E* for an eighth, and so on.

That file of duration equates, which we call the *TEMPO* or simply *T* file, looks like this:

```
TEMP EQU 250 (RANGE 4 THRU 2000)
SX EQU TEMP (64TH NOTE)
T EQU TEMP*2 (32ND NOTE)
S EQU TEMP*4 (16TH NOTE)
E EQU TEMP*8 (8TH NOTE)
Q EQU TEMP*16 (QUARTER NOTE)
H EQU TEMP*32 (HALF NOTE)
TS EQU E/3 (TRIPLET 16TH)
TT EQU S/3 (TRIPLET 32ND)
```

Thus all the durations are derived from a single programmer entry. Notice that the range of possible values for *TEMP* is very broad, but values below 200 make the music very fast indeed, while values above 400 make for truly "dirge" playing.

Our resident musician is very careful about the markings given on the sheet music, and sometimes takes many attempts at the tempo number before she's satisfied that the music is being played *Allgro* *Assai* and not *Presto*, or vice versa.

This musical integrity is one of the things that we insist on, and it's yet another reason that our music is a team effort. Many of the music programs we've seen by others have all kinds of superb graphics and special effects of one kind or another, but they are lacking the musician's touch, in that the rhythms are inaccurate, or the various ornaments are improperly executed. For us, that detracts from the enjoyment of the music. In all fairness, many of these are actually programmed in Basic or Extended Basic, so not all of the musical faults can be laid at the programmers' feet. We know from our own experience that it's sometimes impossible to get *CALL SOUND* to do what you really want.

Of course there's more to the musician's touch than just timing. There's also the matter of dynamics and musical balance to consider. The dynamics are of course not anywhere near the available range of real musical instruments. The total range available on the TI sound generator is only 30 decibels, and because of the way we use volume settings we are limited to only 28 decibels. Even so, we can produce dramatic changes in volume. Our resident musician is very careful in the use of large swings in volume. Rarely do the scores call for maximum *Forze* or minimum *Piano*.

For that matter, some of the sheet music we use gives no indication of dynamics at all. In some of these cases we can listen to recordings of the music played by first-rate musicians, and try to emulate their style. In the *Nutcracker Suite*, for example, many cues were taken from a recording by the Philadelphia Orchestra under Eugene Ormandy. Of course there are many recordings of that work which could be used for guidance. In the cases of such works as Johann Christian Bach's Sonatas, Opus V and XVII, there are no recordings readily available, so the musician must study other works by that composer very carefully to get the sense of how he would wish to have his music played.

Having a musician "on staff" here has made all the difference in the music we create. On occasion, she even finds mistakes in the printed scores, and with her knowledge of music is able to correct them. Sometimes the publisher or editor has misinterpreted an ornament indication, or even put in a wrong note. Being able to detect and correct such errors makes our music sound better.

In much of the music we do here, the ornamentation is indicated simply by markings, and it is left to

the performer to interpret the markings in a correct musical manner. Our musician has studied all this with great care, so that the actual playing of the ornaments is made smoothly and appropriate for its musical context.

She often has to resort to working out the timings on graph paper, to make sure that the notes in the ornamentation work out properly timed with the longer duration notes in the bass line. One of the more challenging instances was in the *Nutcracker Suite*, where Tchaikovsky had a tied group of seven notes in the space of one quarter note. The timing for that had to be worked out on a grid where each square represented 1/28th note. Fortunately, once the timing had been worked out, the numbers could be expressed directly in the sourcefile so that the assembler did the calculations of the actual durations.

For example, if a note needs a duration of one seventh of a quarter note, we can simply enter *Q/7* in a *DATA* statement, and let the assembler take the value assigned to *Q* and divide that by seven. Of course there is some "rounding error" in the calculation, since the assembler works only in integer numbers, but the effect is minimal because the numbers are large enough. The value for a quarter note is somewhere around 1800, so dividing by seven and leaving off the part beyond the decimal point is still well less than a one per cent error, and is not a noticeable difference when the music is playing.

That leads to another important point in the way we do our music. With our methodology we can make the computer play much faster than any human player could. We try very hard to keep within the "humanly possible" limits, including our interpretation of ornaments. If our musician feels that the computer is playing something that no human could match, she'll either slow down the tempo slightly or simplify the ornament so that it's within human capability. A good friend of ours who is a very advanced amateur flutist, and who also owns a TI, has been a "guinea pig" for much of our musica work. He provides some feedback on our musical aspects, and has often remarked on the "human" quality of our work, even though played by a machine.

Of course many of you still have some burning questions as to how this assembly language stuff gets to appear to the computer as if it were simple Extended Basic programs. This was not easy to accomplish the first time, but it's really not all that complicated.

In this example, we're showing what was done in one of our numbers from Anna Magdalena's Notebook, by Johann Sebastian Bach.

12

We use a bit of trickery which we call "submerging" assembly programs:

- First we assemble the program in the usual way with the E/A module.
  - Next, we examine the first line of the object file, to discover the length of the program. That gives us a hexadecimal number, say for example > 1DC4. We use that number to determine where the program needs to be loaded under an Extended Basic program. Extended Basic always loads XB programs in high memory, at a high enough address so that the area from > FFE7 through > FFFF is left open. What we then do is to subtract the length of the assembly program from > FFE6. For our example, > FFE6 - > 1DC4 = > E222.
  - We then go back to our source file and put in an Absolute Origin (AORG) directive at the beginning of the program, which we also insure is the start point of the code. Now we re-assemble the program using that absolute origin. That's where the Loader will place the machine code in memory when we load this new object file.
  - Now the real tricky part begins. We take that address, subtract 16 from it to allow a small buffer zone in memory, giving us in this case > E212. We separate that number into its two bytes and convert each byte to a decimal number. Thus > E2 becomes 226 and > 12 becomes 18.
  - Now we go into Extended Basic, and do:  
> CALL INIT  
> IEW  
> CALL LOAD(-31952, 226, 18, 226, 18)
  - This trick makes the Extended Basic interpreter behave as if the bottom of memory is already filled with program. Now we create a one-line program such as:  
10) CALL LINK ("AM7")
  - That one line program will serve as the link to our assembly code, but first we do the following in command (immediate) mode:  
> CALL LOAD("DSK1.ANHAG7")
  - The Extended Basic loader now loads in the assembly object file ANHAG7 just after our 16-byte buffer into the higher locations in high memory. Now we simply do save under another file name, and the machine code we just loaded in is saved to disk just as if it were part of that one-line program.
- All that would be fine except that the computer would not know how to execute that CALL LINK in line 10. That's where our second trick comes in.

In the LOAD program which puts the selection menu on the screen, we execute a:  
CALL LOAD("DSK1.DEFAM")

This file DEFAM is a zero-length object file. That is, an assembly language program which contains no program as such. We use it merely as a vehicle to trick the loader into placing the addresses of all our 10 pieces into low memory at one whack. The source code for DEFAM looks like this:

```
DEF AM1, AM2, AM3, AM4  
DEF AM5, AM6, AM7, AM8  
(ETC), TRIN  
*  
AM1 EQU >FCF4  
AM2 EQU >FD4A  
...  
AM7 EQU >E222
```

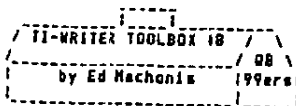
And so on. Thus the absolute address for the assembly part of each piece is loaded along with its name in the lookup table, so that  
CALL LINK ("AM7")  
causes the computer to branch directly to address > E222. That is why we insist that you run these music programs from their LOAD program, otherwise the computer will not know where to find them.

This leaves one more burning question to handle: How do we make the TI's sound generators behave like a piano, celesta, harp, and so on? Again the root of the process lies in the fact that we deal directly with the sound generators, and perform our own timing through a loop.

While that loop is executing, we can make it periodically reduce the volume on the generators that are active, and by tapering the time between these volume reductions, we can simulate the exponential decay of a struck piano string. We make different instruments by varying the overall time of the decay and the change of slope during that decay. The decay time is related to the tempo, so that in slower pieces the sustain is longer, while in faster pieces the faster decay prevents blurring of the sound. Of course the working out of the numbers for a new instrument effect takes some trial and error, but we can now produce a pretty wide variety of such effects.

The one instrument we're still struggling with is the violin. That instrument family is characterized by a triangular waveshape which is difficult to simulate with the TI's square wave generators. Stay tuned, though we're getting closer all the time.

The bottom line to all this is that we have created a new kind of music on the TI, and we're thrilled that other people enjoy it enough to buy our software. If you'd like more information, please feel free to write us.



## TI WRITER MANUALS

Some time back, TI Writer manuals were available from TI by calling TI CARES. The only charge was a \$3.00 Shipping and Handling fee. No need to send money in advance, your address on the package is an invoice. Just open it up and remit. Extended Basic manuals were also available at the same price and it may be that all manuals were being handled in this manner. You can't beat the price. I know I bought back up copies of both manuals. I don't know if any more are available but if you need a manual it would pay to check.

I don't care to go into the ethics of ordering a manual to use with Funnelweb but a mitigating factor might be that when I sent revised versions of TI-Writer to user groups, it effectively placed TI-Writer in public domain and made distribution of Funnelweb possible.

If the manuals are no longer available from TI, Triton is selling TI-Writer complete for \$9.95. The manual is worth the price. Try to combine group orders to save the heavy shipping charges.

## PARALLEL PRINTER CABLES

Did you know that TI made a Parallel Printer Cable? Neither did I until I ordered a printer cable from Triton and received a beautifully made cable TI Model No. PHA 2621.

Parallel printer cables for the TI are fairly scarce items, not available from your average printer dealer. Two that I have purchased in the past are compatible with the TI but to be revised. If you are planning to purchase a printer, it would be a good idea to purchase a cable (and an RS232 card) ahead of time. When you bring that so called TI compatible printer home you can immediately try it out and return it if necessary.

The cable from Triton at \$16.95 is a good value. Why am I telling you all this in a TI-Writer Tool Box? Many folks are not doing word processing with their TI's for lack of a printer. If you have an expanded system, this upgrade is almost a must. A journey of a thousand miles starts with the release of a hand brake!

## IT'S YOUR MOVE

The move command is one of your best aids to good composition. It enables you to place your words where they will do the most good. When I write, I frequently reread the prose I have written, especially just before saving. It serves two purposes. One, I spot a lot of typing errors and misspelled words. Secondly, I am able to check the continuity of my thoughts and see my phrases in relation to the whole.

Often I see where a sentence or two can be better placed in a different part of the essay. It's an easy Move if you know how. One enters the command line, types "M" for Move and answers the prompts "start line", "stop line" and "after line". What is being asked is the line number at the start of the text to be moved, the line number at the end of the text to be moved, and the line number after which the text is to be inserted. You must enter three line numbers, each separated from the others by a space.

The stumbling block is that there may be other text on those lines which are not part of the intended move. Since the most complex move is moving a sentence from the middle of one paragraph to the middle of another paragraph, we will use that as our example. Proceed as follows:

Make sure you are in the Word Wrap mode (Solid Cursor). Place your cursor at the start of the sentence to be moved and press FUNCTION 2 (Insert Character). The paragraph will split in two with the start of the sentence to be moved on its own line. Now cursor to the end of that sentence and again press FUNCTION 2. Again the paragraph will split and the sentence is now isolated with its own start and stop line. Take note of these line numbers.

Next cursor to the point at which the sentence is to be inserted and press FUNCTION 2. The paragraph splits; take note of the line number on which the first half of the paragraph ends. You now have all the information to make your Move. Enter Command Mode, press M and answer the prompts. Flash! It's DONE.

Not quite. You have a couple of broken paragraphs laying around. Cursor back to the point where you opened the paragraph for sentence insertion. Press CONTROL 2 (Reformat) and your paragraph is made whole. Next cursor to the break where the sentence was extracted and Reformat that paragraph. Now you are DONE.

Notice how TI paid attention to the keypresses involved. FUNCTION 2 opens the text and CONTROL 2 closes it up. Also take note that even if you have Line Numbers toggled OFF, whenever you enter command mode, the line numbers are displayed. This is a big assist with the Move, Copy, Save File and Print File commands. Meas!

Moving text from the beginning or end of a paragraph to the beginning or end of another (or the same) paragraph follows the same procedure except that fewer breaks need be made to isolate text and no break is needed at the destination. Reformat as required after the Move. Rearranging whole paragraphs is easier yet, no breaks or reformatting required. And don't forget, 0 (Zero) and E (End) ARE valid line numbers.

But that ain't all folks! You can even Move text from another file into the one you are working on. Like maybe an address or you want to quote a previous letter or append your standard "Check is in the mail" paragraph. In these cases you must use the LF (Load File) command to do your moving. FIRST note the line number after which the text is to be inserted. This will be the FIRST number used with the LF command.

Go to the command line and type SD (Show Directory) and at the prompt enter the drive number containing the file with the text to be moved. Page through

the disk catalog until you see the file containing the text you need. Press the number at the left of the filename and then press V (View). (Just about here the TI-Writer users will discover why they should be using Funnelweb. This just can't be done with TI-Writer.)

The first 2: lines of the file will be displayed and the line numbers of the displayed text will appear in the lower right hand corner of the screen. Page through the file by pressing ENTER until you locate the text you want. Determine the line numbers it resides on by noting the line numbers in the lower right corner and then counting from the top or bottom of the screen. (You may have to page further to find the end of the text you want.)

When you have the line numbers for the start and end of the text, press FUNCTION 9 to stop viewing text and then press ENTER to return to the command line. Press LF and the target file name should appear as the default. Press CONTROL V to move the cursor to the start of the file name and then FUNCTION 2 (Insert). FIRST type the number you FIRST determined after which the text will be inserted, space, the start line number, space, and or stop line number and again space. There should be three line numbers in front of the target file name. Remember, 0 and E are valid line numbers if you want to refer to the beginning or end of a file. Press ENTER and the drive will engage and the desired text moved to the designated point.

I capitalized references to that FIRST line number hoping to fix in your minds which line number is entered first for the LF command. It is also good to remember that if you want to move the entire target file into your working file, it is the only number that need be entered.

Using line numbers in a similar manner, you can move a portion of your text to your printer (with PF and then insert the line numbers in front of the printer name) or to disk (with SF and then insert the line numbers in front of the file name).

TI and the McGovern's have provided the tools. It's your move!

### TRAINING THE FORMATTER

This section is aimed mostly at Newsletter Editors although all others are invited to browse. Who knows, tomorrow may find you taking on this most important job. Many's the Editor has published corrected program listings after being fouled by the Formatter.

Most newsletter articles are run through the Formatter to reformat the margins and/or justify the right margin. However, if there are any program listings in the text, the chances are that the Formatter will take out a couple of bytes. One culprit is the asterisk, "\*". This character is used in the Mail Merge option of the Formatter and the Formatter will not print it or the two following characters. (We point to a string in the value file by including line in our text and the Formatter does not print this pointer, whether we are using the Mail Merge option or not.)

So if an asterisk appears in your text, it and the two following characters will be dropped, which can play havoc with a program listing. (The asterisk appears in most program RND statements.) You can circumvent this by entering two asterisks followed by two funny characters wherever you need an asterisk to be printed. (Whether to disk or hard copy.) But there is an easier way as you will soon learn.

Another booby trap is the ampersand, "&", used by the Formatter as an indicator to start underlining. It is also often used in programs to join strings together (Concatenation). The Formatter will not print the ampersand but will start underlining text if you are printing to hard copy. When printing to disk, the ampersand will be dropped without even the underlining clue to indicated that something is amiss. Again you can circumvent the Formatter by entering two ampersands where one is desired. And again there is an easier way.

### THE EASIER WAY?

The Easier Way is so named because you only have to do it once and you are forever protected from the Formatter's foul blows. First make a copy of your FNB or TI-Writer disk. Then using a sector editor on your copy disk, locate the first sector of file FD (FNB) of FORMAT (TI-Writer). If you have John Birdwell's excellent Disk Utilities this is very easily done. First select 1) File Utilities, then 4) File Editor, enter the file name and the drive number and you are at the first sector of the file. (Alternatively, copy the file onto a blank formatted disk. The first sector of the file will be on sector >22 (hex) or 34 (decimal).

Toggle the ASCII mode (Control A with DSKU) and slightly less than halfway down the sector (starting at byte 112 decimal or 70 hex) you will see \*!@!R232.CR. (Another way to locate the first sector of the first Formatter file is to do a string search for this string.) Now, using the space bar, blank out the asterisk and the ampersand. The at sign, "@" is used for overstriking and this could also be changed or eliminated although I have never found it used in a program. It could appear in text however with unwanted results so it is best eliminated.

Instead of blanking out these trouble makers, you could substitute other seldom used characters for them. I don't recommend doing so justly because I don't like having to remember two ways of accomplishing the same task. However, if you do a lot of cut and paste and want to retain underlining and overstriking then I suggest replacing the ampersand with the reverse slant, ASCII 92, and the at sign with the grave, ASCII 96.

Having blanked out or replaced \*, &, and @, write the sector back to disk. (Control M with DSKU) If you have copied the first Formatter file to a blank disk in order to locate the first sector of the file, copy the file back to your FNB or TI-Writer COPY disk. Mark this disk "Editor's Version". You are DONE.

14

FUNNELMED CONFIGURATION INSTRUCTIONS  
USDC Release 1.2

1) Introduction

A. These instructions are presented as a service to the TI-19/4A community by USDC - the User Group of Orange County (California). They supplement the FUNNELMED Versions documentation. This release supports Versions 4.10 (May 10, 1988), 4.11 (July 4, 1988) and 4.12 (August 12, 1988).

Comments and suggestions should be sent to:

USDC c/o  
Jill Swedlow  
7301 Kirby Way  
Stanton, CA 90680

B. Print and read the FUNNELMED documentation. Using the Formatter, print all files that start with "FNDC" and "-RED-RE".

C. These instructions assume that you will operate FUNNELMED from Drive 1. If you will use another drive, you will need to modify them accordingly.

They also assume a basic system and normal use. If you have special features or uses, you should be able to modify them to suit your needs.

D. If you need to press a key or keys, the "<C>" signs will be used (for example, "Press <ENTER>"). When two keys must be pressed together, it will show this way "Press <CTRL C>". <CTRL C> means hold the <CTRL> key down with one finger and then press <C> with another. Release both fingers simultaneously.

E. In boxes where you input information (a file name, etc), indicate that you are done by pressing <ENTER>. When this is necessary, it will be shown this way: "Enter the correct file name".

F. In Configure menus, you normally choose options by pressing the first letter. For example, the Top Menu has three choices:

Systema  
Quit  
Install

Press <S> for Systema, <Q> to Quit

and <I> to install. Such choices will be shown as "Press <B>it" or "Press <Q> to Quit".

A New layout of all Configure Menus follows these instructions.

G. <CTRL C> is used to move from the current menu to the previous menu. You can use <BACK> or <FCIN 9> instead of <CTRL C>.

A number of times these instructions ask you to press <CTRL C> to return to a previous menu. Sometimes it will be necessary to press <CTRL C> more than once.

<CTRL C> also now works in DM 1000 instead of <BACK>.

H. In any places in the Configuration Program, pressing <Q> or <FCIN 1> windows a help screens. These screens are filled with useful information. You should check them frequently the first time you configure FUNNELMED.

I. Also available in most places in the Configuration Program is the Quick Directory. Press <FCIN 7> to access the disk directory function.

J. Abbreviations:

FB FUNNELMED  
ID Extended Basic  
EA Editor Assembler  
TI W. TI Writer

(2) Starting Up

A. Make a working FMB disk. Use MH 1000 to copy the files you will need to a freshly initialized disk. A list of files and some suggested arrangements follow these instructions. Configure only your working copy. Keep in unmodified master copy "just in case".

B. Load FMB with your working FMB disk in Drive 1. If you're using IB, FMB will autoboot when you choose Extended Basic. If you are using the EA module, choose "S. Run program file" and then press <ENTER>.

C. If you loaded from IB, choose Configure from the IB Menu. If you loaded from EA, choose User List from the EA Menu and then choose Configure. If Configure is not on either menu, you can load it through the LOADERS

function. Choose option 2. The file name is DSX1.CF.

(3) Top Menu

A. Press any key to get past the opening screen.

B. Press <S> to modify the system information (Systema).

(4) System Menu

A. Press <L> to load the system configuration file. Enter the file name (DSX1.SYSCON). You can have multiple configuration files with any names you choose. To implement one, you must activate the Configuration Program, load the configuration file and then install it (see step 12).

B. Press <E> to edit the system configuration.

(5) Loading

A. Press <L> to access the Loading Menu.

B. Boot Tracking should be ON if you are loading from a disk drive. RAM disk users should turn it OFF as FMB cannot track booting in RAM disks. Press <D> to toggle Boot Tracking ON or OFF.

C. The number following "TI Writer side" is the drive number where the files that support the choices on the TI Writer Menu will be found. This number is used if Boot Tracking is OFF or if it fails.

The same applies to "Edit/Asm side". Normally both are 1. To change them, press <T> or <E> and then enter the drive number.

D. "Working Drive" is the drive number of the drive in which you will put your data disk. If you have a two drive system, this will normally be 2. Press <D> to change the Working Drive.

E. If "U. Immediate" is ON, the User List will be the first menu you will see when you load thru EA. Unless you want to access the U. Immediate, you will normally want this off. Press <U> to toggle this ON and OFF.

FUNNELMED CONFIGURATION.....Page 2

Turning this ON allows you to load FMB from EA and have a menu that you designed completely appear first. This feature was added with Version 4.11

F. When all values are correct, press <CTRL C> to return to the main Edit Menu.

(6) Devices

A. Press <D> to access the Devices Menu.

B. Press <E> for the Editor Printer. If you have a parallel printer, enter "PI". If you have a serial printer, make sure that all the switches are correct.

C. Press <F> for the Formatter Printer. If you have a parallel printer, enter "PIFLP". If you have a serial printer, make sure that all the switches are correct and that the printer name includes ".LP".

D. The Object, Work and Program file names are defaults for various FMB functions. If you enter a name, include the drive designator (for example, DSX1.LOADFILE). If you leave the field blank, the drive number you designated as the "Working Drive" in the Loading Menu will appear (for example, if you designated the working drive as 2, it will be "DSK2."). If you want the default to be a different drive than the working drive, you can enter "DSK.". Any file name you tag in the Quick Directory will override these default names.

E. Press <O> to update the Object File. This is the default file name when an object (DF BO) EA file is required (for example, when loading an object file).

F. Press <W> to update the Work File name. If you put a name here, it will be the default when you use LoadFile in the Editor.

G. Press <P> to update the Program file. This is the default when loading EA Program files (TI Pgm, GPL Pgm, EA Pgm, etc).

H. When all values are correct, press <CTRL C> to return to the main Edit Menu.

(7) Colors

A. Press <C> to access the Colors Menu.

B. The cursor will be on the first color choice. This is the screen colors that will appear when FMB boots. The other colors are those that cycle when you press <CTRL 3> in the Editor or 0 (zero) on most screens that invite selection by number.

You have several choices from this point:

<EM>it: modify the current color  
<D>ext: move to the next color on the list  
<B>ack: move to the previous color on the list  
<I>chg: exchange two sets of colors  
<R>edo: restore the colors to what they were when you started the current color  
<V>iew: view the current color

C. When all colors are correct, press <CTRL C> to return to the main Edit Menu.

(8) Menu

A. Press <M> to edit the choices on the main TI Writer and EA Menus.

B. Press <T> to edit the choices on the TI Writer Menu and <E> for the Editor Assembler Menu. You can change items 4 through 7 on both Menus.

C. For each line on the Menu, you have three options: <EM>it, <D>ext, <B>ack and <I>edo. These are the same as in Colors (see step 7b).

D. If you <E>dit a Menu line, you will be required to enter the following information.

1. NAME: This is the name that will appear on the Menu. This can be anything you want up to 10 characters.

II. FILE NAME: This is entered just to the right of the Menu name and must be only two characters. This is the file name that will be loaded when you opt for list Menu choice. You cannot enter "DSK.". FMB gets the drive number from Boot Tracking or the Loading information (see steps 5b and 5c).

III. TYPE: Here you tell FMB the file type. Move the light bar between the options by pressing <N>ext and <B>ack. You have the following choices:

TI PGM: Emulates option 3 from the TI Writer Menu. The file must be in EA Program format.

GPL PGM: This is used for most programs loaded by the "E" "Run Program File" option.

EA PGM: This is for EA "Run Program File" that need EA Utilities. In general, use "EA Pgm" if "PL Pgm" doesn't work.

SCRIPT: This allows you to write a script for loading a series of object (DF BO) EA files. See FNDC/UTIL for details.  
LOWMEM: Loads object (DF BO) files into low memory. See FNDC/UTIL for details.

LR/RM: Use this for most "Load and Run" object (DF BO) EA files.

E. When you are satisfied with the TI Writer and EA Menus, press <CTRL C> to return to the main Edit Menu.

(9) IB List

A. Press <I> to modify the Menu that appears after you load FMB from IB. The first three choices ("TI Writer", "Edit/Asm" and "B Return") cannot be modified.

B. You now have four choices:

<O>BIT: Use this option to Edit the current entries (that is, the IB List in the SYSCON file).

<I>ETCH LIST: Use this option to obtain the menu in the LOAD program. Normally this is the same as what is in the SYSCON file.

<O>AKE RESERVE: If you <F>etch a list, press <R> to make it the reserve list. Then if you later press <R>edo, this is the list that will be restored.

<R>CHG BUFFERS: Restores the previous "reserved" list.

C. During initial configuration, you should press <E>dit.

FUNNELMED CONFIGURATION.....Page 3

4. (Exit will show you the choices in the IB Menu. You have the normal keys active for moving from item to item: (Edit), (Next), (Back) and (Redo). See step 7b for definitions.

5. If you choose to (E)dit an entry, you will be required to enter the following information:

i. NAME: This is the name that will appear on the Menu. This can be anything you want up to 10 characters.

ii. BOOT TRACKING ON/OFF: Keep Boot Tracking ON if the disk with the file will be in the drive from which you loaded FWB. Turn it OFF if it will be in another drive. Boot Tracking does not work with RAH disks.

iii. SECONDARY (O/YES): If the drive for this file is different than the boot drive and if you answer YES to this question, FWB will look for the user file in the drive specified in step 5c.

iv. REMINDER N/YES: If this is YES, FWB will remind you to insert the disk with the file. If it is NO, FWB will read the file immediately. Make this YES if the disk with the file will NOT be in the named drive when you invoke this Menu choice.

v. FILENAME: The name of the file that FWB should load. Include "DSKk." in the file name.

vi. TYPE: Here you tell FWB the file type. Move between options with (Next) and (Back). You cannot leave this field as a blank. You have the choices listed in step 8diii and

IB PRGRM: This is a standard IB Program.

IB RETRN: This returns you to the IB "ready" screen. It works like the MENU command.

9. When you are satisfied with the IB Menu, press (CTRL C) to return to the main Edit Menu.

(2) UL List

a. Press (U) to edit the User Lists.

b. There are at least two User Lists to edit. The first one is called "UL" and is the .usr list that comes up when you press "B. User List" on the EA Menu.

The other one is the Disk Utilities choice in the TI Writer Menu ("3. Disk Util"). The file name is DS.

You can chain User Lists. The main User List can call another User List by asking User List a choice on the User List. FWB convention is to call subsequent User Lists UL1, UL2, etc.

c. From the main User List Menu you have the following choices:

- (E)dit Entries
- (F)etch List
- (N)ake Reserve
- (C)hng Buffers
- (S)ave UL file

The following instructions should be followed for EACH User List. At a sinious, you should modify UL and DS.

i. Press (F) to fetch the User List. Enter the correct file name (DSK1.UL, DSK1.DS, etc).

ii. Press (N) to take the fetched User List the reserve.

iii. Press (E) to edit the user list. Editing is the same as editing the IB Menu (step 9e) except that the file types "IB Program" and "IB Return" are not available.

iv. Press (CTRL C) to return to the main User List Menu.

v. Press (S) to save the User List. Use the correct file name (DSK1.UL, DSK1.DS, etc).

vi. (L)oad, (E)dit and (S)ave any other User Lists you will be using. Each User List is saved under a separate file name - revising one does not affect another.

vii. When you are done editing User Lists, press (CTRL C) to return to the Edit Menu.

(1) Edit Menu - Saving the SYSCON file

a. Press (CTRL C) to return to the Sysinfo Menu.

b. Press (S) to save the SYSCON file. Enter the file name (DSK1.SYSCON or whatever name you choose to use).

c. Press (CTRL C) to return to the Top Menu.

(1) Install Menu

a. Press (I) to invoke the Install Menu.

b. Press (L) for "LOAD IB/IB1". The file name should be DSK1.LOAD. Change it if necessary.

c. Press (ENTER) to load the Source File.

d. When prompted, press (ENTER) to save the Target File.

e. Press (F) for "FW/UTIL others". Change the file name from DSK1.FW to FWK1.UTIL.

f. Press (ENTER) to load the Source file.

g. When prompted, press (ENTER) to save the Target File.

h. Press (CTRL C) to return to the Top Menu.

(3) Final Steps

a. Press (Q) to Quit.

b. Exit FWB and then reload it. The changes you have made will not appear until you reload the program.

c. Check all Menu choices to make sure that they work and that they look the way you want them to. If you go back to Configure to change anything, you only need to change those items in question. After you have (S)aved the SYSCON file, you must (I)nstall the revisions into LOW and UTIL.

d. Load DM 1000. From the first screen, press (FUTN 3). Make sure that the printer name is correct. If you change any information, answer (Y) to the save to disk question.

e. You are almost done. Make a back up copy of your configured FWB disk. If your working copy blows up, you won't have to go through all these steps to reconfigure it.

You now have three disks the master, the working copy and the back-up working copy.

f. NOW YOU ARE DONE.

FUNNELMED CONFIGURATION.....Page 4

FUNNELMED MENU LAYOUT

TOP MENU

Sysinfo  
Quit----->Exits Configuration Program  
Install

INSTALL MENU

LOAD IB/IB1  
FW/UTIL Others

SYSDINFO MENU

Load--->Loads SYSCON file  
Edit  
Save--->Save edited SYSCON file

SYSDINFO EDIT MENU

Loading  
Devices  
Colors  
Menu  
IB List  
UL List

LOADING MENU

Boot Tracking ON  
TI Writer file 1  
Edit/Assn file 1  
Working Drive 2  
UL Immediate OFF

DEVICES MENU

Edit Printer  
Futr Printer  
Object File  
Work File  
Program

COLORS MENU

Edit  
Next  
Back  
Ichg  
Name  
View

MENU MENU

TI Writer file  
Edit/Assn file  
Edit Choices

IB LIST MENU

Edit entries  
Fetch list  
Make reserve  
Ichg buffers

Name  
File name  
Types  
TII Pgn  
SPL Pgn  
E/A Pgn  
Script  
Low Mem  
L/R Run

Edit Menu  
Edit  
Next  
back  
Info  
Edit Choices

USER LIST MENU

Edit entries  
Fetch list  
Make reserve  
Ichg Buffers  
Save UL file

Name  
Boot Tracking ON/OFF  
Secondary NO/YES  
Insider NO/YES  
Release DSKn,MMn  
Type: TII Pgn  
SPL Pgn  
E/A Pgn  
Script  
Low Mem  
L/R A/N

The User List Edit Menu is the same as the IB Edit Menu except that IB Pgrn and IB Retn are not available.

FUNNELMED FILES

File	Purpose	SS50	SS50	SS50	Note
-READ-HE AS/AT CYPF110	Docs file Assembler c99 Utility				1 2
CF/CH CHAR1 CHAR2	Configuration Pgn TI Wr Characters EA Characters				1
CP CTGRAN DP	c99 Loader Cartridge RAM Loader Disk latched				2 2
DS EA EB/EE	Disk Utilities EA Loaders TI Wr Editor				
FWHVE FW/TP FWDIC/EASH	Save Utility TI Wr formatter Docs File				1
FWDIC/LOAD FWDIC/REPT FWDIC/TIMR	Docs File Docs File Docs File				
FWDIC/UTIL FWSAVE LDFW	Docs File Save Utility Aux Load Program				1 3
LN LL LOAD	Line Number Low Mem Loader IB Load Program				1
MB/WH QB SAVIT	DM 1000 Switch Directory Sample script file				
SL SYSCON UL	Script loader System configuration User List				
UTIL VB4TH,3	Loads FW from EA FW Loader				2

1. These files are normally used only for Assembly programming.
2. Used with named programs. See FWDIC/UTIL for details. Includes only if needed.
3. LDFW can be used to load FWB from BASIC in EA, IB II or Mini Memory. See FWDIC/UTIL for details.

16

## ROOM PROBLEM

by

HAROLD HOYT

EXPLORING LOGO BY: CHARLIE MIDLIFF  
PART I: LOGO IS MORE THAN TURTLE GRAPHICS

As you are no doubt aware, Logo is a language which has received considerable attention as an approach for introducing children to the computer. Although I had owned TI Logo II for some time, I had done little with it. Most of what I had read and seen of Logo was devoted to turtle graphics and did not seem of much interest. Recently, I began to look at Logo and realized that it is a language with the potential for some serious programming as well as being educational and fun. The capabilities for graphics, music and spirals on the TI may be unmatched by implementations on most other personal computers. In addition, many of Logo's features are implemented on other systems in a more limited manner or not at all.

What I would like to do is to start a series exploring various aspects of Logo programming. Because of my limited experience with the language, if the series is to be worthwhile, your input is needed. Graphics, games, educational programs, programming tips etc. are welcome. With about a thousand members in the Mid Atlantic 99'ers, there must be a bunch of Logo users in our group. Share your ideas, they don't need to be fancy, and I will pass them along. Also, if you have Logo on another computer rather than, or in addition to, the TI, your comments regarding features, program conversion etc. would be of interest.

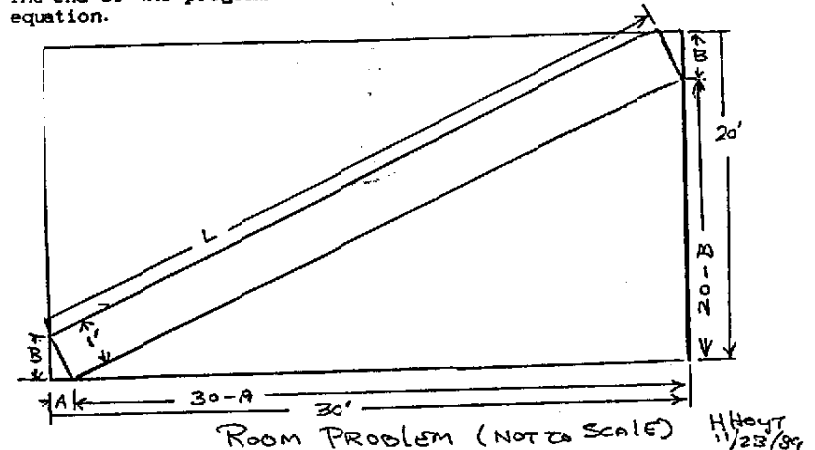
For openers, there are at least three books dealing with TI Logo and others for Commodore 64 and Apple, the TI books are:

1. Conlan, J. and Inman, D.  
"Sprites, A Turtle and TI Logo"  
Reston Publishing Company/Prentice Hall 1984
2. Ross, P.  
"Introducing the Logo for the Apple II, Texas Instrument 99/4A and Tandy Color Computer"  
Addison - Wesley 1983
3. Thornburg, D.E.  
"Computer Art and Animation: A User's Guide to TI 99/4A Color Logo"  
Addison - Wesley 1983

If you know of others and/or have thoughts on the value of these for the beginner, intermediate or advanced Logophile, let us know. The major problem with the several Logo books I have examined is that they focus on turtle graphics and only consider other aspects of the language in the last few pages. This is unfortunate because TI Logo has some excellent features for interactive programs and list processing. One limitation of TI Logo is its restriction to integer operation. This severely limits mathematical programs unless you are really dedicated.

For the math buffs. Given a room 20' x 30', what is the longest board, 1 foot wide, that will fit in the room? The 4 corners of the board will just touch the 4 walls of the room, as shown in the picture. If you use trigonometry, algebra, and even calculus to analyze the room, the equations get very messy. I was able to get two independent equations for L, the length of the board. One equation is made by summing the area of the room. The total area is 30' x 20' = 600 sq ft. The pieces, 4 triangles and a rectangle, total  $(30-A)*(20-B)/2 + (30-A)*(20-B)/2 + A*B/2 + B/2 + L*1$ . (The area of a right triangle is equal to 1/2 times the base times the height.) So  $600 = (30-A)*(20-B) + A*B + L = 600 - 30*B - 20*A + A*B + L = 600$ . Re-arranging leaves  $L = 20*A + 30*B - 2*A*B$ . The second equation, used to check the first uses the Pythagorean equation, where the Hypotenuse of a right triangle is equal to the square root of the sum of the squares of the sides. That is  $L = \text{SQRT}((30-A)^2 + (20-B)^2)$ .

In the program, we use the fact that L is a function of A and B. A is varied, B is calculated for each A, and substituted in the equation for L. The maximum value of L, and the A that goes with it, are stored as LMAX and ANAX. AMAX is used to revise the search area of A. This goes on until L is accurate within the limits of the square root routine in XEasic. Line 100 sets A=0.1 and the step size DA=0.1. The number of steps are set at 9. This equation manipulation corresponds to placing a board in the room, stretchable in length to touch at all 4 corners, and turning it and measuring the length. Two nested loops, I and J, test for LMAX. J, the inner loop, increments A over the test range, and the I loop divides the test step size by 10 and resets the test range. This technique should work, even for poorly behaving functions as long as it is possible to increment the independent variable over a suitable range in the first pass. The end of the program verifies the fit by testing A and B in the second equation.



COMPUTER BRIDGE, DECEMBER 1989.

ROOM PROGRAM

by Harold Hoyt

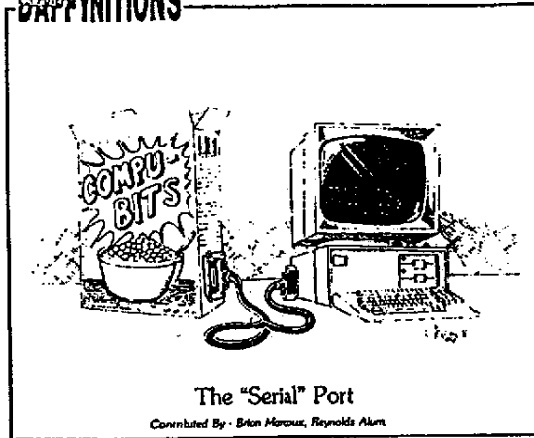
```

1 !SAVE DSK1.ROOM !080
100 !H Hoyt Room Problem 11/
27/89 !247
110 A=0.1 :: DA=0.1 :: N=9 !
008
120 FOR I=1 TO 7 :: FOR J=1
TO N !077
130 B=SQR(1-A*A):: L=20*A+30
*B-2*A*B !182
140 PRINT L;TAB(14);STR$(1E-
5*INT(1E5*A));TAB(22);STR$(1
E-5*INT(1E5*3))!178
150 IF L<LMAX THEN 170 !228
160 LMAX=L :: AMAX=A :: BMAX
=B !142
170 A=A+DA :: NEXT J !232
180 A=AMAX-DA :: N=25 :: DA=
0.1*DA !147
190 NEXT I :: PRINT LMAX;AMA
X;BMAX !230
200 L=LMAX :: A=AMAX :: B=BM
AX :: PRINT SQR((30-A)*(30-A
)+(20-B)*(20-B))!167
210 PRINT 30-A;20-B;A;B !155
220 PRINT SQR(1301-60*A-40*B
)!088

```

Sooner 99ers  
 PO Box 61061  
 Oklahoma City, OK 73146

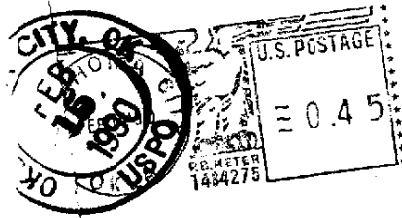
DAFFYNITIONS



The "Serial" Port

Contributed By: Brian Marcus, Reynolds Alum  
 Send YOUR computer definitions to: Gorge Marcus, c/o HORIZONS

LTPA328



DALLAS TI HOME COMPUTER  
 P.O. BOX 29863  
 DALLAS, TX  
 75229

# SOONER 99ERS

This newsletter is the official publication of the **SOONER 99ers**  
 PCB 61061  
 Oklahoma City, OK 73146  
 Sooner 99ers BBS#:  
 (405) 672-8270 300/1200 8N1

Club Officers:  
 Dave Lewis President  
 (405) 329-2274  
 Robert Stepp Vice President  
 Garth Potts Acting Secretary  
 Mark Mitchell Treasurer  
 Jerry Robertson Librarian  
 Harry Peterson Newsletter Editor  
 (405) 721-6930

January, 1990

Hello fellow club members (including clubs we exchange with). Welcome to the New Year! This is the time to make plans, resolutions, and decide what changes will come with the new year. 1990 will certainly bring changes and it is up to us to decide if those changes will be for the better or ..... ?

We are still in the process of selecting new officers for the coming year as are many other clubs around the country. If you are willing to help in any way, PLEASE, volunteer! We should be willing to serve and offering to help makes it easier and more enjoyable.

I have taken the easy way cut again and copied from other newsletters. In this case, we have some 'oldies but goodies' reprinted from some of the MANNERS (Mid-Atlantic Ninety-Niners) fine newsletters.

Opinions expressed in this newsletter are those of the editor or authors. Articles, unless otherwise noted, are written by the editor.

A few months ago, I received a letter from Jack Erice. I think it simplifies a dilemma many members seem to be facing: IBM/'nother computer! If you have considered buying another machine, give serious consideration to Jack's ideas and your purpose for computing.

I would now like to publish my 'wish list'. (things I'd like to see and would buy):

- 1: A good spell checker (I realize Press should contain this but...). It should provide a method of modifying the MAIN dictionary, maybe a utility for the Dragon-slayer program would suffice.
  - 2: A better TI-Artist PLUS print utility (I know, I'm not the first to notice this). I have had a short opportunity to work with TIA+ and it looks GREAT!!!
  - 3: A challenging game I could win. Not just a shoot-em-up reflex type.
- Well, until next month, bye. BP

Advertising related to the TI-99 or Myarc 9640 will be accepted on a 'space-available' basis from individuals or companies with no payment required.