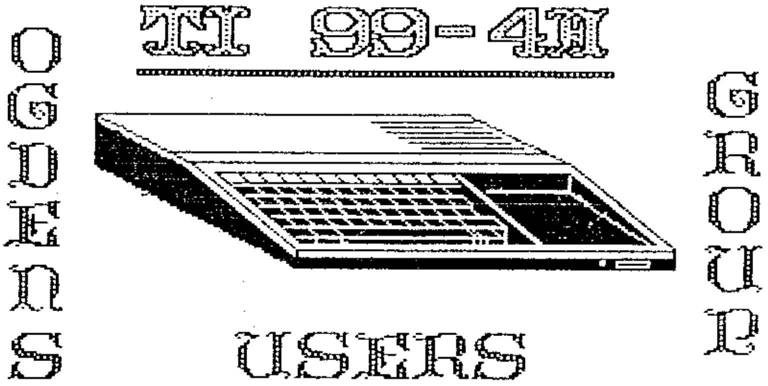
Video chess back

TI OGDENS USER GROUP

a publication of the ti 99-4a and 9640
ogden users group inc.

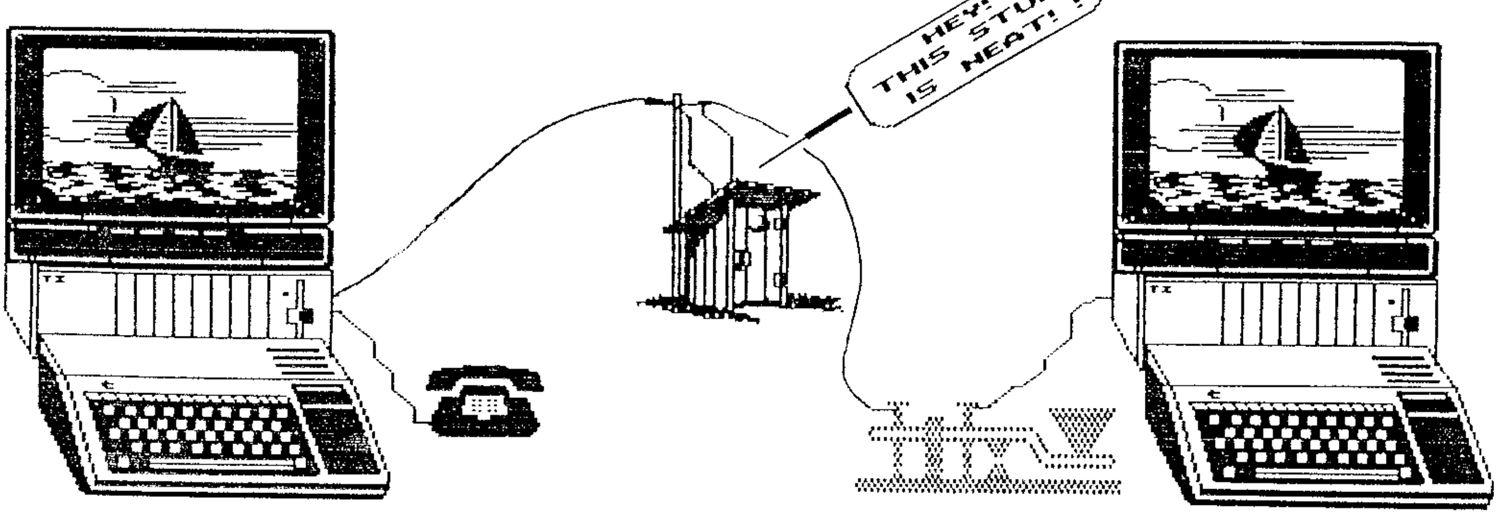


THE CLUB THAT REFUSES TO SAY GOODBY

TI OGDENS USER GROUP

JUNE 1989

COMMUNICATING



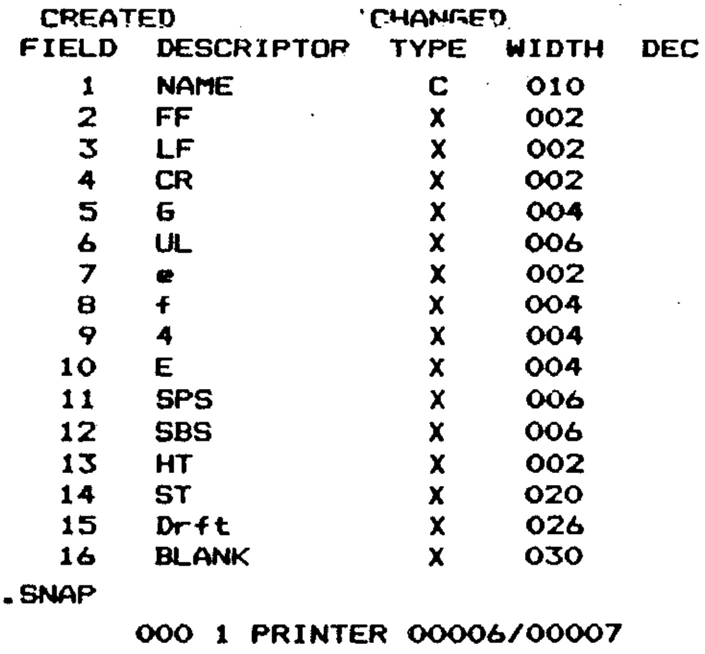
COME TO USER GROUP MEETINGS

TI—BASE - From INSCEBOT
TUTORIAL 8.1 By Martin Smoley
NorthCoast 99'ers - March 12, 1989

I am reserving the copyright on this material, but I will allow the copying of this material by anyone under the following conditions. (1) It must be copied in its entirety with no changes. (2) If it is retyped, credit must be given to myself and the NorthCoast 99ers, as above. (3) The last major condition is that there may not be any profit directly involved in the copying or transfer of this material. In other words, Clubs can use it in their newsletters and you can give a copy to your friend as long as its free.

*** More on Printer Controls ***

Last month I thought I covered printer controls, but then I tried to use them. I made a real mess of things. At that point I decided that instead of trying to remember another set of printer commands, I would change the TIB commands to the type I use the rest of the time. I normally use directly inserted codes which can be found on page 146 of the original TI-Writer manual. I placed these characters in the right hand column of last months FunnelWeb TI-Base Control Code sheet. After firing up TI-Base I typed USE DSK5.PRINTER, which is where my PRINTER DB is located, and then MODIFY STRUCTURE. This brought up the PRINTER STRUCTURE SCREEN, which I then modified to look like the one below.



```
 CREATED              CHANGED
FIELD  DESCRIPTOR    TYPE   WIDTH   DEC
  1     NAME          C      010
  2     FF            X      002
  3     LF            X      002
  4     CR            X      002
  5     G             X      004
  6     UL            X      006
  7     e             X      002
  8     f             X      004
  9     4             X      004
 10     E             X      004
 11     SPS           X      006
 12     SBS           X      006
 13     HT            X      002
 14     ST            X      020
 15     Drft          X      026
 16     BLANK         X      030
.SNAP
       000 1 PRINTER 00006/00007
```

You can type over the name and press FCTN 8 to save your changes. Do not change anything in the width column or you will lose the data in the file. Next I type EDIT <E>, to get into the file and make one change. When the EDIT Screen came up I pressed FCTN 5 to get to the EPSON record and I pressed <E> until the cursor was in the ST field. I then entered 1B4428 followed by all zeros, and FCTN 8 to save my changes. This changed my ST (Set Tab) field to set one tab at 40 columns. 1B44 means Set Tab, and 28 is Hex for 40. The zeros are null bits and don't do anything. I then typed CLOSE ALL <E>, to get out of the PRINTER DB. I had previously removed the asterisk (*) from the beginning of the PRINTER EPSON line in my SETUP CF, so I typed DO DSK5.SETUP. The SETUP file ran and I saw the PRINTER flash on the screen so I new that my new commands were in place. NOTE: I have placed a copy of the PRINTER database at the bottom of this page with my modification to the ST field. I chopped off the BLANK field to make it all fit on one line. The BLANK field contains 30 zeros. You cannot print out your PRINTER DB because of the special X-type field designation. To print this out I place a copy of the two PRINTER files on a disk in drive #1 and entered these commands. COPY DSK1.PRINTER/D DSK2.C_PRNTR/D GO <E>, then COPY DSK1.PRINTER/S DSK2.C_PRNTR/S GO <E>. If you have a two drive system, you must also have a copy of the TIB system in Drive #1, or you can make the copy directly from your TIB system disk. These commands will cause TIB to COPY the printer files to two new files. You can now enter USE DSK2.C_PRNTR <E>, and the MODIFY STRUCTURE <E>. At this point change all of the X-type designations to C, for Character, and FCTN 8 to save the changes, and FCTN 9 to get out of that mode. At that point I entered TOP <E>, then PRINT (f) and PRINT ALL. My printout was roughly what you see at the bottom of this page except for added line because of field lengths. With this printout you can visualize the complete PRINTER database and see where your own particular printer fits in. Remember to modify the original PRINTER database if needed. The C_PRNTR database is completely useless except for a printout.

I really hope this helps some of you get into the swing of printer controls. I spent a lot of time and messed up a lot of data in order to produce this tutorial.

Hopefully you now have modified your printer control database and have put it in use by removing the asterisk from the SETUP file on your system disk. Now let's put the PRINTER COMMANDS to some use. I had a specific question at one time that I will answer in the same Command File. The question was, "How can I print more than one label across? I got some labels cheap that are two across."

Continued Next Page.

PRINTER DATABASE

REC	NAME	FF	LF	CR	G	UL	e	f	4	E	SPS	SBS	HT	ST	Drft	BLANK
0006	DIABO	0C	0A	0D	0000	000000	00	0000	0000	0000	000000	000000	09	00000000 000000000000	00000000000000000000000000	0000000000000
0000	EPSON	0C	0A	0D	1B47	1B2D01	0E	000F	1B34	1B45	1B5300	1B5301	09	1B442800 000000000000	1B481B2D3014121B351B461B54	0000000000000
0003	MX-80	0C	0A	0D	1B47	000000	0E	000F	0000	1B45	000000	000000	09	1B440A0A 0A0A0A0A0A00	1B4814121B4600000000000000	0000000000000
0002	NEC	0C	0A	0D	0000	000000	0E	1B51	0000	1B21	000000	000000	09	1B283031 302C30320000	1B220F1B4E0000000000000000	0000000000000
0004	OKIDATA	0C	0A	0D	1B48	1B43	1F	001D	0000	1B54	001B4A	001B4C	09	00000000 000000000000	00000000000000000000000000	0000000000000
0005	PROPRINTER	0C	0A	0D	0000	000000	00	0000	0000	0000	000000	000000	09	00000000 000000000000	00000000000000000000000000	0000000000000
0001	TI-850	0C	0A	0D	1B47	000000	0E	000F	0000	1B45	000000	000000	09	1B440A0A 0A0A0A0A0A00	1B4814121B4600000000000000	0000000000000

TI-BASE - From INSCEBOT
TUTORIAL 8.2 By Martin Smoley
NorthCoast 99'ers - March 12, 1989

```
* Command file 2WLBLS/C
* Copyright Martin Smoley 1989
*
* SET TALK OFF
* TRACE ON
SET RECNUM OFF
SET HEADING OFF
SET PAGE=000
CLEAR
COLOR WHITE,DARK-BLUE
LOCAL XPDT C 38
LOCAL NAME C 38
LOCAL ADDRS C 38
LOCAL CTSTZ C 38
LOCAL TEMP C 38
USE TNAMES
TOP
PRINT (ST),(G)
WHILE .NOT. (EOF)
   REPLACE XPDT WITH "                 " ;
     ! "          Exp. Date " ! XP
   REPLACE NAME WITH TRIM(FN) ! " ";
            ! MI ! " " ! LN
   REPLACE ADDRS WITH SA
   REPLACE CTSTZ WITH TRIM(CT) ! ".";
            ! ST ! ". " ! ZP
   MOVE
 IF (EOF)
   PRINT (G),XPDT,(Drft)
   PRINT (CR),(LF)
   PRINT (E),NAME,(Drft),(G)
   PRINT (4),ADDRS
   PRINT CTSTZ,(Drft)
   PRINT (CR),(LF)
 ELSE
   REPLACE TEMP WITH "                 ";
     ! "          Exp. Date " ! XP
   PRINT (G),XPDT,(HT),TEMP,(Drft)
   PRINT (CR),(LF)
   REPLACE TEMP WITH TRIM(FN) ! " ";
            ! MI ! " " ! LN
   PRINT (E),NAME,(HT),TEMP,(Drft),(G)
   REPLACE TEMP WITH SA
   PRINT (4),ADDRS,(HT),TEMP
   REPLACE TEMP WITH TRIM(CT) ! " ";
            ! ST ! ". " ! ZP
   PRINT CTSTZ,(HT),TEMP,(Drft)
   PRINT (CR),(LF)
   MOVE
 ENDIF
ENDWHILE
CLOSE ALL
DO DSK5.SETUP
RETURN
*
* CF to print two across labels
```

The Command File on this page is the working result of my effort for this month. It is probably not the most efficient as far as programming is concerned, but it does print labels two across and it also demonstrates the use of the printer controls in the Command File mode. If you wish to type this CF in and use it, it is too large for the TIB Editor, so it must be entered with FunnelWeb in the non-wordwrap mode. Lines 4 and 5 do not execute because of the asterisk in the first column. If you run into a lot of bugs, you can find many of them by watching the program lines scroll up the screen. When the CF works well, remove the asterisk from that line to turn the screen junk off. If your problems are massive, as mine were, remove the asterisk from the TRACE ON line and all lines which are executed will also be sent to the printer, along with line numbers. The printout will help you find your problems. When the CF runs well remove the TRACE line completely. The line PRINT (ST,(G) is the first line to issue printer controls. It sets the printer tab at 40 columns and turns double strike on. Note: This line can be typed at the Dot Prompt and executed provided a DB is in use at the time. I am using TNAMES which contains 5 records. "WHILE .NOT. (EOF)", is roughly the same as saying to TIB, if you have any data right, now load the respective fields into XPDT, NAME, ADDRS and CTSTZ, then MOVE to the next record. The IF statement says, IF you have turned up the EOF marker with that MOVE, then print out the fields you have and jump to the end or ENDIF. This will print one label. If the MOVE has not brought us to the EOF marker, TIB will jump to the ELSE statement. The ELSE statement is the same as saying, we must have another record so proceed with the statements right after the ELSE, which will produce two across labels. TEMP is reused for each line so that portion works like this. REPLACE TEMP WITH " Exp. Date" XP, is filling TEMP with items from the second record because of the MOVE statement. Therefore, PRINT (G),XPDT,(HT),TEMP,(Drft) says this; set double strike on, print whatever is in XPDT (which we gathered from the previous data record), Horizontal Tab to column 40, print TEMP (which contains data from the second record, and last, change the printer back to Draft mode. This will print across the page side by side with a tab of 40. The next line prints a Carriage Return and a Line Feed. I like the PRINT (CR),(LF), it eliminates the need to initialize space for BLNK, which is what I previously used for a line feed. This process continues until the two labels are printed, another MOVE is executed and the whole process starts over. When the EOF is reached the WHILE statement no longer executes, and in my case the SETUP file is run. I have started using my SETUP CF at the end for other Command Files. I turn on the RECNUM etc. and it DISPLAYs the STATUS so I can see exactly what is happening. This way I always return to a system I am familiar with, and it's easy to do. You may not want to use my symbols for control codes, (G) double strike, (f) condensed etc., but you should have enough information at this time to set up whatever you want and be able to use whatever you do set up. I am beginning to like this type of printer control, because it is available at all times. I do not need to jump to FunnelWeb to insert special characters. I have not tried this yet, but I believe that the control codes for your personal system can be greatly expanded by dedicating the whole PRINTER database to one printer. For example, you would not have DIABLO, EPSON, MX-80, etc. You would have EPSON1, EPSON2, EPSON3 etc. and fill the whole database with EPSON commands. With the PRINTER database on the PRGDISK you could then SELECT an unused slot within a running CF and execute PRINTER EPSONn (n=1-0). At that point TIB would load another 15 printer commands which you could use. Interesting thought isn't it. Next Month.

FLUG - May 1988

TI-WRITER TIPS #1
- by Bob Seddon -

EDITOR MARGINS VERSUS FORMATTER MARGINS

There are two kinds of margins in TI-Writer: Editor Tabs & Formatter Dot commands.

EDITOR TABS

When you create text with the Editor you use margins called Tabs. Tabs are set via CTRL c (PROMPTS), t (Tabs), Enter. This sequence of keystrokes makes the Tab Line appear across the top of the screen. You can type over the Default settings at 0 and 79 and repositon L and R to make on-screen tabs any width within that range.

PRINTING WITH THE EDITOR

You can print text created in the Editor with the Editor itself by CTRL c (PROMPTS), f (FILES), pf (Print File), Enter. This sequence of keystrokes prints text with margins equal to the tab settings; the printer output resembles the screen.

However, there are advantages in NOT using this method to print. If you use the Formatter instead of the Editor you can print lines longer than 80 spaces. You can automatically number successive pages. You can put Headers at the top of each page, Footers at each bottom. You can make the R margin flush with the .FI;AD command. You might want to use the Ampersand to underline, the Each to Print Bold. You can double space and set page length.

PRINTING WITH THE FORMATTER

To print through the Formatter you must Save the file, Exit the Editor, Load the Formatter, reLoad the file, and then print. The Formatter prints the file according to the Dot Command instructions.

FORMATTER DOT COMMANDS

Formatter margins are also set in the Editor, but not the same way as the Tabs. Instead, Formatter margins are typed in (usually on line 0001) as Dot Commands (.LM n;RM n). Dot Command margins (if present) override Tab margins when text is printed through the Formatter. If there are no Dot Commands the file will print out according to the Tab margins. The Formatter follows the Dot Command instructions but does not print the Dot Commands as it does text. The Editor, on the other hand, not only ignores Dot Commands but also prints them just as it will any other text, since it cannot make the distinction between Dot Commands and regular text.

Quite often you will want to print your text with margins EXACTLY the same as on-screen. There are at least two reasons you might want to do this:

(1) HYPHENS

If you pack in as much text per line as possible you will want to break words and hyphenate them; if you do so, the final printed output must break the words at the same place you did. Other- wise, your text will take on the appear- ance of this par- ticu- lar sentence.

(2) MULTIPLE COLUMNS

If you create text with narrow columns so that you can put several parallel columns on one page you need to count the EXACT number of lines. Line numbers down the left column give you this number (minus the lines devoted to printer commands) if your on-screen equals your printed work. (NOTE: see box at end of article about the advantages of narrow columns.)

There is a trick you must use to make your on-screen work created in the Editor resemble the printed output of the Formatter so that each resembles the other line-by-line..

R TAB ONE # HIGHER THAN .RM

If you use the Formatter you must set the R Tab one digit higher than the setting of the .RM dot command. The difference between the Formatter and Editor is that the Formatter will print ON the .RM column. The Editor prints UP TO (but not on) the R Tab.

TAB SETTINGS:	0 & 31
SPACES USED ON-SCREEN:	0 - 30
(31 ACTUAL SPACES OCCUPIED)	
DOT SETTINGS:	0 & 30
COLUMNS PRINTED ON:	0 - 30
(31 ACTUAL SPACES OCCUPIED)	

In this article I set the Editor margins at 0 and 31 to fit three columns on the page. The Formatter settings are at .LM 0;RM 30. You can see that the columns printed at a width of 31, not 30.

```
 123456789 123456789 123456789
L....T....T....T....T....T....T
```

When counting, remember to begin ON zero: call the 0 a 1, 1 a 2, 2 a 3, etc. R follows the last T. We cannot print it here for the very reason being discussed! Counting the number of spaces used in each line on printed work is not difficult. Nor is it hard to count spaces used on screen. It is even easy to memorize the rule for making Editor margin width one character longer than dot command margin width. The one frustrating thing about this whole business is allowance for a L margin on zero!

Instead of making .RM one digit smaller than the R tab setting you have the option of achieving the same effect by making the .LM one digit greater than the L Tab of the Editor. The option of using zero as a L Tab adds confusion to this issue because you can also set .LM at zero; thus, this tactic is of no particular benefit for you.

SPACE(S) AFTER . : ? ! REFORMAT VERSUS .FI

When you Reformat, the Editor packs in text according to its own set of rules, rules different from the Formatter .FI command. If you leave only one space after a period, the Editor's Reformat command will NOT increase the spacing to two places. The Formatter, on the other hand, ALWAYS leaves two spaces after periods, whether you want it to (at sentence ends) or not (after initials).

We can prevent the Formatter from increasing the single space after the final dots of initials, abbreviations, etc. by putting a carat between such dots and the next letter.

Similarly, we need to force in two places after :, ?, and ! If you merely leave two spaces after each one the Formatter will reduce your two spaces after each of these down to one unless you follow them with a carat, then the space. Optionally, you can key in two carats (and no space).

SPACES LEFT AFTER:

	period	initial	?!:
Editor Reformat	1	1	1
Formatter .FI	2	2	1
Remedy	space twice	carat once	carat twice

FORCED IN CARRIAGE RETURN

The Formatter makes a decision to Wrap based on the R tab setting and whether a word (or ANY group of characters) occupies or exceeds that setting. Usually the last PRINTED character in a paragraph is a period and if it falls on the last occupiable position (R tab setting minus one) you must be careful where you place the carriage return.

(1) If you space once after the period, then Key CTRL m, there is no problem.
(2) If you cursor down below your text, then Key CTRL m, there is no problem.
(3) If you key CTRL m in the position directly following the period, the last word in a paragraph will not fit at the end of the line and will drop to the next line.

When it drops, you notice that it SHOULD fit, even when you account for the space before the word and the period following. The Editor Wraps the word around to the next line because it treats the carriage return following the period as part of the word, even though the carriage return is not a printed symbol.

If this happens to you, you must break the text after the period and before the carriage return (CTRL g), then Reformat. The word will now NOT wrap to the bottom line. The carriage return also moves up to the original line.

FORCED IN FORMATTER COMMAND

A similar problem occurs when you precede a word with an ampersand, carat, or @. Let us consider the ampersand which is, of course, a Formatter command to underline any word it precedes. The Formatter .FI Command ignores the ampersand and packs in Text as though the ampersand were not there. Unfortunately, the Editor treats the ampersand as a regular character when Reformatting and, so, will make a decision to Wrap a line based on the presence of it within a line of text, just like the carriage return. This anomaly makes it difficult to create a line of text which appears on screen exactly as it will print.

There is a technique to insert these codes in front of any (or even every) word on the line. Unlike the carriage return which FOLLOWS a word, a Formatter command PRECEDES it, making the previous technique impossible. Turn off wordwrap with CTRL 0. This turns your cursor into a hollow rectangle. Move the cursor to the letter before which you want an &. Key in Insert (FCTN 2). Key in the ampersand. Everything right of it will move right one column. This is the only way you can make a character appear ON column R.

You can only insert one such ampersand per line using this trick UNLESS YOU RESET THE EDITOR R MARGIN TO A HIGHER NUMBER. If you insert an additional character anywhere else on the line and do not first increase the R margin the last character on the line will vanish. You can precede EVERY word in the line with a non-printable character so long as you increase the R Tab enough so that all text and all codes fit on that line. The only restrictions which apply are that you may not mix text and code such that you excede 80 spaces; nor can you Reformat afterwards.

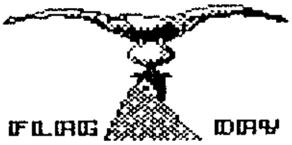
THE NARROW COLUMN ADVANTAGE: NO WINDOWING

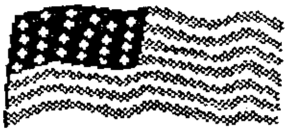
I find it convenient to set on-screen margins so I can see all text without Windowing left and right. Since I also like to leave the four-digit column numbers on the left side of the screen at all times, the highest possible R margin setting is 34. (34 is off screen, but Wordwrap causes text to occupy 33 by briefly Windowing right, then left, when you type on 33 itself.)

You can turn off the column numbers (FCTN 0) to see six more columns of text (4 digits and 2 spaces) which lets you set the margins at 0 and 39. You can even set R at 40 (which is off screen) so that Wordwrap will cause text to occupy 39; this makes a REAL 40 column screen. Naturally, the R Tab can be reset anytime to Reformat to any desired margin width up to the on-screen maximum of 0-79.

79 COLUMN SCREEN

I am sorry to break the news to you that you only have a 79 column screen, not the 80 column screen you thought you did! When in Wordwrap the Editor does not let text occupy the column of the R margin (on screen or when printed with the Editor). If you create text with Wordwrap on, the Editor alone cannot print 80 columns: its extremes are 0 and 79, and 79 is not printed on. The arithmetic is tricky because of the presence of the zero. If your Editor margins are on 0 and 79 you can only print 80 column text IF your dot commands are set at 0 and 79! AND IF you use .FI; nor will on-screen equal output.





#57

Tigercub Software
156 Collingwood Ave.
Columbus OH 43213

I am still offering over 120 original and unique entertainment, educational and utility programs at just $1.00 each, or on collection disks at $5.00 per disk.

The contents of the first 52 issues of this newsletter are available as ready-to-run programs on 5 Tips Disks at $10 each.

And my three Nuts & Bolts Disk, $15 each, each contain over 100 subprograms for you to merge into your own programs to do all kinds of wonderful things.

My catalog is available for $1, deductable from your first order (specify TIGERCUB catalog).

****

TI-PD LIBRARY

I have selected public domain programs, by category, to fill over 200 disks, as full as possible if I had enough programs of the category, with all the Basic-only programs converted to XBasic, with an E/A loader provided for assembly programs if possible, instructions added and any obvious bugs corrected, and with an autoloader by full program name on each disk. These are available as a copying service for just $1.50 postpaid in U.S. and Canada. No fairware will be offered without the author's permission. Send SASE for list or $1, refundable for 9-page catalog listing all titles and authors. Be sure to specify TI-PD catalog.

****

I like little programs that load quickly and do just what I want to do at the moment. And one of the things I wanted to do quickly was to find phone numbers. So, I used FUNLWEB to create a little file -

SMITH,JOHN (999) 111-2222
BUSH, GEO. (000) 123-1234
GHADDAFI, O. (666)66-6666

and all my other frequently called numbers. I SAVEd it as DSK1.PHONELIST and wrote this little routine to use it.

```
100 CALL CLEAR
110 OPEN #1:"DSK1.PHONELIST"
,INPUT
120 DISPLAY AT(12,1):"LAST N
AME?" :: ACCEPT AT(14,1):N$
130 LINPUT #1:M$ :: IF POS(M
$,N$,1)<>0 THEN DISPLAY AT(1
6,1):M$ :: RESTORE #1 :: GOT
O 120
140 IF EOF(1)<>1 THEN 130
150 DISPLAY AT(16,1):"NAME N
OT FOUND" :: RESTORE #1 :: G
OTO 120
```

Now actually, that was all I needed,(even though it did take several seconds to find a name at the end of the file), and it was easy enough to load the file into FUNLWEB when it needed updating. But, programmers are never satisfied, so I decided to write a self-contained program -

```
100 CALL CLEAR
200 DATA "ALDA, ALAN 888-999
9"
201 !@P-
300 DATA "BUSH, GEORGE 111-1
111"
400 DATA "PRESLEY, ELVIS 000
-0000"
499 !@P+
500 DISPLAY AT(12,1):"LAST N
AME?" :: ACCEPT AT(14,1):N$
600 READ M$ :: IF POS(M$,N$,
1)<>0 THEN DISPLAY AT(16,1):
M$ :: RESTORE 200 :: GOTO 50
0
700 ON ERROR 800 :: GOTO 600
800 DISPLAY AT(16,1):"NAME N
OT FOUND" :: RESTORE 200 ::
GOTO 500
```

That funny thing in line 201 turns off the prescan and speeds up initialization. This routine is no faster than the last, but can be updated by editing the program itself. It is limited to about 500 records due to the least-known and greatest weakness of the TI, that string storage is limited to console memory.

But, computer users are paranoid about speed, so I decided to put my data into a pre-loaded array with self incrementing subscript numbers, and find the data by a binary search.

```
100 !QUICKFINDER by Jim Pete
rson
200 DIM D$(50):: GOTO 300 ::
 D$(),X :: !@P-
300 X=X+1 :: D$(X)="ALDA, AL
AN (999) 666-1234"
400 X=X+1 :: D$(X)="BUSH, GE
ORGE (111) 111-1111"
500 X=X+1 :: D$(X)="GHADDAFI
, OMAR (999) 456-1234567"
600 X=X+1 :: D$(X)="KHOMEINI
, AYATOLLAH  (666) 666-6666"
700 !@P+
800 INPUT "NAME? ":M$
900 IF M$>D$(X)THEN PRINT "N
OT FOUND":"CLOSEST IS":D$(X)
:: GOTO 800
1000 IF M$<D$(1)THEN PRINT "
NOT FOUND":"CLOSEST IS":D$(1
):: GOTO 800
1100 H=X :: S=INT(X/2)
1200 S$=D$(S):: IF POS(S$,M$
,1)=1 THEN 1700
1300 S$=D$(S+1):: IF POS(S$,
M$,1)=1 THEN S=S+1 :: GOTO 1
700
1400 IF S$>M$ THEN H=S :: S=
INT(H/2):: GOTO 1600
1500 S=S+INT((H-S)/2)
1600 IF S=S2 THEN 1800 ELSE
S2=S :: GOTO 1200
1700 PRINT D$(S):: GOTO 800
1800 PRINT "NOT FOUND":"CLOS
EST ARE"
1900 IF D$(S2)>M$ THEN PRINT
 D$(S2-1):D$(S2+1):: GOTO 80
0
2000 PRINT D$(S2+1):D$(S2+2)
:: GOTO 800
```

Note that in this case the records must be in alphabetical sequence. New records can be inserted in intermediate line numbers, in alphabetic sequence, always preceded by X=X+1 :: D$(X)= . Obsolete records can be deleted, and records can be corrected in place if the correction does not change the alphabetic sequence.

This idea did not work out as well as I hoped. The maximum number of records is less than 300, for the reason mentioned above, and this leaves so little free memory that even a binary search is slow. However, for a smaller file this is perhaps the best method.

For a large file, the best method is certainly a fixed sequential disk file, accessed by a binary search routine. But, that requires other routines to delete, add or change records, and had best be the subject of another Tips.

There is apparently a mistaken belief that sprites cannot be used together with my BXB routine. Not so - you can use all 28 of them! However, you cannot change their color with CALL COLOR(#,N). The only other limitations of BXB that I can think of, are that a single CALL COLOR cannot be used for multiple character sets and a single CALL CHAR can only reidentify one character. CALL CHARPAT cannot return the hex code of an ASCII above 143 because those ASCII's were not supposed to be available in Extended Basic.

I have used BXB on hundreds of Basic-only programs and have had only two rare problems. If the program contains multiple line feed colons ::::::, the computer may rearrange them into pairs of double colons :: :: and lock up. Or, if the colons are before the text, as in PRINT :"something" you may get a puzzling error message.

Also on rare occasions you might get an error message indicating the subprogram was called from a line containing a CALL CHAR, if the programmer had inadvertently put more than 16 characters in the hex code. Basic just ignores any extra characters, and XBasic uses them to reidentify the following ASCII, but BXB crashes.

From the T*I*M*E*S newsletter from England, here is an extremely useful bit of assembly which should be assembled as ALPHA/O and placed on the disk of every joystick program, or imbedded in it with ALSAVE.

```
       DEF   ALPHA
* save old R12
ALPHA   MOV   R12,@>FFFC
* 9900 CRU base=Ø
       CLR   R12
* signal alphalock key line
       SBZ   21
* check alphalock other side
       TB    7
* jump if state=on
       JNE   STATE
*  state=off
       SETO  @>FFFE
* as off skip next line
       JMP   JUMPA
*  state=on
STATE   CLR   @>FFFE
* stop sending to alpha key
JUMPA   SBO   21
* restore R12
       MOV   @>FFFC,R12
* standard XB return now
* clear error for basic
       SB    @>837C,@>837C
* return to calling program
       B     @>ØØ7Ø
       END   ALPHA
```

Now, put this in the first lines of the joystick program -

```
1 ! by M. Gikow, Andover
   MA  August 1988
2 ! used with ALPHA/O,
   will detect whether
   Alpha Lock is up (A=
   255) or down (A=Ø)
3 CALL CLEAR :: CALL INIT ::
 CALL LOAD("DSK1.ALPHA/O")
4 CALL LINK("ALPHA"):: CALL
PEEK(-1,A):: IF A=Ø THEN DIS
PLAY AT(12,1):"RELEASE ALPHA
LOCK" :: GOTO 4 ELSE CALL CL
EAR
```

I published this one in the C.O.N.N.I. newsletter. Barry Traver picked it up and put it in the TI Forum in Computer Shopper, but their typesetter garbled it, so here is how it was supposed to be -

According to the TI-Writer Reference Guide, page 77, when you select the PrintF command, then type C and space once and then the device name, any control characters with ASCII less than 32 are removed before the file is printed.

With Funlweb, at least, this is not quite true. A carriage return character, ASCII 13, or a line feed character, ASCII 1Ø, at the end of a line is actually not deleted but is changed to the space bar character, ASCII 32. This can be proved by running this little routine -

```
100 OPEN #1:"DSK1.(filename)
",INPUT
11Ø LINPUT #1:M$ :: PRINT M$
:LEN(M$):: IF LEN(M$)>Ø THEN
 PRINT ASC(SEG$(M$,LEN(M$),1
))
12Ø CALL KEY(Ø,K,S):: IF S=Ø
 THEN 12Ø ELSE 11Ø
```

Therefore, when a file is Filled/Adjusted and the line feed characters are stripped with the C option, the lines are one character longer than they appear to be. An apparently blank line also contains ASCII 32.

Since these characters are blank, they normally do no harm. However, they can create problems when records are read into programs for multiple column printing or concatenation of strings. In these cases, this routine can be used to strip out any ASCII below 33 at the ends of records.

```
1ØØ DATA INPUT,OUTPUT
11Ø FOR J=1 TO 2 :: READ J$
:: DISPLAY AT(12,1)ERASE ALL
:J$&" FILENAME?":"DSK" :: AC
CEPT AT(13,4):F$(J):: OPEN #
J:"DSK"&F$(J),UPDATE :: NEXT
 J
12Ø LINPUT #1:M$
13Ø IF ASC(SEG$(M$,LEN(M$),1
))<33 THEN M$=SEG$(M$,1,LEN(
M$)-1):: IF LEN(M$)>Ø THEN 1
3Ø
14Ø PRINT #2:M$ :: IF EOF(1)
<>1 THEN 12Ø :: CLOSE #1 ::
CLOSE #2
```

Attention all newsletter editors! If you are going to print my Tips (or anything else that contains program listings!) through the Formatter, PLEASE first replace and transliterate the ampersand, asterisk, period, carat and "@" sign!

Print this one through the Formatter and see why -

```
1ØØ A=A*264 :: @=1
11Ø PRINT "1 . . . 2 . . . 3
. . .4 . . . 5 . . . 6 . . .
7 . . . 8 . . . 9 . . . Ø"
12Ø M$=M$&A$&B$&C$ :: K=K^3
```

Here's how you do it. Load the above in the Editor, position the cursor at the beginning of the 1st line, hit FCTN 9, type RS and Enter, then /&/}/ and Enter. At the prompt, type A. Now get the cursor back to the beginning, repeat the above with /*/¦/, and then /./\/ and /^/~/ and /@/{/ and the file should now look like this -

```
1ØØ A=A¦264 :: {=1
11Ø PRINT "1 \ \ \ 2 \ \ \ 3
 \ \ \ 4 \ \ \ 5 \ \ \ 6 \ \
 \ 7 \ \ \ 8 \ \ \ 9 \
\ \ Ø"
12Ø M$=M$}A$}B$}C$ :: K=K~3
```

Now use FCTN 8 to open 5 lines at the top and add this transliteration -

```
.TL 92:46
.TL 123:64
.TL 124:42
.TL 125:38
.TL 126:94
```

Save the result, go to the Formatter and print it.

If my multi-column Printall program (Tips from the Tigercub #45) won't run on your Epson-compatible printer, try changing line 25Ø to -

```
25Ø ACCEPT AT(12,3)VALIDATE(
"123")SIZE(1):P :: IF P=2 TH
EN PRINT #1:CHR$(27);CHR$(77
)ELSE IF P=3 THEN PRINT #1:C
HR$(15)
```

You might also need to change the 136 in line 28Ø to 132.

If your printer offers the elite condensed option, you might want to add -
 :" (4) ELITE CONDENSED" to

(Continued on page 8)

IS THE TI-99/4A A CHILDS TOY OR A COMPETITOR IN THE PC WORLD

BY

RICHARD L. SCOTT

TAKEN FROM AN ARTICLE IN COMPUTER SHOPPER

BY BARRY TRAVER AND JONATHAN ZITTRAIN.

THOSE OF YOU WHO MAY OCCASIONALLY GET TOGETHER WITH GENERAL (NON-TI) COMPUTER TYPES MAY IDENTIFY WITH THE FOLLOWING EXPERIENCE. IN SUCH A SETTING (LIKE AN AREA COMPUTER SOCIETY WAS ONE SUCH SETTING). SOONER OR LATER THE QUESTION COMES. "AND WHAT COMPUTER DO YOU HAVE?" IF YOU RESPOND AT ONCE TO THE QUESTION, "A TI-99/4A," THE RESPONSE MAY BE GALES OF LAUGHTER ACCOMPANIED BY COMMENTS NOT APPROPRIATE TO BE WRITTEN HERE. SO STRATEGY HAS BEEN DEVISED TO MEET SUCH SITUATIONS.

RATHER THAN ANSWERING AT ONCE. I WOULD SAY SOMETHING LIKE THIS: "I'LL LET YOU GUESS. UNLIKE THE LOWLY APPLE II OR COMMADORE 64 (OR WHATEVER), IT HAS A 16-BIT MICROPROSSER. GIVING ME MAINFRAME-STYLE ASSEMBLY LANGUAGE WITH SOME RATHER NICE FEATURES, SUCH AS RELOCATABLE CODE AND MOVABLE SOFTWARE REGISTERS. I CAN PUT 16 COLORS ON THE SCREEN AT ONE TIME, AND THE AUTOMATED SPRITES WILL STAY IN MOTIONS WITH NO ATTENTION FROM THE CPU. FOR LANGUAGES, I ALSO HAVE BASIC (AT LEAST A DOZEN DIFFERENT VARIETIES, INCLUDING THE MAJOR EXTENDED BASICS), FORTRAN, FORTH (TWO MAJOR KINDS), LOGO (WWITH TURTLE, TILES, SPRITES AND MUSIC), PASCAL (UCDS AND TURBO), PILOT (THREE DIFFERENT VERSIONS, AND MORE (INCLUDING ASPIC AND A SMALL C). IN ADDITION TO RUNNING PROGRAMS USING SPEECH, I CAN RUN PROGRAMS USING SPEECH RECOGNITION..." AND SO ON.

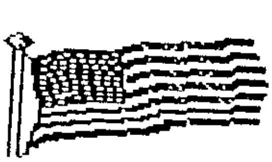
AS YOU CAN GUESS. THEIR GUESSES WERE RATHER COMICAL (ALMOST SENDING ME INTO GALES OF LAUGHTER!), BUT WHAT I ENJOYED MOST WAS THE LOOK ON THEIR FACES "NO, IT'S A TI-99/4A." YOU SEE, BY THAT POINT I HAD PROVEN THAT THE /4A WAS NOT JUST AN EDUCATIONAL TOY FOR CHILDREN SOLD BY A WELL KNOWN JELLO SALESMAN ON TV, BUT A SERIOUS COMPUTER WITH SERIOUS CAPABILITIES FOR ADULTS.

YOU CAN SEE THAT WE HAVE COME TO THE PLACE WHERE THE PENDALUM HAS SWUNG, SO MUCH SO THAT WE HAVE FORGOTTEN THAT THE /4A - INDEED A GOOD COMPUTER FOR ADULTS - IS STILL ONE OF THE BEST COMPUTERS AROUND FOR CHILDREN. THERE IS MUCH SOFTWARE AVILABLE FOR /4A ON MANY TELECOMMUNICATIONS NETWORKS BUT THERE IS NOT MUCH SOFTWARE IN THE EDUCATION AREA FOR CHILDREN. LOGO II IS AN EXCELLENT PROGRAM FOR CHILDREN EVEN BETTER THAN SOME OF THE PROGRAMS FOR APPLE. THERE HAS NOT BEEN ANY NEW PROGRAMS FOR LOGO IN A LONG TIME. SO THERE IS MUCH ROOM FOR IMPROVEMENT. AND THERE IS NEED FOR MORE SOFTWARE TO UTILIZE THE CAPABILIES OF THE TI-99/4A.

(Continued from page 7).

line 24Ø, change the VALIDATE string in 25Ø to "1234", add ELSE IF P=4 THEN PRINT #1:CHR$(27);CHR$(77);CHR$(15) to the revised line 25Ø and add +(P=4)*16Ø to the first statement in line 28Ø.

Memory almost full,
Jim Peterson



MENU SELECTION OF TYPE STYLES................by Ed Machonis

This 10 line TI BASIC program enables selection of any of the 128 type styles available on the Epson RX-80 printer. If line spacing and margin combinations are included, more than 1024 variations are available. It will also print a test line of print, showing the appearance of the selected style.

Selections should always start by pressing 1 for RESET to insure that previous selections are canceled. Printers that do not support a master reset should be turned off and then back on at this point.

Styles are combined by successive selections, i.e., COMPRESSED EXPANDED UNDERLINED DOUBLE STRIKE is obtained by selecting: 1 <ENTER> 4 <ENTER> 3 <ENTER> 8 <ENTER> 7 <ENTER>

The control codes are entered in LINE 10. CHR$(27), the ESCape code is obtained by pressing CONTROL and PERIOD at the same time. CHR$(15), turning on Compressed style, is obtained by pressing CONTROL AND O.(Not Zero)

Due to its short length, the program loads quickly and can be placed on the TI-WRITER and MULTIPLAN disks to enable selection of different type styles before printing.(Compressed Underlined is great for printing MULTIPLAN files, making 132 columns available on 8-1/2" paper.)

It can also be placed at the beginning of other programs which utilize a printer, where it will permit setting up the printer each time the program is run.

RX-80

```
1 DIM P$(15)

2 READ P$(1),P$(2),P$(3),P$(
4),P$(5),P$(6),P$(7),P$(8),P
$(9),P$(10),P$(11),P$(12),P$
(13),P$(14),P$(15)

3 OPEN #1:"PIO"

4 PRINT :"COMBINE STYLES BY
SUCCESSIVESELECTIONS- I.E. C
OMPRESSED EXPANDED UNDERLINE
D DOUBLE STRIKE=1-4-3-8-7"

5 PRINT ::"1 PICA/RESET","8
UNDERLINE","2 ELITE","9 TEST
","3 EXPANDED","10 EXIT","4
COMPRESSED","11 SUPERSCRIPT"

6 INPUT "5 EMPHASIZED  12 SU
BSCRIPT  6 ITALIC      13 1/
2 LINE SP7 D'BLE STRIK 14 R
MARGIN 6715 L MARGIN 13 ?":I

7 IF (I<1)+(I>15)THEN 5

8 PRINT #1:P$(I)

9 IF P$(I)<>"" THEN 5

10 DATA •@,•M,•W1,$,•E,•4,•6
,•-1,QUICK BROWN FOX JUMPS O
VER THE LAZY RED DOG 1234567
890 TIMES,,•S0,•S1,•1,•QC,•l
```

In LINE 10:
•=CONTROL PERIOD
$=CONTROL O (Not Zero.
The last character is a lower case L, NOT the figure 1.

NOTE: when program is listed to a printer, LINE 10 will not print properly and will send control codes to the printer.

When listed to screen, and when entering, a graphic symbol or a blank space will appear in place of the CONTROL character.

The program can be adapted to other printers by changing the OPEN statement in LINE 3 and the codes in LINES 8 & 10 as required. Refer to pg III-2 in TI's User's Reference Guide for the CONTROL KEY equivelants (Pascal Mode) of the printer's control codes. Appropriate changes should also be made in LINES 5 and 6. The sequence of the printer control codes in LINE 10 must match the numerical sequence of the style names. Note that EXIT is accomplished with a comma immediately following the comma after TIMES

Certain printers, such as the AXIOM, will not recognize CONTROL PERIOD as an escape code. For these printers the program must be modified to send the ESCAPE code as CHR$(27), etc.

The following program shows such a modification for the RX-80 printer. We have to give up the instruction display and the test for a valid input in order to hold the program down to 10 lines.

Please note the space immediately following the first quotation mark in Line 10. The space is important and the program will not work properly without it. (Can you tell why?)

PRINTSTYLE (For RX-80)

```
1 DIM P$(15)

2 READ P$(1),P$(2),P$(3),P$(
5),P$(6),P$(7),P$(8),P$(9),P
$(10),P$(11),P$(12),P$(13),P
$(14),P$(15)

3 OPEN #1:"PIO"

4 PRINT :"1 PICA/RESET","9 T
EST","2 ELITE","10 EXIT","3
EXPANDED","11 SUPERSCRIPT","
4 COMPRESSED","12 SUBSCRIPT"

5 INPUT "5 EMPHASIZED  13 1/
2 LINE SP6 ITALIC      14 L
MARGIN 137 D'BLE STRIK 15 R
MARGIN 678 UNDERLINE    ?":I

6 PRINT #1:CHR$(27)&P$(I)

7 IF I<>4 THEN 9

8 PRINT #1:CHR$(27)&CHR$(15)

9 IF I<>10 THEN 4

10 DATA @,M,W1,E,4,6,-1," QU
ICK BROWN FOX JUMPS OVER THE
 LAZY RED DOG 1234567890 TIM
ES",,S0,S1,l,1,QC
```

Note: P$(14), the next to last data item, is a lower case letter.L, not the figure 1.

Ogdens 99'er User Group
Mail Address: 1396 LINCOLN Ave. APT.B Ogden Utah 84404
GROUP OFFICERS
President: Jim Buck 773-2552
Vice President: Richard Morrison 621-3897
Secretary-Treasurer: Richard Scott 776-2551
Librarian: Harold Hilburn 773-0622
Asst. Librarian: Mel Bragg 393-9605
Newsletter Editor: Mel Bragg 393-9605
Associate Editor: Harold Hilburn 773-0622

JUNE 1989 NEWSLETTER

NEXT MEETING DATES

The next meetings dates are: JUNE 03 AND JUNE 20 1989 at 9:00 am on Saturday AND 7:00pm on Tuesday the 20th. We will be meeting in the CIVIL AIR PATROL building at the OGDEN MUNICIPAL AIRPORT, AIRPORT ROAD.

TI OGDEN USERS GROUP
1396 LINCLON AVE. APT. B
OGDEN, UTAH 84404