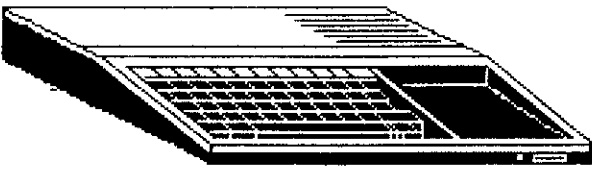


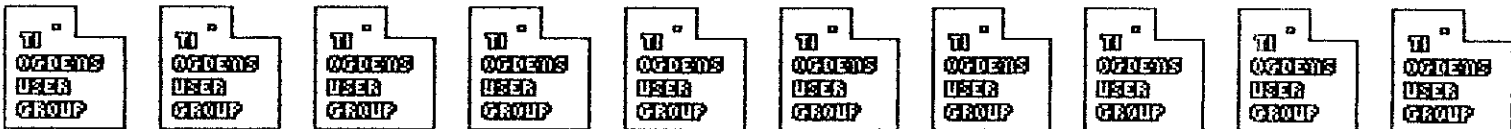
a publication of the ti 99-4a and 9640
occams users group inc.

TI 99-4A

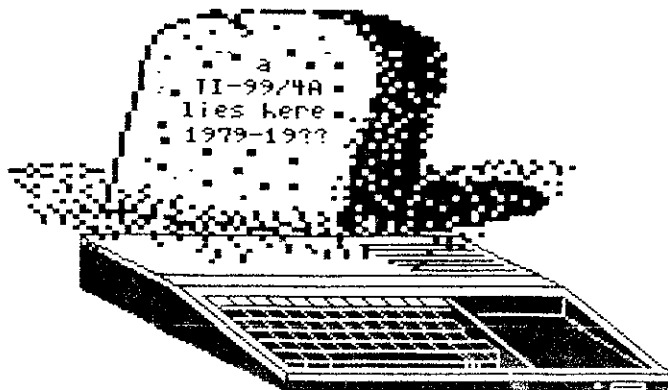


OCCAMS **GROUP**
USERS

**THE CLUB THAT REFUSES
TO SAY GOODBYE**



MAY 1989



**DON'T BURY YOUR
TI COMPUTER!
LEARN HOW TO USE
YOUR COMPUTER!
COME TO USERS
GROUP MEETINGS
IN MAY 89**

#56

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, deductible 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 auto-loader by full program name on each disk. These are available as a copying service for just \$1.50 post-paid 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.

In Tips #55, I showed you some quick and easy ways to create new character sets. Since folks nowadays don't like to key in long programs, let's continue with "tinygram" programming, and at the same time show you how to manipulate strings, and teach you the value of using MERGE format. First, let's make a screen to display our new characters. Some of them will have to be double-spaced horizontally or vertically, so -
100 CALL CLEAR :: X=1 :: FOR CH=48 TO 159 :: PRINT CHR\$(CH)&" " :: X=X+2 :: IF X<29 THEN 110 ELSE PRINT "":"": "" :: X=1
110 NEXT CH
Save it- SAVE DSK1.100,MERGE
Now, you might like to move the common punctuation marks into the same character sets as the characters, so that you will not have to reidentify so many sets, also so you can color them easier.
120 DATA 32,33,34,44,46
130 FOR J=1 TO 5 :: READ CH :: CALL CHARPAT(CH,CH\$):: CALL CHAR(J+90,CH\$):: CALL CHAR(J+122,CH\$)
140 NEXT J :: CALL CHARPAT(63,CH\$):: CALL CHAR(64,CH\$):: CALL CHAR(96,CH\$)
If you want to program in Basic, or use BXB with characters all the way up to ASCII 159, add CALL CHAR(J+54,CH\$) to the end of line 130 and CALL CHAR(128,CH\$) to the end of line 140.
Save by SAVE DSK1.120,MERGE
If you are using that translation, you must remember that with upper case characters the ? is @, space is [, ! is \, " is], comma

is , period is _ . With the lower case they are FCTN keys C, F, A, G, W and V and for the 3rd set (ASCII 129 to 154) they are CTRL comma, period,;,=,* and (.
You can transfer upper case to lower by -
CALL CHARPAT(CH,CH\$) and then CALL CHAR(CH+32,CH\$) or the opposite by CH-32 and if you have BXB merged in you can create a 3rd set by CH+64.
The following are all incompatible with each other, so give them all line number 150 and save them in merge format as 150A, 150B, etc.
The numerals and the upper case letters all have the topmost pixel row blank to provide spacing between lines of text. We can make taller letters by deleting the top row and doubling the 7th row -
150 FOR CH=48 TO 126 :: CALL CHARPAT(CH,CH\$):: CALL CHAR(CH,SEG\$(CH\$,3,12)&SEG\$(CH\$,13,4)):: NEXT CH
151 REM
Or, you can double the 3rd row -
150 FOR CH=48 TO 95 :: CALL CHARPAT(CH,CH\$):: CALL CHAR(CH,SEG\$(CH\$,3,4)&SEG\$(CH\$,5,12)):: NEXT CH
151 REM
The lower case letters are really small upper case with the upper 3 rows blank. All their vertical bars are in the 4th, 6th and 8th rows, so let's drop the first 3 rows and quadruple the 7th.
150 FOR CH=97 TO 127 :: CALL CHARPAT(CH,CH\$):: CALL CHAR(CH,SEG\$(CH\$,7,6)&RPT\$(SEG\$(CH\$,13,2),4)&SEG\$(CH\$,15,2)):: NEXT CH
151 REM

Or, for topheavy letters, quadruple the 5th row -
150 FOR CH=97 TO 127 :: CALL CHARPAT(CH,CH\$):: CALL CHAR(CH,SEG\$(CH\$,7,2)&RPT\$(SEG\$(CH\$,9,2),4)&SEG\$(CH\$,11,6)):: NEXT CH
151 REM
Or, if you want line spacing -
150 FOR CH=97 TO 122 :: CALL CHARPAT(CH,CH\$):: CH=SEG\$(CH\$,5,8)&RPT\$(SEG\$(CH\$,13,2),3)&SEG\$(CH\$,15,2):: CALL CHAR(CH,CH\$):: NEXT CH
151 REM
Or, for something silly -
150 FOR CH=48 TO 90 :: CALL CHARPAT(CH,CH\$):: CALL CHAR(CH,SEG\$(CH\$,3,2)&RPT\$(SEG\$(CH\$,5,2),4)&SEG\$(CH\$,9,4)&SEG\$(CH\$,15,2)):: NEXT CH
151 REM
For some good blocky characters -
150 FOR CH=48 TO 90 :: CALL CHARPAT(CH,CH\$):: CALL CHAR(CH,RPT\$(SEG\$(CH\$,3,2),2)&SEG\$(CH\$,5,8)&RPT\$(SEG\$(CH\$,15,2),2)):: NEXT CH
151 REM
Or, if you would prefer them shorter for single-line spacing -
150 FOR CH=48 TO 90 :: CALL CHARPAT(CH,CH\$):: CALL CHAR(CH,"000"&RPT\$(SEG\$(CH\$,3,2),2)&SEG\$(CH\$,7,6)&RPT\$(SEG\$(CH\$,15,2),2)):: NEXT CH
151 REM
If you would like numerals the same size as lower case,
150 FOR CH=48 TO 57 :: CALL CHARPAT(CH,CH\$):: CALL CHAR(CH,"000"&SEG\$(CH\$,1,6)&SEG\$(CH\$,9,4)&SEG\$(CH\$,15,2)):: NEXT CH

```

151 REM      : Add a line 500 GOTO 500      : &"60" :: GOTO 163      :
      : And start MERGEing in your      : 162 CH2%=CH2%&SEG$(CH$,J-1,1) : 160 FOR CH=48 TO 122 :: CALL
      You can even shrink the      : series of "150" routines and      : )&SEG$( "0367CBEF",POS("0246B      : CHARPAT(CH,CH%)
      lower case to only 4 rows      : running them to see what you      : ACE",SEG$(CH$,J,1),1),1)      : 161 FOR J=1 TO 15 STEP 2 ::
      high, although some letters      : have created.                      : 163 NEXT J :: CALL CHAR(CH,C      : CH2%=CH2%&SEG$(CH$,J,1)&SEG$
      are not very legible -        : Then, save these next             : H2%): CH2%="" :: NEXT CH      : ("0CBC",POS("048C",SEG$(CH$,
      : routines in MERGE format as      : J+1,1),1),1):: NEXT J
150 FOR CH=97 TO 122 :: CALL      : 160A, 160B, etc.                : Or shade them both left      : 162 CALL CHAR(CH,CH2%): CH2
      CHARPAT(CH,CH%): CALL CHAR      : All normal characters have      : and right -                    : $="" :: NEXT CH
      (CH,SEG$(CH$,1,6)&SEG$(CH$,5      : the leftmost column of          :
      ,4)&SEG$(CH$,11,6)): NEXT C      : pixels and the two right-        :
      H                                : most columns blank, for         : 160 FOR CH=48 TO 122 :: CALL      :
151 REM      : spacing between letters. We      : CHARPAT(CH,CH%): FOR J=1 T      : Or inward at right, out-
      Something modernistic -        : can widen the character into      : 0 15 STEP 2 :: A%=SEG$(CH$,J      : ward at left -
      : the left column -                : ,1):: P=POS("0123456789ABCDE      :
      : F",A$,1)                          : 160 FOR CH=48 TO 122 :: CALL      :
150 A%="" :: FOR CH=48 TO          : 161 A%=SEG$( "0367CDEF89ABCDE      : CHARPAT(CH,CH%): FOR J=1 T
      90 :: CALL CHARPAT(CH,CH%):      : F",P,1): B%=SEG$(CH$,J+1,1)      : 0 15 STEP 2
      CALL CHAR(CH,SEG$(CH$,1,4)&      : CHARPAT(CH,CH%): FOR J=1 T      : :: P=POS("02468ACE",B$,1)::
      A%&SEG$(CH$,7,6)&A%&SEG$(CH$      : 0 15 STEP 2                      : B%=SEG$( "0367CBEF",P,1):: CH
      ,15,2)): NEXT CH              : 161 CH2%=CH2%&SEG$( "014589CD      : 2%=CH2%&A%&B%
151 REM      : ",POS("01234567",SEG$(CH$,J,      : 162 NEXT J :: CALL CHAR(CH,C      :
      Or perhaps even better -        : 1),1),1)&SEG$(CH$,J+1,1):: N      : H2%): CH2%="" :: NEXT CH
      : EXT J :: CALL CHAR(CH,CH2%):      : 163 CALL CHAR(74,"000C0C0C0C      :
150 A%="" :: FOR CH=48 TO          : : CH2%="" :: NEXT CH            : 0C4C38"): CALL CHAR(106,"00      :
      90 :: CALL CHARPAT(CH,CH%):      : 162 REM                          : 00000C0C0C4C38")
      CH%=SEG$(CH$,3,10)&RPT$(SEG      : 163 REM                          :
      $(CH$,13,2),2)&SEG$(CH$,15,2      :
      )                                : Or shaded into both of the
151 CALL CHAR(CH,SEG$(CH$,1,        : Or widen it both left and        : rightmost columns -
      4)&A%&SEG$(CH$,7,2)&A%&SEG$(      : right -
      CH$,11,2)&A%&SEG$(CH$,15,2)      :
      :: NEXT CH                    : 160 FOR CH=48 TO 122 :: CALL
      : CHARPAT(CH,CH%): FOR J=1 T      : CHARPAT(CH,CH%): FOR J=2 T
      : 0 15 STEP 2                      : 0 16 STEP 2 :: CH2%=CH2%&SEG
      : $(CH$,J-1,1)&SEG$( "0377EBFF      : ",POS("01234567",SEG$(CH$,J,
      ,POS("02468ACE",SEG$(CH$,J,1      : ),1),1):: NEXT J :: CALL CHA
      ),1),1):: NEXT J :: CALL CHA      : R(CH,CH2%): CH2%="" :: NEXT
      : CH                                  : CH
150 FOR CH=48 TO 122 :: CALL        : 161 CH2%=CH2%&SEG$( "014589CD      :
      CHARPAT(CH,CH%): CH%=SEG$(      : ",POS("01234567",SEG$(CH$,J,      :
      CH$,3,14)&SEG$(CH$,1,2):: X$      : 1),1),1)&SEG$( "028A",POS("04
      =SEG$(CH$,1,1)&"0"              : 8C",SEG$(CH$,J+1,1),1),1)
151 FOR J=3 TO 15 STEP 2 ::          : 162 NEXT J :: CALL CHAR(CH,C      :
      X%=X%&SEG$(CH$,J,1)&SEG$(CH$      : H2%): CH2%="" :: NEXT CH
      ,J-1,1):: NEXT J :: CALL CHA      : 161 REM
      R(CH,X%): X%="" :: NEXT CH      : 162 REM
      : 163 REM                          :
      : Or even a full 8 columns          : Or into all 8 columns -
      : wide by just changing the         : Or shaded into both of the
      : "028A" in line 161 to "0129"      : rightmost columns -
      :
      : For darker characters, we        : Here's a weirdo -
      : can shade them into the 7th      :
      : column -                          : 160 FOR CH=48 TO 122 :: CALL
150 FOR CH=48 TO 122 :: CALL        : CHARPAT(CH,CH%): FOR J=1 T      : CHARPAT(CH,CH%): FOR J=9 T
      CHARPAT(CH,CH%): FOR J=1 T      : 0 15 STEP 2                      : 0 15 STEP 2
      0 15 STEP 2 :: CH2%=CH2%&SEG      : 0 16 STEP 2 :: CH2%=CH2%&SEG
      $(CH$,J,1)&SEG$(CH$,J+3,1)::      : $(CH$,J-1,1)&SEG$( "0377EBFF      : ",POS("01234567",SEG$(CH$,J,
      NEXT J :: CALL CHAR(CH,CH2%      : ),1),1):: NEXT J :: CALL CHA
      ): CH2%="" :: NEXT CH          : R(CH,CH2%): CH2%="" :: NEXT
151 REM      : CH                                  : CH
      : 161 REM                          : 161 REM
      : 162 REM                          : 162 REM
      : 163 REM                          : 163 REM
      :
      : Try changing that to FOR J
      : =1 TO 7 and CALL CHAR(CH,CH2
      : %&SEG$(CH$,9,8))
      :
      : And one more -
      :
      : 160 FOR CH=48 TO 122 :: CALL
      : CHARPAT(CH,CH%): FOR J=1 T
      : 0 15 STEP 2 :: P=POS("012345
      : 6789ABCDEF",SEG$(CH$,J,1),1)
      : 161 A%=SEG$( "0367CDEF89ABCDE
      : F",P,1): P=POS("02468ACE",S
      : EG$(CH$,J+1,1),1): B%=SEG$(
      : "0367EBFF",P,1):: CH2%=CH2%&
      : A%&B%
      : 162 NEXT J :: CALL CHAR(CH,C
      : H2%): CH2%="" :: NEXT CH
      : 163 REM
      :
      : 161 IF CH=67 OR CH=71 OR CH=
      : 99 OR CH=103 THEN 162 :: IF
      : SEG$(CH$,J-1,1)="4" AND SEG$
      : (CH$,J,1)="0" THEN CH2%=CH2%
      : at right - 3
      : 163 REM

```

MERGE in 100 and 120, put in ; after sets you can make!
a holding line 500 GOTO 500 ; Memory full,
and start MERGEing in all of ; Jim Peterson
the different combinations ;
of the 150 and 160 lines and ;

A SURPRISING DISCOVERY WITH TI'S PROGRAMMING AIDS III

TI'S Programming Aids III opens the door to some powerful programming techniques. The *EDITOR* capabilities of this software will be appreciated by the serious extended *BASIC* programmer. But the excitement really begins when you realize how this software does its thing.

The *EDITOR* can provide the ability to delete, move, or resequence specified groups of lines within a program much more quickly than could be done manually at the keyboard.

Required Hardware:

EDITOR comes on the *Programming Aids III* disk which is a set of four Extended *BASIC* programs (*LINPUT*, *CREF*, *CREFPRINT* and *EDITOR*). In addition to a disk controller, disk drive, and the Extended *BASIC* Command Cartridge, a printer is a practical necessity, there is no provision for screen display of the output from the Cross Reference procedure.

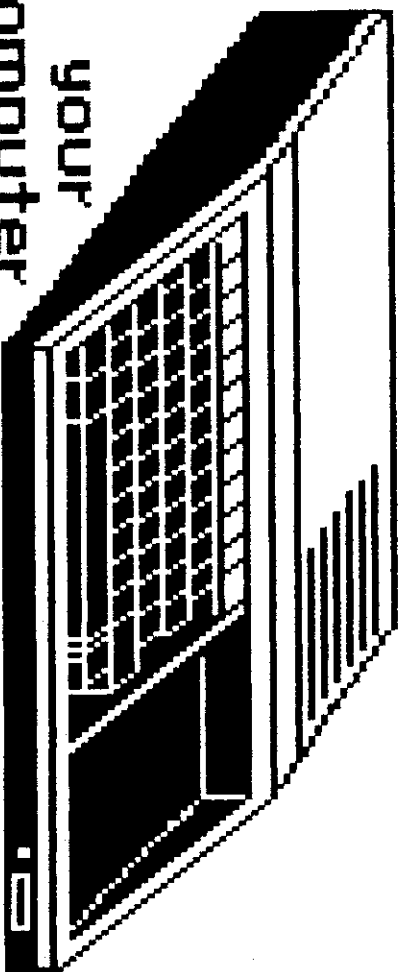
EDITOR:

The *EDITOR* program makes possible any desired modification of line numbers in a *BASIC* or Extended *BASIC* program. The only way to resequence a program was to use the *RESEQUENCE (RES)* command, which effects all line numbers within a program. *EDITOR* allows one to resequence specified sections of a program without affecting others. If you have numbered subroutine statements in a manner which is easy to remember (1000, 2000, 3000, etc), you can retain this number and "open up" a previous part of the program for insertion of additional lines. An even more useful application would be the rearrangement of sections of *BASIC* code. Suppose you want to merge several programs, each of which contains subroutines. Without *EDITOR*, you would be faced with the time consuming job of moving all subroutines to the end of the merged program. With *EDITOR*, this procedure can be done very simply and quickly by renumbering all subroutine lines. Finally, the *EDITOR* program allows deletion of sections of *BASIC* code. If you want to get a subroutine out of one program to use in another, it's no problem.

How EDITOR Works:

The program to be edited is first saved with the *MERGE* option, and then the *EDITOR* program is loaded and run. Upon entry of the "OLD" command provided, *EDITOR* inputs each record in the condensed format file and constructs the line number from the *ASCII* codes of the first two bytes. Program line numbers thus obtained are stored in an array, with array position corresponding to record number. After the user has altered these numbers using the *DELETE (DEL)* and *RESEQUENCE (RES)* commands provided, the *SAVE* command initiates the process in which altered numbers are reassigned to records in the file. As each record is read a second time, the corresponding line number in the array is translated into two *ASCII* characters which are substituted for those on the record, and the new record is written to a new file (after making the necessary changes to any line references). At the end of this process, the end-of-file mark is written as the record on the file. After initializing program memory with the *NEW* command, all you need to do is load the new file with the *MERGE* command. The program will then be

Do you own a computer
like this one?



Take your
TI Computer
out of the closet and join the
Ogden TI user group. **COMPUTER
CLUB!**

You can use your TI- 99 / 4A

for:

- Business
- Word Processing
- Games
- Programming
- Home Budget
- Graphics
- and More!

FOR INFORMATION, HELP ETC. **MEMBERSHIP**
CALL THE CLUB PUBLICITY GUYS **\$ 10**
AT 393-9605 OR 773-2552

per year

A member of the
Ogden TI user
Group, will receive
all of the following:

- A YEARLY SUBSCRIPTION
to the clubs informative
monthly newsletter.
 - Access to our clubs
honorarium library of
public domain programs
 - A permanent invitation
to our clubs biweekly
meetings, which are
occasionally interesting
and informative.
 - A Tape or Disk with
complimentary programs on.
CALL THE NUMBERS AT THE LEFT
FOR INFORMATION ON MEETINGS
- We will be teaching BASIC
if people are willing
to come out!

TI-BASE - From INSCEB01
TUTORIAL 7.1 By Martin Smoley
NorthCoast 99'ers - Feb. 12, 1989
Copyright 1989 By Martin A. Smoley

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.

***** Miscellaneous Information *****

This month I'm going to try to answer some questions and include some new information from the new TI-Base manual. The first thing I'd like to say is "read the manual several times". There is a lot of information in it that you will probably miss the first or even the second time you read it, "I know I did". Make the two readings at least a day apart. If you read it twice in a row, you'll read in mistakes the first time through and rather than find the mistakes you'll reinforce those mistakes the second time through. It is very apparent from letters I receive that some people try to start using TI-Base after just skimming through the manual. TI-Base is too complicated for you to bluff your way through. I have been working with it for many months and I still keep the manual close at hand, and refer to it or the new QUICK REFERENCE GUIDE constantly. As far as programming problems go, typo's are my biggest. If you enter LOCAL BLNK C 30, and later in the program you try REPLACE BLANK WITH " Good morning", TI-Base will not match BLNK with BLANK. The next big mistake I make is to try and concatenate too many characters together. I then try to jam them into a variable with too little space. If BLNK has 30 spaces available you must add up the characters you have in the fields you want to stuff into BLNK. It's easy to say REPLACE BLNK WITH FN ; LN. FN and LN don't look like much but they may represent 25 characters each, for a total of 50. You must keep these things in mind or written down, the system will not give you any help with these problems. One important thing I try to remember is "don't try to write a big program in one chunk". You should be creating a library of small CFs or programs. Each of these programs should do a fairly specific job, like printing labels or writing screen messages. Work on an individual CF until it works well and then use a bunch of these small programs together to do a larger job. If you are having problems with a CF, or a group of CFs you are trying to use, leave the TALK ON. All the lines will scroll up the screen as you watch. If TI-Base cannot execute a line for some reason, it will place an asterisk at the very beginning of the line for you to see. If you can't get the hang of this method, Version 2.01 has a TRACE command. It is simply TRACE ON/TRACE OFF. If you enter TRACE ON, all lines which TI-Base can execute will be sent to your printer. You can then compare that printout to your original CF and the lines that didn't print out did not execute. When you find lines that won't execute, check them for variables first. See if you are trying to put too much data into a variable, or possibly the data is the wrong type. With version 2.0 and up I can't seem to mix those C)character and N)umeric fields the way I could with 1.02.

I hate to admit it but on several occasions I found that I was trying to put things into a variable that I had not created or initialized at the beginning of the CF. If you are attempting to use a command or function for the first time and you do not understand how to phrase the statement, work small. Create a very small CF that does little more than test your new function. If you need to use a database with this function, use one with only a few names in it. "I use Tnames, it contains only six names." That speeds up the test time and allows me to try the maximum amount of variations in the least amount of time. When you are having problems with a CF and you'd like a hard copy to look at, there is a new command called LIST. If you type LIST DSK1.filename/C at the dot prompt, TIB will go to the disk number you entered and try to find the filename you want, remember the /C is needed. If it can find that filename, it will LIST it to your printer. I find this command and SNAP, which is the new TIB screen dump, very helpful. I seem to be able to find many small errors on a printer page that I overlook on the monitor.

***** Printer Controls *****

Since I am already talking about printing hardcopies, I'd like to throw in some printer information. TIB Version 2.01 has built in printer control codes. They are present in the form of a database on the PRGDISK, named PRINTER. If you LIST or EDIT the SETUP CF that comes with TIB you will see * PRINTER EPSON in that file. Because there is an asterisk at the beginning of that line, TIB ignores the command and none of those control codes are available to you. That's because of the wide variety of printers on the market today. You must set up your own printer control codes in this system and make them available for TIBs access. Here's a little help on approaching this utility. We will look at the EPSON section because I am familiar with EPSON printers. Type USE DSK1.PRINTER <E>. The name PRINTER should come up in the bottom line across your screen as a normal database in use. Next type DISPLAY STRUCTURE. Now, with your printer turned on, type SNAP <E> and hopefully the screen will be dumped to your printer. It should look like the screen 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	DS	X	004	
6	UL	X	006	
7	EX	X	002	
8	CM	X	004	
9	IT	X	004	
10	B	X	004	
11	SPS	X	006	
12	SBS	X	006	
13	HT	X	002	
14	ST	X	020	
15	NM	X	026	
16	BLANK	X	030	

. SNAP

000 1 PRINTER 00006/00007

Continued Next Page.

**TI-BASE - From INSCEBOT
TUTORIAL 7.2 By Martin Smoley
NorthCoast 99'ers - Feb. 17, 1989
Copyright 1989 By Martin A. Smoley**

You will notice that we are looking at a database structure, even though it is a little abnormal. It's abnormalities are what make it so useful. TIB recognizes this database as the place where it will find the control codes for your printer. The first field must be NAME. In the NAME field you would store your printer name, such as Epson. The next three fields you should recognize as Form Feed, Line Feed, and Carriage Return. These must stay exactly as you see them for TIB to work properly. The rest of the fields are Double Strike, Under Line, Enlarged Print, Condensed Mode, Italics, Bold, SuPerScript, SuBScript, Horizontal Tab, Set Tab, Normal Mode (NM cancels all of the previous commands), and a Blank line 30 characters in length. One item of great importance is the X type field designation. The X type tells TIB that the item stored in this field is specifically a printer control code. It also tells TIB that it should interpret the normal numbers and characters you type into the X field as Hexidecimal. If you USE DSK1.PRINTER and type EDIT, you will see the screen below. DIABO, "I assume for Diablo", is the first screen you will encounter. Pressing FCTN 5 will move you to the EPSON screen.

EDIT

```

NAME      EPSON
FF        OC
LF        OA
CR        OD
DS        1B47
UL        1B2D01
EX        OE
CM        000F
IT        1B34
B         1B45
SPS       1B5300
SBS       1B5301
HT        09
ST        1B440A0A0A0A0A0A0A00
NM        1B481B2D3014121B351B461B5>
BLANK    000000000000000000000000>

```

000 1 PRINTER 00000/00007

Any of these screens can be edited to your needs. You can change the names to your own liking. For example you can change EX to ENL, for ENLARGED. You can also change the data in the fields to match the codes your printer expects, if they are not already set properly. This part is a little confusing, but not really hard. NOTE: If you change the length of a data field the data will be lost and must be re-entered. Make sure you understand what you are doing before you change any lengths. Changing the data and thus changing the control code is as simple as typing over what is currently in a particular field. Lets look at DS, which stands for Double Strike for Epson or Bold on the Star MX-10. The printer command for DS or Bold is <ESC> "6". What you see in the field is 1B47. If you wish to change this, you would merely type over it with normal characters and numbers. TIB, however, will interpret these to be Hexadecimal numbers.

The code for <ESC>, escape, is decimal 27, and in <Hex> or Hexadecimal, it is 1B. So the first two characters in the DS field stand for Escape (1B). The second half of the command is "6" or 6. The decimal for 6 is 71 and the Hex for 6 is 47, so 1B47 is Hex for <ESC> 6. In every printer handbook I can remember seeing, the ASCII, the Decimal and the Hexadecimal are given for the printer commands. You may have to search through your printer manual, but you don't have to figure these out, just look them up and type them in. One more example would be B for Bold, which actually produces Emphasized on my MX-10. My book says <ESC> "E" turns on Emphasized. Just below that it says 27 69 and below that it says 1B 45. If you check the EDIT screen, you'll see that the B field contains 1B45. OK, how do you use this stuff. Well in most cases it's easier and more convenient than you might think. First EDIT the SETUP file. Remove the asterisk in front of the PRINTER EPSON. If you have an Epson, beam! Star or a similar printer, try the FCTN 8 command first. After using FCTN 8 to save the new CF, type DO DSK1.SETUP to rerun the SETUP file. As SETUP runs watch the bottom of your screen and you will see PRINTER come up while the EPSON commands are being installed. After SETUP finishes all traces of this operation will disappear, but the results will still be usable. Open the Tnames DB by typing USE Tnames. With your printer turned on type PRINT (B) <E>. If everything worked, you just set your printer to TIB Bold. Type PRINT ALL FN,MI,LM <E>. This should print out ALL of the names in Tnames, in Bold Faced print. Typing PRINT (NM) <E> should return you to normal print, or DRAFT MODE. Try PRINT ALL (B) FN (CM) MI (NM) (IT) LM (NM) <E>. This should PRINT ALL the First Names in Bold, the Middle Initials in Condensed and the Last Names in Italics. Notice that the printer controls are enclosed in parenthesis and that a delimiter (comma or blank space) is needed between the fields or control codes. The <E> represents Press Enter. This type of control code use is very helpful when you are dumping a complete database to the printer. You can type PRINT (CM) <E> and then PRINT ALL <E>. With TIB version 2.01 that troublesome line crawling that produced a printout that looked like a printer test pattern has been eliminated. Condensed mode now produces a nice neat columnar type page. I hope this is enough to get you started. Remember, read that manual. There is more on printer controls in the manual and I'll probably slip some of them into the tutorials in the future.

*** Geneve Myarc ***

Here's an important tip from Bob Stevens of Battle Creek, MI. Bob says he had some disasterous results on his Geneve with a Myarc Disk Controller when he tried to run CFs off the PRGDISK. The cure he found was to initialize some new disks using 16 rather than 18 sectors and reset the interlace to 10. He then copied his TI-Base files to the new disks and lived happily ever after. Thanks for sharing that with us Bob, it will probably save someone in Florida a lot of aggravation.

*** THE LIME FAIR ***

This is a little early but I'd like to start plugging the Lima Fair. Quite a few people from the Cleveland area will be going to the Lima Fair this year. The fair will be held on May 20, and I hope to give a Demonstration of TI-Base at that time. I am also looking forward to meeting some of the people I have corresponded with and talked to over the phone.

Continued Next Month.

(Editors note: Thanks to Burr Settles of the Bluegrass 99er Computer Society, Inc. Jan 1988).

KIDKEYS

Since we are at the beginning of a new year, we're going to start it off with a bang and announce a brand new column. KIDKEYS is TI-FYI (For Your Information) especially for the younger set. Since many TI's were initially bought to contribute to the technological information being taught in our schools, lots of kids out there (ours and others) are keying in information on TI's everywhere. Many of these midget programmers are doing things still only imagined by this simple user. Please feel free to contribute your efforts on behalf of your children, or the efforts of your children themselves.

This month's issue features a program written for a third-grade school project by Burr Settles. Burr had just turned nine when he began work on the program, which features graphics, music, and a menu that takes you between screens that offer a variety of information on solitary or social bees. Although some help was required by his father Bill (who ran as fast as he could to Club expert Wes Richardson) in order to make the graphics appear to move, Burr demonstrated his understanding of the hexadecimal system of graphics design with his incorporation of a beehive and bees, and his comprehension of CALL SOUND routines by adding a current piano assignment to the program, Beethoven's Ode to Joy. Burr will be demonstrating this program at the January meeting.

B
E
E
S



By Burr Settles

```

100 REM *****
110 REM **BEES**
120 REM *****
130 REM BY BURR H. SETTLES
140 REM COPYRIGHT 1987
150 REM BLUEGRASS 99 COMPUTER SOCIETY,
    INC.
160 CALL CLEAR
170 CALL SCREEN(10)
180 PRINT "    BURR'S BEE BOOK": : :
    : : : : :

```

```

190 PRINT "    Solitary vs. Social Bees"
    : : : : :
200 PRINT "    PROGRAMMER: BURR SETTLES"
210 GOSUB 1310
220 GOSUB 780
230 CALL CLEAR
240 CALL SCREEN(14)
250 PRINT "    READ THE CATAGORIES BELOW
    ": : : : :
260 PRINT "        1    Solitary Bee"
270 PRINT "            Characteristics":
280 PRINT
290 PRINT "        2    Social Bees"
300 PRINT "            Characteristics":
310 PRINT
320 PRINT "        3    Facts & Figures": :
330 PRINT
340 PRINT "        4    End Program": : : :
350 PRINT "            WHICH OPTION (?)"
360 CALL KEY(0,KEY,S)
370 IF KEY<49 THEN 360
380 IF KEY>52 THEN 360
390 IF S<1 THEN 360
400 IF KEY=49 THEN 440
410 IF KEY=50 THEN 530
420 IF KEY=51 THEN 620
430 IF KEY=52 THEN 720
440 CALL CLEAR
450 CALL SCREEN(5)
460 PRINT "            SOLITARY BEES": :
470 PRINT "            Solitary means being
            alone. There are many kinds of So
            litary Bees: Mining    Bees, Mason
            Bees, Wool Hang-"
480 PRINT "            er Bees, Carpenter Bees, an
            d Leaf-Cutter Bees. They may nest
            close together, but    they always
            live alone.":
490 PRINT "            Since they live apart,
            males put out a scent to    attra
            ct females to mate.    Solitary Be
            es can mate many"
500 PRINT "            times. Each female then
            builds the nest and cares    for t
            he eggs alone.": : : :
510 INPUT "PRESS ENTER TO RETURN TO
            MENU":A$
520 GOTO 230
530 CALL CLEAR
540 CALL SCREEN(16)
550 PRINT "            SOCIAL BEES": : :
560 PRINT "            Social bees live to-
            gether in colonies called    hives
            . Each hive is made up of three cl
            asses of bees to"
570 PRINT "            do the work. The Queen lay
            s the eggs. The worker gath-    ers p

```


BEES

```
ollen and nectar to      feed the be
bias with each"
580 PRINT "day. The drone's only job
   is to mate once with the Queen
   , then he dies."
590 PRINT "      The nests are man-made
   hives, or hollow logs which are n
   earby. There are also ground-nest
   ing Bumblebees.": : :
600 INPUT "PRESS ENTER TO RETURN TO
   MENU":A$
610 GOTO 230
620 CALL CLEAR
630 CALL SCREEN(4)
640 PRINT "      FACTS & FIGURES": :
650 PRINT "      There are over 10,000
   types of Solitary Bees, and only
   two kinds of social"
660 PRINT "bees. But there are over
   200 kinds of Bumblebees."
670 PRINT "      When the temperature i
   s above 50 degrees bees are activ
   e. They are paralyzed below 50 de
   grees. There are!"
680 PRINT "thirty to forty thousand
   bees in an average hive. The c
   olor of honey depends"
690 PRINT "on the source of nectar.
   The darker the color of the honey
   , the stronger the flavor. Ea
   ch hive will give about 100 lbs. of
   honey.": :
700 INPUT "PRESS ENTER TO RETURN TO
   MENU":A$
710 GOTO 230
720 CALL CLEAR
730 CALL SCREEN(9)
740 CALL SOUND(2000,-3,1)
750 PRINT "      BURR BUZZING OFF!!": :
   :
760 PRINT "      HAVE A HONEY OF A DAY!!":
   : : : : :
770 END
780 CALL CLEAR
790 CALL SCREEN(11)
800 CALL CHAR(128,"0000771C3E382000")
810 CALL CHAR(129,"084828181C3E3820")
820 CALL CHAR(130,"0000EE387C1C0400")
830 CALL CHAR(131,"10121418387C1C04")
840 CALL CHAR(132,"001F3F7FFFFFFF")
850 CALL CHAR(133,"00F8FCFEFFFFFF")
860 CALL CHAR(134,"00FFFFFFF")
870 CALL CHAR(135,"0080C0E0F0F0F0F0")
880 CALL CHAR(136,"000103070F0F0F0F")
890 CALL CHAR(137,"F0E0C08000000000")
900 CALL CHAR(138,"0F07030100000000")
910 CALL HCHAR(12,16,132)
920 CALL HCHAR(12,17,133)
930 CALL HCHAR(13,15,132)
940 CALL HCHAR(13,16,134,2)
950 CALL HCHAR(13,18,133)
960 CALL HCHAR(14,14,132)
970 CALL HCHAR(14,15,134,4)
980 CALL HCHAR(14,19,133)
990 CALL HCHAR(15,13,136)
1000 CALL HCHAR(15,14,134,6)
1010 CALL HCHAR(15,20,135)
1020 CALL HCHAR(16,13,132)
1030 CALL HCHAR(16,14,134,6)
1040 CALL HCHAR(16,20,133)
1050 CALL HCHAR(17,13,132)
1060 CALL HCHAR(17,14,134,6)
1070 CALL HCHAR(17,20,133)
1080 CALL HCHAR(18,13,132)
1090 CALL HCHAR(18,14,134,2)
1100 CALL HCHAR(18,16,137)
1110 CALL HCHAR(18,17,138)
1120 CALL HCHAR(18,18,134,2)
1130 CALL HCHAR(18,20,133)
1140 CALL HCHAR(19,13,132)
1150 CALL HCHAR(19,14,134,2)
1160 CALL HCHAR(19,18,134,2)
1170 CALL HCHAR(19,20,133)
1180 CALL HCHAR(20,13,132)
1190 CALL HCHAR(20,14,134,6)
1200 CALL HCHAR(20,20,133)
1210 PRINT "      PRESS ANY KEY TO CONTINUE
   "
1220 RANDOMIZE
1230 CALL SOUND(110,-3,0)
1240 ROW=INT(21*RND)+1
1250 COLUMN=INT(32*RND)+1
1260 CHAR=INT(4*RND)+128
1270 CALL VCHAR(ROW,COLUMN,CHAR)
1280 CALL KEY(0,KEY,R)
1290 IF R=0 THEN 1220
1300 IF R=1 THEN 230
1310 CALL SOUND(500,494,3)
1320 CALL SOUND(500,494,3)
1330 CALL SOUND(500,523,3)
1340 CALL SOUND(500,578,3)
1350 CALL SOUND(500,578,3)
1360 CALL SOUND(500,523,3)
1370 CALL SOUND(500,494,3)
1380 CALL SOUND(500,440,3)
1390 CALL SOUND(500,392,3)
1400 CALL SOUND(500,392,3)
1410 CALL SOUND(500,440,3)
1420 CALL SOUND(500,494,3)
1430 CALL SOUND(750,440,3)
1440 CALL SOUND(250,392,3)
1450 CALL SOUND(250,392,3)
1460 GOTO 220
```



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

MAY 1989 NEWSLETTER
NEXT MEETING DATES

The next meetings dates are: MAY 06 AND MAY 16 1989 at 9:00 am on Saturday AND 7:00pm on Tuesday the 16th. We will be meeting in the CIVIL AIR PATROL building at the OGDEN MUNICIPAL AIRPORT, AIRPORT ROAD.

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