



## JUNE 1985 Vol. 3 No. 6

The June meeting will be held in Thursday, June 20th at Cuyahoga Falls High School at the corner of Fourth and Stow Streets in Room 413 - Physic's Lab. Please remember to sign in.

The program this month will be "Flow Charts and Diagraming Programs". Dan and Ian will be presenting the program. The July program will be a fun and swap night. Everyone should bring their equipment to get maximum enjoyment out of your computer.

A nominating committee needs to be organized for the coming elections in September. Anyone wishing to be nominated or would like to find nominees, speak up at the June meeting.

We are now exchanging newsletters with Cin-Day users Group from West Chester, Ohio and Carnation City 99er User Group from Alliance, Ohio. We welcome these two new users groups that exchange newsletters with us as well as the others that we currently exchange newsletters with.

### CALENDER OF EVENTS:

JUNE 20 MEETING-- Flow Charts and Diagraming Programs  
JUNE 27 BOARD MEETING  
JULY 12 MEETING\_\_ Fun Night and Swap Night. Everyone should bring their equipment to get maximum enjoyment out of your computer.  
JULY 25 BOARD MEETING  
AUG. 15 MEETING\_\_ Advantages of Extended Basic over Basic  
AUG. 22 BOARD MEETING

TIPS FROM THE TIGERCUB

#22

Copyright 1985

TIGERCUB SOFTWARE  
156 Collingwood Ave.  
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

The entire contents of Tips from the Tigercub Nos. 1 through 14, with more added, are now available as a full disk of 50 programs, routines and files for just \$15.00 postpaid!

Nuts Bolts is a diskfull of 100 (that's right, 100!) XBasic utility subprograms in MERGE format, ready for you to merge into your own programs. Contents include 13 type fonts, 14 text display routines, 12 sorts and shuffles, 9 data saving and reading routines, 9 wipes, 8 pauses, 6 music, 2 protection, etc., and now also a tutorial on using subprograms, all for just \$19.95 postpaid!

And I have about 140 other absolutely original programs in Basic and XBasic at only \$3.00 each!(plus \$1.50 per order for cassette, packing and postage, or \$3.00 for diskette, PPM) Some users groups charge their members that much for public domain programs! I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

This challenge was printed in Tips #21 -

100!The Unprintable Unkeyable Program!

110!To shuffle the numbers 1 to 255 into a random sequence without duplication

120!The strings contain the ASCII characters 1 to 127 and 128 to 255

130!Most of the ASCII characters below 32 or above 159 cannot be input from the keyboard

140!So how was this program programmed?

150 M\$="

!""%&'()\*+,-./0

123456789:;<=>@ABCDEFGHIJKLMN  
OPQRSTUVWXYZ[\]^\_`abcdefg  
hijklmnopqrstuvwxyz{|}~"

160 M2\$="

170 M\$=M\$&M2\$

180 L=LEN(M\$):: RANDOMIZE ::  
X=INT(L&RND+1):: M=ASC(SEG\$(M\$,X,1)):  
M\$=SEG\$(M\$,1,X-1)&SEG\$(M\$,X+1,LEN(M\$))

190 PRINT M\$:: IF LEN(M\$)=0 THEN STOP ELSE 180

And here is the answer - It was written by a program that writes a program! Key this in and run it to create a MERGE format disk file. Then type NEW, then type MERGE DSK1.LONGSTRING and you will have a RUNable program consisting of lines 150-170 of the puzzle!

100 OPEN #1:"DSK1.LONGSTRING",VARIABLE 163

110 LN=100 :: GOSUB 190 :: A\$=L\$&M\$&CHR\$(190)

120 FOR J=1 TO 127 :: C\$=C\$&CHR\$(J):: NEXT J :: A\$=A\$&CHR\$(199)&CHR\$(127)&C\$&CHR\$(0)

130 PRINT #1:A\$

140 GOSUB 190 :: B\$=L\$&M2\$&CHR\$(190)

150 FOR J=128 TO 255 :: D\$=D\$&CHR\$(J):: NEXT J :: B\$=B\$&CHR\$(199)&CHR\$(128)&D\$&CHR\$(0)

160 PRINT #1:B\$

170 GOSUB 190 :: F\$=L\$&M\$&CHR\$(190)&M\$&CHR\$(184)&M2\$

\$&CHR\$(0)

180 PRINT #1:F\$ :: PRINT #1:CHR\$(255)&CHR\$(255):: CLOSE #1 :: END

190 L\$=CHR\$(INT(LN/256))&CHR\$(LN-256\*INT(LN/256)):: LN=LN+10 :: RETURN

Now type in the remaining lines, and you will have a speeded-up version of the Tigercub Scramble which was published in Tips #10. It is still not as fast as the CALL PEEK versions but is much more useful because you can modify it to scramble a sequence of any length anywhere between 1 and 255. For example, to shuffle the numbers 100 to 150 into a random sequence without duplication, just add a line 175 M\$=SEG\$(M\$,100,50).

The method of writing a "program that writes a program" was fully explained by John Clulow in the 99er magazine Vol. 1 Nos. 3 and 4. It is a little-used but very valuable technique.

For instance, Tips#9 contained the following routine to turn the alphabet upside-down.

```
100 FOR CH=33 TO 127 :: CALL CHARPAT(CH,CH$):: FOR J=1 TO 16 STEP 2 :: X$=SEG$(CH$,J,2)&X$ :: NEXT J :: CALL CHAR(CH,X$):: X$="" :: NEXT CH
110 INPUT A$ :: GOTO 110
```

The only trouble with that is that it takes about 50 seconds to run. Try this instead -

```
100 FOR CH=33 TO 127 :: CALL CHARPAT(CH,CH$):: FOR J=1 TO 16 STEP 2 :: X$=SEG$(CH$,J,2)&X$ :: NEXT J :: CALL WRITE(CH,X$):: X$="" :: NEXT CH
1000 SUB WRITE(CH,X$):: IF FLAG=1 THEN 1010 :: FLAG=1 :: OPEN #1:"DSK1.WRITE",OUTPUT,DISPLAY,VARIABLE 163 :: LN=3000 :: GOSUB 3000
1010 X=X+1 :: L$=L$&CHR$(200
```

&CHR\$(16)&X\$ :: IF X<5 AND CH<127 THEN L\$=L\$&CHR\$(179):: SUBEXIT

1020 X=0 :: PRINT #1:L\$&CHR\$(0):: L\$="" :: IF CH=127 THEN 1030 :: GOSUB 3000 :: SUBEXIT

1030 PRINT #1:CHR\$(255)&CHR\$(255):: CLOSE #1 :: GOTO 3010

3000 L1=INT(LN/256):: L2=LN-256&L1 :: L\$=CHR\$(L1)&CHR\$(L2)&CHR\$(147):: LN=LN+10 :: RETURN  
3010 SUBEND

RUN that, type NEW, then MERGE DSK1.WRITE, and you will have a program consisting of DATA statements containing the hex codes for all the upside-down characters. Add a line 100 FOR CH=33 TO 127 :: READ CH\$ :: CALL CHAR(CH,CH\$):: NEXT CH, and you can turn everything upside-down in only 12 seconds.

Someone sent me a classified ad, clipped from an unknown publication, which read -

TI-WRITER COMPANION. Loaded with ingenious ways to make your TI-Writer more effective. Well written. Send \$2.50 to Dr. Bill Browning, 7541 Jersey Avenue North, Brooklyn Park, MN 55420. Money back guarantee.

I sent off my money and have just received 29 pages, 3-hole punched, loaded with useful and ingenious tips and ideas for getting more out of TI-Writer. I recommend it - it's worth twice the money and then some!

The K-Town newsletter recently published a utility routine that is so useful that I want to pass it on to everyone. If a program is not resequenced after it is modified, this will compare

it with the original and prepare a MERGE format file of all the changes, for the use of others to update their copy.

```

100 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!
110 !# COMPARE PROGRAM #
120 ' by Mike Dodd #
130 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!
131 ! In k-Town 99'er V.2 #!
April 1985
140 !Version 05.0406.1XB
Requires disk drive.
Compares two programs,
gives list of all diff-
erences.
150 !SAVE old program in
MERGE format (SAVE DSK1.(old
filename),MERGE). SAVE up-
dated program in MERGE for-
mat(SAVE DSK1.(newfilename)
,MERGE)
160 !RUN this program, answe
r prompts for OLD FILE name,
NEW FILE name, and a differ
ent OUTPUT FILE name.
170 !When finished, type NEW
, then MERGE DSK1.(outputfil
ename) and ENTER
180 !Can be MERGED into othe
r copies of OLD program to
update them
190 DEF @(#)=ASC(SEG$(@#,1,
1))*256+ASC(SEG$(@#,2,1))
200 A$=CHR$(255)&CHR$(255)::
DISPLAY AT(1,1)ERASE ALL:"O
LD FILE:" : "NEW FILE:
": "OUTPUT FILE:"
210 ACCEPT AT(1,13)BEEP:B$ :
: ACCEPT AT(3,13)BEEP:C$ ::
ACCEPT AT(5,13)BEEP:D$ :: OP
EN #1:B$,INPUT ,VARIABLE 163
220 OPEN #2:C$,INPUT ,VARIAB
LE 163 :: OPEN #3:D$,OUTPUT,
VARIABLE 163
230 LINPUT #1:@$ :: LINPUT #
2:ES :: F$=SEG$(@#,1,2):: G$
=SEG$(@#,1,2):: A=@(F$):: B=
@(G$)
240 IF F$=A$ AND G$=A$ THEN
CLOSE #1 :: CLOSE #2 :: PRIM
T #3:A$ :: CLOSE #3 :: STOP
250 IF B>A THEN PRINT #3:F$&
CHR$(131)&" **DELETED LINE #
"&CHR$(0):: LINPUT #1 :: @#
:1 F$=SEG$(@#,1,2):: A=@(F$
):: GOTO 240
260 IF A>B THEN PRINT #3:ES
:: LINPUT #2:ES :: G$=SEG$(E

```

```

$,1,2):: B=@(G$):: GOTO 240
270 IF @<>E$ THEN PRINT #3:
E$
280 GOTO 230

```

Thanks to some ideas from Joyce Corker, I have made some more improvements to the Tigercub Menuloader, and I have used the above utility routine to list all the changes made since it was published in Tips#15.

```

100 !by A. Kludge/M. Gordon/
T. Boisseau/J. Peterson/etc.
modified in Tips #22
102 OPTION BASE 1 :: DIM P6$
(127),VV(127),VX(127):: GOTO
110
105 @,A,A$,B,C,D$,FLAG,I,J,K
,KD,KK,M$,MM,P$,P6$( ),Q$,S,S
T,T$( ),TT,VT,VV( ),VX( ),W$,X
,X$,K2,S2
106 CALL INIT :: CALL LOAD :
: CALL LINK :: CALL PEEK ::
CALL KEY :: CALL SCREEN :: C
ALL COLOR :: CALL CLEAR :: C
ALL VCHAR :: CALL SOUND :: !
@P-
150 ! **DELETED LINE **
160 T$(1)="d/f" :: T$(2)="d/
v" :: T$(3)="1/f" :: T$(4)="
1/v" :: T$(5)="pro" :: ON WA
RNING NEXT
170 IMAGE ###
180 DISPLAY AT(1,4):"TIGERCU
B MENU LOADER"
210 D$="DSK1." :: OPEN #1:D$
,INPUT ,RELATIVE,INTERNAL ::
INPUT #1:M$,A,J,K :: DISPLA
Y AT(1,2)SIZE(27):SEG$(D$,1,
4)&" - Diskname= "&M$:
230 FOR X=1 TO 127 :: IF X/2
0<>INT(X/20)THEN 260
240 DISPLAY AT(24,1):"Type c
hoice or 0 for more 0" :: AC
CEPT AT(24,27)VALIDATE(DIGIT
)SIZE(-3):K :: IF K=0 THEN 2
50 :: IF VV(K)<>5 THEN 411 :
: IF K>0 AND K<MM+1 THEN 420
ELSE 240
290 DISPLAY AT(X+4,2):USING
170:MM :: DISPLAY AT(X+4,6):
P$ :: P6$(MM)=P$ :: DISPLAY
AT(X+4,18):USING 170:J :: DI
SPLAY AT(X+4,22):T$(ABS(A))
291 VV(MM)=ABS(A):: VX(MM)=A
BS(B)
295 X$=" *&STR$(B):: DISPLA

```

```

Y AT(X+4,26):SEG$(X$,LEN(X$)
-2,3):: VT=VT+J
350 DISPLAY AT(X+6,1):" C
hoice?" :: ACCEPT AT(X+6,16)
SIZE(3)VALIDATE(DIGIT):K ::
IF K<>MM AND K<MM+1 THEN 41
0
410 IF K<1 OR K>127 OR LEN(P
6$(K))=0 THEN 320
411 IF VV(K)=5 OR(VV(K)=4 AND
D VX(K)=254)THEN 420
412 ON ERROR 417 :: CALL CLE
AR :: OPEN #2:D$&P6$(K):: CA
LL SCREEN(16)
413 LINPUT #2:W$ :: IF EOF(2
)THEN 416 :: PRINT W$
414 CALL KEY(0,K,S):: IF S=0
THEN 413
415 CALL KEY(0,K2,S2):: IF S
2<1 THEN 415 ELSE 413
416 CLOSE #1 :: CLOSE #2 ::
END
417 DISPLAY AT(12,10):"UNLIS
TABLE" :: CALL SOUND(200,110
,0):: RETURN 400
430 ON ERROR 417 :: CALL INI
T :: CALL PEEK(-31952,A,B)::
CALL PEEK(A#256+B-65534,A,B
):: C=A#256+B-65534 :: A$=D$
&P6$(K):: CALL LOAD(C,LEN(A$
))
The Menu Loader will
now list up to 127 programs
and files, showing the
number of sectors in each
and the file type, record
type and record length of
each file. It will stop at
the end of each page, and
continue on a default value
of 0, or will stop for
selection when any key is
pressed. It gives disk
name, number of sectors used
and available. It adds up
sectors actually used and
gives a warning if all
sectors are not accounted
for. It will load and run
any program which can be
loaded from Extended Basic,
displaying the program being
loaded. It will delete any
program or file, after first
displaying the filename and
requesting verification. It
will list any listable file
to the screen, pausing on
any key input, and can be

```

very easily modified to list to a printer. If a file is not listable, it will inform you so, and restart the menu selection. It has the pre-scan option to speed it up.

Fairly often, the disk directory will lose track of one or a few sectors during the process of loading records, even though the Disk Manager showed all 358 were initialized. That's why I put the checking routine in the Menu Loader. The figure shown as "used" is actually 358 minus the number of sectors still available, and is checked against the total sectors of all files.

The loss of a few sectors is no serious matter, but once in a great while you may notice that the "available" and "used" sector quantities have obviously been reversed. I have found that this is a signal that the disk is about to go haywire and you had best back it up immediately!

Programs and files are loaded in the first available sector, and continued in the next available sector. If a number of small files are deleted from a disk, and a long file is then loaded, it may thus be fractured into many parts. If you have a work disk on which you continually add and delete files of various lengths, it will become badly fractured. This can cause disk errors, and it also badly overworks your drive. It is a good idea to recopy your work disk occasionally - file by file, not sector by sector with a quick copier.

MEMORY FULL' - Jim Peterson

## TI TERMINAL EMULATOR II ALTERNATIVES

by William M. Lucid

This article will cover two terminal emulator programs for the TI 99/4A system. I use the word system because these programs require: 1. RS232, 2. 32K memory expansion, 3. Disk system, and 4. Editor/Assembler module or CorComp disk controller. Programs covered in this article are TE-1200, and PTERM-99.

The purpose of a terminal emulator is to enable transfer of data between computers, allowing sending and receiving systems to "talk" in a recognized method of handling data, even when sending computer is different than receiving computer. The terminal emulator program is used to set communication parameters for using a MODEM with a RS232 interface. Some parameters encountered in terminal emulators are BAUD rate, RS232 (1-4), Parity, Stop bits, Echo, and Data bits. Another use of a terminal emulator program is to software interface radio amateur equipment. I have been able to use these programs to interface with KANTRONICS UTU for receiving RTTY as well as other modes of communications without having to layout additional money for KANTRONICS, HAMSOFT software for the TI 99 4/A.

TE-1200 by E. Earle Thompson was the first alternative for the TI TERMINAL EMULATOR II, that had capabilities of "auto-logging". Auto-logging allows you to use 32K memory expansion to "hold" incoming data. When the 12.5 k of ram buffer is full, program automatically dumps the 12.5 k bytes, (approximately 48 sectors) of data to a disk file that has been set-up pressing "Control" and "4" keys when parameters were inputed. The outputted file is a display, variable 128. Outputting of the file is done while ON-LINE, this increases your on-line charges, possibly long distance such as SOURCE and COMPUSERVE. TE-1200 communications parameters can be re-entered anytime by pressing "control" and "1" keys. Also to recall an Auto-logged file to use with a editor program such as TI-WRITER or EDITOR/ASSEMBLER the display, variable 128 file must be converted to a display, variable 80 file. TE-1200 allows the user to select baud rates of 110 upto 9600, baud rate dependent on modem baud rate, capability of RS232, and some other variables such as line noise, tolerance for error. The last item is important in cases where no "missing" data can be

tolerated, such as transmitting an assembly file. The faster the baud rate, the greater the risk of garbled or lost data when using non-deciated telephone lines. TE-1200 supports same file transfers as TI TERMINAL EMULATOR II, this feature allow you to use downloading feature of T.I.B.B.S. bulletin boards which require either TI TERMINAL EMULATOR II or TE-1200.

Another excellent terminal emulator program is P-TERM-99 by C. Richard Bryant. This program lives up to its claim of, "the ultimate terminal program. 300/1200 baud, 24K download buffer, 20K upload buffer, 256 color combinations and many more options." One of those other options is the ability to toggle your printer on and off by pressing "control" and "1" keys, with this feature you can simultaneously display and print at the same time or toggle printer off to only display data. Default values are for 300 baud, RS232 port 1, even parity, 1 stop bit, and 7 bits data. Default values can be selected-at first prompt by pressing enter twice. PTERM-99 will load from Extended Basic, Editor/Assembler or Mini-Memory modules. This program allows for resetting communication parameters at any time by pressing "control" and "7" keys. Pressing "function" and "7" allows you to select foreground and background colors of your preference, choices are presented on screen to aid in making selection. Outputting of the download buffer when full will occupy about 110 sectors. When the download buffer is within 1K of being full the screen will turn red, this feature works very well. When screen turns red you must dump the download buffer or the download buffer will be over written. Download buffer file output, if a disk file is in the display variable 80 format. Upload text files need to be in display variable 80 format. I have found this works best by removing control characters. T.I.B.B.S. bulletin boards will not "recognize" PTERM-99 the first time you attempt to log on. Another disadvantage is true TE II "File transfer protocol" is not a feature of this well planned, easy to use, and economical program. PTERM-99 is a reliable, dependable, and proven program, well worth the \$17.50!

(TE-1200 is published by Softmail, PO Box 745, Rockwall, TX 75087. TE-1200 is listed in UNISOURCE catalog. UNISOURCE has a toll free telephone number 1-800-858-4580, there address is Box 64240, Lubbock, TX 79464 the last price update I received shows TE-1200 costing \$39.95. PTERM-99 is being sold by TEXAMENTS, 53 Center Street, Patchogue, NY 11772, program costs \$17.50 and includes shipping.)

The following article comes to us from 99 HOCUS, May 1985 issue.

## NOISELESS PERIPHERAL EXPANSION BOX

By Bob Hubel MSP US

Are you distracted & disturbed by the tornadic roar of the Peripheral Expansion Box ? TI engineered this box to provide sufficient cooling capacity for the most strenuous of circumstances -- all 8 card slots occupied and under heavy, continuous usage. Since my use didn't approach the design limits, I experimented with ways to slow the fan down, and I was successful in reducing that noise level to a barely perceptible purr ! I have now been testing this change for a sufficient period of time, and I feel comfortable in recommending the modification to all. In fact I have even run under light loads for moderate periods of time without any fan at all, but I don't advise going to that extreme.

I had considered 2 options, buy a quieter fan or slow down the fan I have. A quieter fan costs about \$20 or more, so I quickly discarded that option. I could reduce the fan speed with either a special solid-state voltage regulation device or

merely install a power resistor in series with the fan. Since I have no experience with such solid-state devices and the components would as likely cost around \$10, I elected for the latter method at a cost of \$0, using components from my box of electronic junk. If you had to buy the components at retail, the cost would amount to no more than \$2-\$3

The PEB fan is rated at 14 Watts. I have found that inserting a 500 - 700 Ohm power resistor, 10 Watts power dissipation reduces the speed to a very acceptable level. 700 Ohms is my own preference. Series of 2 or more resistors may be combined to add up to 700 Ohms. The resistor "steals" the energy that originally was intended for the fan, thus the fan doesn't work as hard. However in doing so, the resistor must shed the extra energy itself and does so by producing heat. Therefore you should mount the resistor outside of the PEB, immediately behind the fan thus allowing the circulating air to cool it. Mounting it inside the cabinet, although aesthetically more pleasing will add unwanted heat inside the cabinet. Procedure to disassemble the PEB to access the fan lead wires

- 1) Unplug power cord and remove lid
- 2) Disconnect and remove disk drive
- 3) Remove all slide-in cards
- 4) Remove cabinet screws (7/14?) on bottom, back and sides
- 5) Slide cabinet base out
- 6) Disconnect 1 of the fan leads
- 7) Extend the 2 wires thru holes in the cabinet back
- 8) Splice resistor(s) in series and mount on box near exhaust
- 9) Reassemble box

Since the fan air intake is thru the card cage, I cut foam to fit in the empty slots at the far forward side of the card cage, forcing the reduced airflow thru the existing cards, increasing their cooling.

The resistors will run warm and possibly hot, this is normal. Don't cheat and use less than a 10 Watt resistor which may get too hot and cause problems.

You'll be amazed at the reduction in the noise pollution. If anyone knows of an inexpensive solid-state equivalent to perform the identical function, please notify me.

Good Luck !!

This article comes to us from Chattanooga TI 99/4A Users Group, May 1985 issue.

### Software Piracy and the Death of a Computer: by Donald M. Thomson III of M & T Utilityware

I would like to address a problem that is continually plaguing our fantastic computer. This problem, as just about everyone knows, is called "piracy". I currently market three programs for the TI-99/4A, of which are considered by many to be pretty good programs and not the "ripoffs" that quite a few people have mentioned to me, seem to keep coming down the pike. Contrary to popular "piracy" beliefs, my software is not "public domain" and is not supposed to be distributed by unauthorized persons. I know however, my software has entered the dreaded "piracy pipeline", and is now in the hands of many people who have obtained it illegally. I would like to say something to the people who are doing it: "I sincerely hope that a burglar breaks into your house and steals something that is very valuable to you, since it is the same thing you are doing to me!" I started out in this business with the idea that I had some unique programs that I thought were pretty good and might be of some value to other people. I have quite a few ideas for other utility programs written in both Assembly Language and Pascal. I purchased the Pascal System in hopes that some day software would be available for it. So far, nothing has happened, so I will be writing some myself. So, those of you who have purchased the Pascal System, don't give up on it yet. I decided to market my utility programs, with the idea that it would be very nice to have all of your utility software on one disk. That way you don't have to keep swapping disks in and out of your disk drives. Armed with that idea, I did not spend the time dreaming up some elaborate protection system that would have been broken anyway. I felt it would be more worthwhile to the TI-99/4A owners for me to spend my time writing programs they could use. I figured that it would take quite a long time to write protection scheme just to outsmart the pirates. So much for good will. The pirates have decided, to take it upon themselves, to distribute my software illegally for me. I even went so far as to include the source code with my programs, so that people might be able to learn "assembly language" a little easier. I would like to repeat a comment that was made to me by a "pirate" who approached me at the "TI FAIRE", which was held in Chicago on November 10, 1984. He told me that "his pirating of software, software that someone had spent many hours of their life creating, actually stimulates the market". How can software piracy possibly stimulate a market? All it does is contribute to the premature "death" of a computer. He told me that if people get software, then they will buy hardware. All I have to say about this is "hogwash"! The only thing software piracy

accomplishes is to make programmers, like myself and others, say "TO HELL WITH IT". At that time a major void has been created in an area that makes computers tick, the software. Can you imagine what it would be like if programmers started saying "TO HELL WITH IT"? Well there would be no new and better software. The next logical chain of events is that people would start dumping their computers! The outcome is simple: "the premature death of a computer due to the selfishness of people who think they are stimulating the market!". The only reason this fantastic computer has held on so long, after being abandoned by Texas Instruments, is because of the good will, hard work and grace of honest people. I think that we should thank GOD that there are more "honest" people than there are "computer killers". I feel that software for the TI-99/4A is already relatively low priced. I am beginning to see better quality software coming out, on a regularly basis, and with lower price tags. I simply can not understand why the "pirates" find it necessary to try and ruin a good thing? Why are you folks trying to kill our computer? I simply do not understand it! The mad Texas Instruments bailout panic is over. There is now a lot of good quality software available for our machine. I think that instead of trying to kill it, why don't you folks try and do something constructive, like contribute to the long life of the machine instead of its premature death? Those who have purchased my software legally will find that they will receive full technical and update support, as well as, a money back guarantee. I don't know, at this time, of any software company for any other computer that makes this offer. A note to all of the honest TI-99/4A owners: Think about that when you are offered a "pirated" copy of software. If you buy my programs from a pirate, you will get no support, as well as, no money back guarantee. I don't know of any "pirates" who guarantee stuff they steal! In conclusion, we don't want to prove the industry right by letting the "pirates" kill our computer. Ignore the "pirates" and support those people who are working to make one of the most powerful home computers around survive!

The following article comes to us from 99 HOCUS, May 1985 issue.

## FORCING PRINTER PAUSES

by Abdallah Clark

If you want to change, in the middle of your text, to another printwheel or differently colored ribbon, use the ALTERNATE INPUT Command at the point in your text where you want to make the switch. When used in text without a separate file being specified for a "Mailing List" option in the Text Formatter, the ALTERNATE INPUT Command stops the printer and then leaves TI-WRITER waiting for your input from the keyboard. Change your daisywheel or dot-matrix printer as needed, then press ENTER, and the printing continues to completion from that spot.

Since no harm is done by just pushing ENTER, you have an easy way of pausing, not aborting, the printout process! However, only one such change may be made per line of text, unless you do some really fancy work with the Transliterate Command.

Also, it seems the results will be more dependable if you put the DEFINE PROMPT Command (and its carriage return) on a separate line. If your ALTERNATE INPUT is on a line to itself, though, you will have a linefeed there by pressing ENTER. However, place it where needed or substitute them for a carriage return symbol or reduce the ".SP n" format command by one to retain your text's form.

It is also important to note that the ALTERNATE INPUT Command is the only Command that does not use a leading period as a signal to the Text Formatter that it is a special function symbol. This has two effects. One, don't let habit make you use a period, or you will have an unwanted period in your text. Two, you cannot use that combination of characters in your own document, unless indirectly by way of transliteration.

Another note to be mentioned about the ALTERNATE INPUT Command is that the digit used between the asterisks may only be used once, whether this pertains to a single document, or a series of documents "linked" by the INCLUDE FILE command. Be careful not to exceed the maximum of 99 for that digit in the ALTERNATE INPUT Command when you have a series of documents, too.

You may also use the DEFINE PROMPT Command in conjunction with the ALTERNATE INPUT Command, to compose a memory jogger message for the task needed. (Remember: always give the ".DP xxxxx" first, then the particular ALTERNATE INPUT Command.) This way, when the Text Formatter prints the document, the printer will stop where the ALTERNATE INPUT group of characters are located and your prompt appears on the monitor/TV display. It's even easier than you would think if you list all the prompts at the beginning of your document, because then you may move/rearrange text to your heart's content without having to worry about whether you're keeping the prompt ahead of the input. A handy convenience if you make more than one of these printer changes in the course of one document!

----- Abdallah Clark

Copyright 1985

TIGERCUB SOFTWARE  
156 Collingwood Ave.  
Columbus, OH 43213

Distributed by Tigercub Software to T1-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

The entire contents of Tips from the Tigercub Nos. 1 through 14, with more added, are now available as a full disk of 50 programs, routines and files for just \$15.00 postpaid!

Nuts & Bolts is a diskfull of 100 (that's right, 100!) XBasic utility subprograms in MERGE format, ready for you to merge into your own programs. Contents include 13 type fonts, 14 text display routines, 12 sorts and shuffles, 9 data saving and reading routines, 9 wipes, 8 pauses, 6 music, 2 protection, etc., and now also a tutorial on using subprograms, all for just \$19.95 postpaid!

And I have about 140 other absolutely original programs in Basic and XBasic at only \$3.00 each!(plus \$1.50 per order for cassette, packing and postage, or \$3.00 for diskette, PPM) I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

Several different routines have been published which will extract and save a specified series of lines out of a program, but this one by George Steffen of the L.A. 99ers is certainly the

best.

```
1 !SUBROUTINE EXTRACTOR by G
eorge F. Steffen. SAVE in ME
RGE format. MERGE into any p
rogram (with line 0 starting
above 8). RUN to extract
2 !selected lines. Deletes i
tself. Then BE SURE to SAVE
the selected lines in MERGE
format because the remaining
lines are still in memory!
3 CALL CLEAR :: CALL INIT ::
INPUT "Line numbers of rout
ine to be saved: First,Last?
":L,M :: G=256 :: CAL
L PEEK(-31952,H,I,J,K)
4 C=INT(M/6):: D=M-C*6 :: F=
(J-G)*6+K :: FOR E=(H-6)*6+I
TO F STEP 4 :: CALL PEEK(E,
A,B):: IF A=C AND B=D THEN 6
5 NEXT E :: PRINT "LINE";M;
"NOT FOUND!" :: STOP !@P-
6 H=INT(E/6):: I=E-(6*H):: H
=H+6 :: C=INT(L/6):: D=L-C*6
:: FOR E=E+4 TO F STEP 4 ::
CALL PEEK(E,A,B):: IF A=C A
ND B=D THEN 8 !@P-
7 NEXT E :: PRINT "LINE";L;
"not found!" :: STOP !@P-
8 E=E+3 :: J=INT(E/6):: K=E-
(6*J):: J=J+6 :: CALL LOAD(-
31952,H,I,J,K):: STOP !@P-
```

The enhancements to my Menu Loader, published in Tips #22, contained an error. Please change line 413 to read -  
413 LINPUT #2:M\$ :: PRINT M\$  
:: IF EOF(2)THEN 416

Some folks were interested in the idea of a program that writes a program, so let's write a program that will write a program to list the token codes that you need to use to write a program that will write a program -

```
100 OPEN #1:"DSK1.TOKENLIST"
,OUTPUT,DISPLAY ,VARIABLE 16
3 :: FOR N=129 TO 254 :: L1=
INT(N/256):: L2=N-256*L1
110 PRINT #1:CHR$(L1)&CHR$(L
2)&CHR$(131)&CHR$(N)&CHR$(0)
:: NEXT N
120 PRINT #1:CHR$(255)&CHR$(
```

255):: CLOSE #1 :: END

Key that in and SAVE it just in case, then RUN it. When READY, type NEW, then MERGE DSK1.TOKENLIST. Now LIST it and you will see a list of ASCII codes 129 through 254 and their token meanings. Delete lines 171 through 175, 185, 198, 226 through 231, and 242. Change the definition of 199 to QUOTED STRING, of 200 to UNQUOTED STRING, and add line 255 END OF FILE.

You don't need all those exclamation points, so change the program to a DIS/VAR B0 file by LIST "DSK1.TOKENLIST". Then key in this little routine.

```
100 OPEN #1:"DSK1.TOKENLIST"
:: OPEN #2:"PIO"
110 LINPUT #1:A$ :: PRINT #2
:SEG$(A$,1,4)&SEG$(A$,6,LEN(
A$)): IF EOF(1)<>1 THEN 110
120 CLOSE #1 :: CLOSE #2 ::
END
```

RUN it, and print out a list of all the token codes. More on this next month - if someone buys a few programs so that I can afford another month.

Now that we've done about all that we can with the Menu Loader, here is another version to use on your finalized library disks of programs. It lacks the features that you will no longer need, but will list your programs by their full names, up to 24 characters long.

```
100 !NAMELOADER by A. Kludge
/M. Gordon/T. Boisseau/J. Pe
terson/etc.
110 CALL CLEAR :: CALL SCREE
N(5):: FOR S=1 TO 14 :: CALL
COLOR(S,7,16):: NEXT S :: C
ALL VCHAR(1,31,1,96):: CALL
COLOR(0,2,16)
120 OPTION BASE 1 :: DIM P$(
99),M$(99)
```

```
130 ! List the full names of
the programs on the disk in
the DATA statements, in the
sequence in which they are
listed by an ordinary disk
cataloger program
140 !Then SAVE this program
under the filename LOAD
150 DATA
160 DATA
170 DATA
180 DATA
190 DATA END
200 FOR J=1 TO 99 :: READ M$(
J):: M$(J)=SEG$(M$(J),1,24)
210 IF M$(J)="END" THEN M$(J
)=" " :: GOTO 230
220 NEXT J
230 IMAGE ##
240 DISPLAY AT(1,4):"TIGERCU
B NAMELOADER"
250 D$="DSK1." :: OPEN #1:D$
,INPUT ,RELATIVE,INTERNAL ::
INPUT #1:P$
260 FOR X=1 TO 99 :: IF X/20
<>INT(X/20)THEN 290
270 DISPLAY AT(24,1):"Type #
of choice or Enter 0" :: AC
CEPT AT(24,27)VALIDATE(DIGIT
)SIZE(-3):K :: IF K=0 THEN 2
80 :: IF K>0 AND K<=NN+1 THEN
390 ELSE 270
280 X=1
290 I=I+1 :: IF I>127 THEN K
=X :: GOTO 370
300 INPUT #1:P$ :: NN=NN+1
310 IF LEN(P$)=0 THEN 350
320 DISPLAY AT(X+3,2):USING
230:NN :: DISPLAY AT(X+3,5):
M$(NN):: P$(NN)=P$
330 CALL KEY(0,KK,ST):: IF S
T=0 THEN 340 :: FLAG=1 :: GO
TO 350
340 NEXT X
350 DISPLAY AT(X+4,1):" " ::
DISPLAY AT(X+5,2):USING 230
:NN+1 :: DISPLAY AT(X+5,6):"
Terminate"
360 DISPLAY AT(X+6,1):" C
hoice?" :: ACCEPT AT(X+6,16)
SIZE(2)VALIDATE(DIGIT):K ::
IF K<=NN AND K<=NN+1 THEN 38
0
370 IF K=NN+1 THEN CALL CLEA
R :: CLOSE #1 :: END
380 !IF K<1 OR K>99 OR LEN(P
$(K))=0 THEN 350
390 CLOSE #1
400 CALL INIT :: CALL PEEK(
31952,A,B):: CALL PEEK(A;256
```

```

+B-65534,A,B):: C=A*256+B-65
534 :: A=D*P6*(K):: CALL L
OAD(C,LEN(A*))
410 FOR I=1 TO LEN(A*): CAL
L LOAD(C+I,ASC(SE6*(A*,I,1)
):: NEXT I :: CALL LOAD(C+I,
0)
420 CALL VCHAR(1,3,32,672)::
CALL SCREEN(0):: FOR S=0 TO
14 :: CALL COLOR(S,2,1):: M
EXT S :: DISPLAY AT(12,2):"L
OADING *;M*(K)
430 RUN "DSK1.1234567890"

```

Last month I forgot to have anything for the kids, or anything in Basic, so -

```

100 CALL CLEAR
110 REM by Jim Peterson of
Tigercub Software
120 PRINT TAB(1);"****AUTOMA
TIC MOUSE MAZE****": : : "
Choose your mouse and":"wa
tch it try to find its way"
130 PRINT "through the maze.
": : " When one of the mice
has":"taken 50 extra steps,
the":"cat gets it!"
140 PRINT : : "Touch any key"
150 CALL KEY(0,K,ST)
160 IF ST<1 THEN 150
170 CALL CLEAR
180 CALL CHAR(120,"007BFEFF
E78")
190 CALL CHAR(121,"1038387C7
C7C7C38")
200 CALL CHAR(122,"387C7C7C7
C383810")
210 CALL CHAR(123,"001E7FFF
F1E")
220 CALL CHAR(128,"001E61816
11E")
230 CALL CHAR(129,"384444444
4242410")
240 CALL CHAR(130,"102828444
4444438")
250 CALL CHAR(131,"007886818
678")
260 CALL SCREEN(5)
270 T1=610
280 T2=610
290 CALL CHAR(136,"FFFFFFF
FFFFFF")
300 CALL COLOR(14,16,16)
310 CALL COLOR(13,2,16)
320 CALL COLOR(12,2,16)
330 R=10
340 GOSUB 1460

```

```

350 R1=10
360 C=2
370 C1=2
380 CALL HCHAR(R,C,136,2)
390 C=C+1
400 M=120
410 M2=128
420 RANDOMIZE
430 A=(INT(2*8RND)+1)*2
440 B=INT(10*8RND)+1
450 ON B GOSUB 470,470,470,4
70,510,510,550,550,590,590
460 GOTO 420
470 IF C+A>30 THEN 630
480 CALL HCHAR(R,C,136,A)
490 C=C+A
500 RETURN
510 IF R+A>20 THEN 540
520 CALL VCHAR(R,C,136,A)
530 R=R+A
540 RETURN
550 IF R-A<2 THEN 580
560 CALL VCHAR(R-A+1,C,136,A
)
570 R=R-A
580 RETURN
590 IF C-A<3 THEN 620
600 CALL HCHAR(R,C-A+1,136,A
)
610 C=C-A
620 RETURN
630 CALL HCHAR(R,C,136)
640 C=C+1
650 IF C<31 THEN 630
660 R2=R
670 C2=C
680 CALL HCHAR(R1,C1,M)
690 CALL HCHAR(R2,C2,M2)
700 Y=Y+1+(Y=2)*2
710 IF Y=2 THEN 1020
720 CALL HCHAR(R1,C1,136)
730 ON M-119 GOTO 800,900,74
0,850
740 IF C1=31 THEN 950
750 CALL GCHAR(R1,C1+1,6)
760 IF 6=32 THEN 850
770 C1=C1+1
780 M=120
790 GOTO 950
800 CALL GCHAR(R1-1,C1,6)
810 IF 6=32 THEN 740
820 R1=R1-1
830 M=121
840 GOTO 950
850 CALL GCHAR(R1+1,C1,6)
860 IF 6=32 THEN 900
870 R1=R1+1
880 M=122
890 GOTO 950
900 CALL GCHAR(R1,C1-1,6)

```

```

910 IF 6=32 THEN 800
920 C1=C1-1
930 M=123
940 GOTO 950
950 CALL HCHAR(R1,C1,M)
960 IF (C1=31)*(C2=2) THEN 13
20
970 IF C1<31 THEN 700
980 T2=T2-10
990 CALL SOUND(50,T2,5)
1000 IF T2=110 THEN 1340
1010 GOTO 700
1020 CALL HCHAR(R2,C2,136)
1030 ON M2-127 GOTO 1040,120
0,1090,1150
1040 CALL GCHAR(R2+1,C2,6)
1050 IF 6=32 THEN 1090
1060 R2=R2+1
1070 M2=129
1080 GOTO 1250
1090 IF C2=2 THEN 1250
1100 CALL GCHAR(R2,C2-1,6)
1110 IF 6=32 THEN 1150
1120 C2=C2-1
1130 M2=128
1140 GOTO 1250
1150 CALL GCHAR(R2-1,C2,6)
1160 IF 6=32 THEN 1200
1170 R2=R2-1
1180 M2=130
1190 GOTO 1250
1200 CALL GCHAR(R2,C2+1,6)
1210 IF 6=32 THEN 1040
1220 C2=C2+1
1230 M2=131
1240 GOTO 1250
1250 CALL HCHAR(R2,C2,M2)
1260 IF (C2=2)*(C1=31) THEN 1
320
1270 IF C2>2 THEN 700
1280 T1=T1-10
1290 CALL SOUND(50,T1,5)
1300 IF T1=110 THEN 1370
1310 GOTO 700
1320 CALL HCHAR(1,1,32,760)
1330 GOTO 330
1340 GOSUB 1460
1350 PRINT "THE CAT GOT THE
WHITE MOUSE": :
1360 GOTO 1390
1370 GOSUB 1460
1380 PRINT "THE CAT GOT THE
BLACK MOUSE": :
1390 PRINT "TO PLAY AGAIN, T
OUCH ANY KEY"
1400 CALL KEY(0,K,ST)
1410 IF ST<1 THEN 1400
1420 T1=610
1430 T2=610
1440 CALL HCHAR(1,1,32,760)

```

```

1450 GOTO 330
1460 CALL HCHAR(23,1,32,32)
1470 PRINT CHR$(120);(610-T1
)/10;TAB(20);CHR$(128);(610-
T2)/10
1480 RETURN

```

Did you know that ACCEPT AT(1,0) will accept a full line of 28 characters? Did you know that ACCEPT AT(R,0)SIZE(-28) and Enter will accept everything on row R? And did you know that ACCEPT M\$ will accept a string of 255 characters?

Need a filler, so -

```

100 !MUSICAL BARGRAPH by Jim
Peterson
110 CALL CLEAR :: CALL SCREE
N(5):: FOR J=2 TO 14 :: X=J-
(J>4):: CALL COLOR(J,X,1)::
NEXT J
120 DIM M$(13),N(13):: M$=" (
00HPX'hpX"&CHR$(128)&CHR$(1
36):: FOR J=1 TO 13 :: M$(J)
=SE6*(M$,J,1):: DISPLAY AT(J
+6,1)SIZE(1):M$(J):: NEXT J
130 X=110 :: FOR J=1 TO 13 :
: N(J)=X*1.059463094^(J-1)::
NEXT J
140 A=INT(13*8RND+1):: B=INT(
25*8RND+1):: DISPLAY AT(A+6,2
)SIZE(28):RPT$(M$(A),B):: CA
LL SOUND(B*40,N(A),0,N(A)*2+
4,0,N(A)*4+6,0)
150 DISPLAY AT(A+6,2):" ::
GOTO 140

```

MEMORY FULL

Jim Peterson



This article comes to us from BYTEMONGER, June 1985 issue.

# GRAPHICS - CHARACTER SETS

A Review by Don MacClellan

I have been waiting anxiously for some time since Dave Rose released his 'Gothic Letters' to public Domain to see what his additions and improvements to it would be. 'Character Sets and Graphic Design' is the name of a recently released program which makes possible printing of 10 or more large (and unique) character sets along with graphics and pictures.

The minute I received the CIN-DAY Newsletter with the announcement of its availability I ordered it. It was waiting when I got back from vacation and to say the least, I am most impressed. The package consists of three diskettes: Instructions (about 10 pages), Programs and a Data Diskette. Once you have printed out the instructions you may store that one in a safe place. The Program and Data diskettes are SSSD and the program reminds the user when to install the appropriate diskette. I combined my set on a DSDO diskette but have not taken the time to revise the prompts.

The Character Fonts which were shown at the May meeting and the more professional appearance of the BYTEMONGER are the result of using this very versatile program. The maximum character size is three columns wide by four rows (lines) as illustrated by the Gay Ninetys and the Oblong characters. The program also allows compressed letters and automatic centering. The program is available for two printer formats: that of the Prowriter and the Epson-Gemini. In addition to the

diskette DIS/VAR 80 instructions, written detailed instructions are also included. There are several picture files included so that you may try out printing pictures in normal or double size and the optional picture framing feature.

One of the versatile features of this program which, in addition to allowing you to design your own pictures and graphics, is it allows you to design your own character sets. The instructions tell you how, and also show you how much work Dave has in all these character definitions. If you create a character set using this package and send it to Dave, he will send you a diskette of character sets when he has received enough to fill a diskette. Overall, I think the price of \$17.95 makes this one of the best computer software buys ever; particularly since it includes \$2.88 in postage and three diskettes. The amount of work which went into defining 10 character sets alone is tremendous, not to mention a very professional programming job. If you need to make posters, announcements, newsletters, dress up correspondence, reports and make your work look like it came from a professional print shop - this is it. Order from: Dave Rose, 2781 Resor Rd., Fairfield, Ohio 45814-5853

## LIST OF BOARD MEMBERS AND THEIR HOME PHONE NUMBERS

President, Norm Sorkin	678-2360
Vice President,	
Librarian, Bert Haase	753-7846
V.P. Program, John Tuesday	644-2616
Secretary, Vicky Chrisman	784-0943
Treasurer, Betty Duncan	633-5217
Educational Director, Rich Williams	626-2423
Editor, Kathi Anderson	923-7530



Guyahoga Falls, Ohio 44223  
 P.O. Box 3201  
 SUMMIT 99ers USERS GROUP

# - FORTH -

The following FORTH programs originated from the San Antonio Area 99UG. The first - MINI-FORTH WRITER takes less than 250 bytes if you leave out the comments. The program written by J. Volk allows the user to compose up to 80 character lines. Do your editing before hitting enter. Enter is also used for LF. The second program is a sprite demo and comes from Channel 99 Hamilton UG of Ontario.

SCR #116  
 0 ( MINI FORTH WRITER BY J. VOLK )  
 1 ( Load options needed: -PRINT -SYNONYMS then 'RUN' to start.  
 2 FCTN 4 OR CLEAR TO END ) : END ." TYPE FORGET END" ;  
 3 : CIT PAD 80 BLANKS ; ( Clears 80 spaces in PAD )  
 4 : ETR PAD 80 EXPECT ; ( Enter text into PAD mem location )  
 5 : PTX SWCH PAD 80 TYPE CR UNSWCH ; ( After text is entered  
 6 by ETR, this word turns on the printer, according to your

7 parameters on screen #72 on the System Disk, and takes the  
 8 characters kept in PAD and prints them to the printer. A CR  
 9 is issued, the printer is turned off and control returns to  
 10 the screen.) : TOP CLS 0 @ @ GUTDXY ; ( Clears screen and puts  
 11 cursor in home position.) : RUN TOP BEGIN CIT ETR CR PTX  
 12 ?TERMINAL UNTIL END ; ( Final run word. These sections can  
 13 be entered directly from terminal without loading screen. )  
 14  
 15

SCR #114  
 0 ( SPRITE DEMONSTRATION 15 FEB 1984 J. VOLK )  
 1 ( Load these first -GRAPH -VDFMODES )  
 2 ( then type 'RUN' any key will stop demonstration )  
 3 MULTI MINIT 0 MAGNIFY  
 4 HEX 800 SSDT  
 5 HEX 0008 001C 7F1C 0008 60 SPCHAR  
 6 DECIMAL  
 7 : PUT 32 0 DO 124 96 13 2+ RND 96 I SPRITE LOOP ;  
 8 : MD 32 0 DO RANDOMIZE 125 RND 1+ -125 RND 1+ I MOTION LOOP ;  
 9 : MESSGE DELALL TEXT ." TYPE 'FORGET PUT' TO SAVE MEMORY " ;  
 10 : RUN PUT 32 MOTION MD BEGIN ?KEY UNTIL MESSGE ;