



## OCTOBER 1985 Vol. 3 No. 10

The October meeting will be held on Thursday, Oct. 17 at Cuyahoga Falls High School at the corner of Fourth and Stow Streets in Room 413 - Physic's Lab. The room will be open at 7:00 and the meeting will start at 7:30 PM. The November meeting will be held on Thursday, November 21. Please remember to sign in.

This month will will vote on whether to merger with the Akron Area Users Group instead of having a program. Norm has mailed out a letter explaining some of the terms of the merger. If you can not attend the October meeting, we must receive your proxy by Oct. 16.

We will hold the November meeting at the same location, but the December meeting will be held on Thrusday, Dec. 19 at the Akron Library auditorium.

NAME CONTEST: If we merge, we will need a new name. Norm will take all your suggestions. If your name is chosen, you will receive one year's dues free.

### LIST OF BOARD MEMBERS AND THEIR HOME PHONE NUMBERS

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

## MENUS & ON GOTO

Let's take a look at how this labor saving function can be used with a menu. Program 1 shows a typical menu using IF statements to direct traffic to the correct part of the program.

```
100 REM ** PROGRAM 1 **
110 PRINT "1 - LOAD"
120 PRINT "2 - SAVE"
130 PRINT "3 - EDIT"
140 PRINT "4 - PRINT" :::
150 INPUT A
160 IF A=1 THEN 300
170 IF A=2 THEN 400
180 IF A=3 THEN 500
190 IF A=4 THEN 600
200 GOTO 150
300 PRINT "LOAD"
310 GOTO 1000
400 PRINT "SAVE"
410 GOTO 1000
500 PRINT "EDIT"
510 GOTO 1000
600 PRINT "PRINT"
1000 END
```

Since this program has an INPUT statement, I get to discuss my favorite topic, error trapping. INPUT is one time the computer has to put up with the incompetence of humankind.

First type in program 1. Statements 110 through 140 display the menu, 150 accepts option selected, 160 through 190 identify the option and sends the program to the proper location. 300 through 600 prints the selected option. Line 200 is an error trapper. If any number other than 1,2,3 or 4 is entered, you are sent back to INPUT. Try it. Now try entering a letter. Next try pressing <ENTER> without pressing anything else first. As you can see, there is still other error trapping needed. More later.

Next let's shorten the program by using ON GOTO. First delete lines 160 through 200. Then enter the following new lines:

```
160 IF(A<1)+(A>4) THEN 150
170 ON A GOTO 300,400,500,600
```

Depending on the value of A, line 170 will send the program to 1st, 2nd, 3rd or 4th line number in the list. Read the + in line 160 as OR. It acts as an error trapper by sending the program back to INPUT if a number other than 1,

2, 3 or 4 is entered. We still have the problem with the other error types.

We can trap normal errors by using the ASCII numbers for 1, 2, 3 and 4. Change the following lines:

```
150 INPUT A$
160 IF(B<49)+(B>52) THEN 150
170 ON B-48 GOTO 300,400,500,600
```

Also add:

```
154 IF A$="" THEN 150
157 B=ASC(A$)
```

Line 150 will accept any character, 1, 2, 3,4,A,Q,etc. Line 154 will trap pressing <ENTER> without first pressing a number or character. Line 157 sets B equal to the ASCII number of the character entered. Line 160 will send the program back to INPUT if any character but 1, 2, 3 or 4 is entered—ASCII 49,50,51 or 52. Line 170 subtracts 48 from the ASCII number to get the 1, 2, 3 or 4 needed to make ON GOTO work. Try it, enter 2,8,-3, K or #. Also press <ENTER> without first entering a character.

Russ Cook

## MORE "ON GOTO"

Numbers for a menu are easy for the computer but what if you want to make it easy for the user with mnemonics—L for Load, S for Save, etc. Up to now I would have used the original program with IF statements. I didn't have any bright ideas but Randy Thompson of Home Computer Magazine did. He gets credit for bringing the following technique to my attention (Vol. 5, No. 5, page 111).

```
100 REM ** PROGRAM 2 **
105 B$="LSEP"
110 PRINT "L - LOAD"
120 PRINT "S - SAVE"
130 PRINT "E - EDIT"
140 PRINT "P - PRINT" :::
150 INPUT A$
155 IF A$="" THEN 150
160 IF POS(B$,A$,1)=0 THEN 150
170 ON POS(B$,A$,1) GOTO 300,400,500,600
```

```

300 PRINT "LOAD"
310 GOTO 1000
400 PRINT "SAVE"
410 GOTO 1000
500 PRINT "EDIT"
510 GOTO 1000
600 PRINT "PRINT"
1000 END

```

Line 105 places the four letters into the string variable B\$ in the correct order for the ON GOTO statement. Lines 110 through 140 display the menu, line 150 inputs selected letter, line 155 traps pressing <ENTER> without entering a letter first, line 160 using POS locates the position of the character INPUT to A\$ in variable B\$. If the character in A\$ is not in B\$, a 0 is returned and would bomb out the ON GOTO statement. Line 160 therefore traps any character that is not desired. If a good character is found then the program moves to 170 and POS returns the position number for the proper line number from the list. Lines 300 through 600 print the proper titles.

It would be neater to replace lines 160 and 170 with:

```

160 P=POS(B$,A$,1)
165 IF P=0 THEN 150
170 ON P GOTO 300,400,500,600

```

I was trying to show the use of POS in an ON GOTO statement that could be used in TI or EXTENDED BASIC.

Russ Cook

FOR SALE

TI 99/4A Console  
Modulator  
Power Supply  
Cassette Cable

\$50.00

Call Lee Martin  
666-3984

BASIC CLASS

Rich Williams class will be on Sprites. Bring your blue book.

Again notes form the last board meeting. Norm has sent all members a letter about the merger. Here are the details:

- A) That everything is subject to 2/3 majority vote.
- B) That the meeting night would be the third Thursday of each month.
- C) The meeting place would be at the Akron Main Library, subject to the group's needs.
- D) Dues will be due by April 30th of each year. Our dues structure will credit to the 1986 year.
- E) Meeting time will be from 6:00 to 8:45 P.M.
- F) Until May election, each group's officers will co-chair the offices.
- G) Contest for a new name for our combined groups. The winner will receive one year's dues free.

We will still hold the November meeting at the same time and place. The December meeting will be held on the third Thursday at the Akron Main Library in the auditorium.

Think about what was purposed. We will discuss this further at our October meeting.

I would like to thank Russ Cook for his two articles. I would like to thank all the board members of both groups for thier input in this undertaking. See you at the meeting.

Kathi Anderson, Editor

TIPS FROM THE TIGERCUB

#26

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 available as a full disk of 58 programs, routines and files for just \$15.00 postpaid.

>>>>>NOW AVAILABLE<<<<<<<<<<

Tips from the TIGERCUB VOLUME 2 The entire contents of Tips Nos. 15 through 24, with 64 routines and files, also \$15.00 postpaid.

Or both for \$27 postpaid.

>>>>>ALSO NOW AVAILABLE<<<<<<<<<<

New Catalog #6, for \$1 which is deductible from your first order. Describes 148 original programs for only \$3 each (plus \$1.50 per order for cassette or disk, package and postage).

If you have my previous catalog, the following are now available in Extended Basic versions - Fast Addition Practice, Submarine Hunt, Rithmatik, Mawaland (also now available in Basic with Speech), Long Division Cryptograms, Miss Spell, Scramblation, Bargraffer, Squinch, Dry Gulch, Name That Tune, Scrum, Midnight Trail, Nimbo, Kindertimes, Optical Illusion, Bazoo, Synonymy, Speeder Reader, Changeroo, Glunk, Fraction Math, Three Buckets Puzzle, Roman Numbers, Match A

Patch, Kinderminus, I E Spelling, Casting Out Nines, Haunted Graveyard, Spalling Teecher, Homonymy, Antonymy, Old -Timer Puzzle, Ten Thousand Sights, Mechanical Aptitude Test, Junior Speeder Reader, and Bars and Balls.

Due to reduced prices for disks and mailers, the PPM charge is now \$1.50 for either disk or cassette - BUT PLEASE BE SURE TO SPECIFY WHICH!

And my best seller - NUTS & BOLTS, a full disk of 188 (yes, I said 188) utility subprograms in MERGE format, ready for you to merge into your own programs. 13 type fonts, 14 text display routines, 9 wipes, 8 pauses, 3 programming aids, 9 data saving and reading routines, 5 graphics routines, 4 time and date, 6 music, 12 sorts and shuffles, 2 printer aids, 4 key and joystick, 4 math, 2 protection and 7 miscellaneous, plus a tutorial on subprograms. With documentation, example of using each subprogram. All for only \$19.95 postpaid.

Now for the old business -

I was mortified to find an error in the Unprintable Unkeyable Program in Tips #22. The last line should end with ELSE 180, not ELSE 130. In the Grocery Shopping program in Tips #21, your wife will never get to the zucchini unless you delete line 140 and change line 200 to -  
200 IF EOF(1)<>1 THEN 130

Sorry about that. And the update to the Menu Loader in Tips #22 will not list all listable files, just D/V80 files. I now have a version to really list all listable files, I think, plus show protection,

dump the catalog to the printer, rescan, etc., but am not sure all the bugs are out so will publish it next month.

Folks have been asking why their orders for TI-WRITER COMPANION, mentioned in Tips #22, were being returned unopened, so I called Bill Browning. He said he found he was going broke selling it for \$2.50, but he is now prepared to supply it for \$6.50. Still a bargain, in my opinion.

Barry Ensley warns that when FCTN V is used for a blank in a filename, as mentioned in Tips #25, it is not recognized by the Disk Manager.

In Tips #21, I said that the special characters available on the Gemini printer could not be accessed from TI-Writer. I have since learned that Star Micronics had a valuable feature of their printer in a paragraph of gobbledegoose computerese in the manual. See "Other Function Codes", ESC ">", ESC "=" and ESC "#". In plain English, you can access these codes by CTRL U, FCTN R, CTRL U, SHIFT >, then type the character with an ASCII 128 less than the character you want. In other words, if you want CHR\$(160), hit the space bar (ASCII 32), etc. To get back to the normal character mode, use CTRL U, FCTN R, CTRL U, SHIFT #. Many thanks to David Aragon (San Antonio Area 99ers newsletter, Aug. 1985), who described how to do the same by transliteration.

In Tips #25, I said that a program which had been converted to I/V 254 format by adding REM lines could be converted back to program format by deleting

the REM lines and reSAVing. Well, it usually can - but not always'

I have been receiving inquiries as to whether my programs published in the Tips are public domain programs which can be placed in user group libraries and on BBS's. Well, the copyright notice on this newsletter is really only intended to keep anyone from reprinting it for personal profit. I have always thought that programs published for the purpose of being keyed in should be OK to copy, and I don't intend to claim that "you must own the magazine"! However, a peculiar situation has developed. The short programs which I wrote to give away to promote my other programs, have become the bread and butter of my business! If it was not for the sales of the Tips disk and the Nuts & Bolts disk, I would long ago have gone out of business. So, I would appreciate it if you would exercise some restraint in putting my Tips programs in your libraries or in downloadable form on your BBS.

And I do consider my two Tips disks, as complete collections of programs, to be copyrighted material which should not be placed in libraries for copying.

In the Automatic Mouse Maze in Tips #23, you can improve the maze by adding these lines -  
475:IF (C)20)=\$(X)18)THEN 500  
515:X=X+1  
555:Y=Y+1  
595:X=X+1  
1325 X=#

And the last word - I think - on the challenge to quickly scramble the numbers 1 to 255. Ian Swales sent me, from Belgium, two

routines which beat everyone else - and then sent me two more which beat his first ones! His PEEK version -

```

100 DIM A(255),C(255):: FOR
K=255 TO 1 STEP -1 :: RANDOM
IZE :: CALL PEEK(-31000,B)::
J=INT(B#K/256+1):: C(K)=MAX
(J,A(J)):: A(J)=MAX(K,A(K))::
: NEXT K

```

And see if you can unravel the logic of this truly elegant bit of code!

```

100 DIM A(255):: RANDOMIZE :
: FOR K=255 TO 1 STEP -1 ::
J=INT(RND#K+1):: T=MAX(J,A(J)
):: A(J)=MAX(K,A(K)):: A(K)
=T :: NEXT K

```

So, on to new business -

#### ANNOUNCING

The TI-99/4A TRAVELER  
a magazine-on-disk!

700 sectors of articles  
and programs in each issue  
(SS/SD or DS/DD)!

with contributions by  
Mack McCormick, Ron  
Albright, and many others!

Special pre-publication  
prices - \$30 for 6 issues;  
\$7 for sample issue (first  
issue will be Sept 85)

Send your check now to  
Barry A. Traver, Editor  
835 Green Valley Drive,  
Philadelphia PA 19128  
phone (215) 483-1379

To give you an idea of  
Barry Traver's knowledge of  
our computer, try this one.  
I've figured out the why,  
but I'll have to ask Barry  
to explain the why of the  
why!

```

100 ! LINPUT PUZZLE/BUG by
B.A. Traver
110 ! QUESTIONS? Send SASE
to Barry Traver
120 ! 552 Seville St.
Phila. PA 19128
130 CALL CLEAR :: PRINT "LIN
PUT PUZZLE/BUG": "BY BARRY TR
AVER"
140 PRINT "Can you figure ou
t why your computer will not
obey?"
150 PRINT "Why won't it stop

```

```

when you tell it to?": :
160 LINPUT "Want me to stop?
(YES/NO)":M$
170 IF M$="YES" THEN STOP EL
SE 160
180 END

```

It seems that many of  
you still haven't heard of  
Super 99 Monthly, published  
monthly (and on time!) by  
Bytemaster Computer  
Services, 171 Mustang  
Street, Sulphur, LA 70663,  
for \$12 per year. The May  
issue contained a Word  
Processor Dump, to dump a  
graphics/text screen into a  
D/V80 file which can be  
printed out of the TI-Writer  
Formatter - that program  
alone is worth the annual  
subscription price!

I've said it before,  
there is more than one way  
to skin that poor cat. This  
is my routine to alternate  
between the #1 and #2  
joysticks.

```

Z=Z+1/(Z=2)*2 :: CALL (JOYST
(Z,X,Y)
Compact, isn't it? Now, the  
Reading-Berks 99ers publish  
a newsletter called "A Byte  
of Info", which is hardly  
more than a byte long, but  
the August byte was a  
mouthful! Check this -
100 Z=2
110 Z=1/Z#2 :: CALL JOYST(Z,
X,Y)
And this! Elegant!
Z=Z#2 :: CALL JOYST(Z+2,X,Y)

```

Here is another of  
those programs that write a  
program. This one will read  
a screen of graphics and/or  
text and convert it into a  
RUNable program of DISPLAY  
AT statements which will  
recreate the screen.

First, we need a file  
of the hex codes of all the  
normal characters, to check  
against to see if any have  
been redefined. Rather than  
key in all 95 of the  
16-digit codes, let's write

```

a program to write a program
of them -
110 OPEN #1:"DSK1.HEXCODES",
VARIABLE 163 :: LN=30000 ::
FOR D=32 TO 124 STEP 8 :: FO
R CH=D TO D+7 :: CALL CHARPA
T(CH,CH#)
120 D$=D#CHR$(179)&CHR$(200
)&CHR$(16)&CH# :: NEXT CH
130 PRINT #1:CHR$(INT(LN/256
))&CHR$(LN-256#INT(LN/256))&
CHR$(147)&SEB$(D$,2,LEN(D$))
&CHR$(#):: LN=LN+1 :: D$=""
:: NEXT D
140 PRINT #1:CHR$(255)&CHR$(
255):: CLOSE #1 :: END

```

RUN that to create a  
MERGE format program of DATA  
statements. Now, key in the  
GRAFWRITER program -

```

31000 SUB GRAFWRITER
31001 OPEN #1:"DSK1.P6",OUTP
UT,DISPLAY ,VARIABLE 163
31002 RESTORE 30000 :: L=300
00 :: GOSUB 31018
31003 FOR CH=32 TO 127 :: CA
LL CHARPAT(CH,CH#):: READ A$
:: IF CH#=#A$ THEN 31004 ELS
E GOSUB 31019 :: GOSUB 31018
31004 NEXT CH
31005 FOR CH=128 TO 143 :: C
ALL CHARPAT(CH,CH#):: IF CH#
=#RPT$(#$,16) THEN 31006 ELSE
GOSUB 31019 :: GOSUB 31018
31006 NEXT CH
31007 PRINT #1:L$&CHR$(157)&
CHR$(200)&CHR$(5)&"CLEAR"&CH
R$(#):: GOSUB 31018
31008 FOR R=1 TO 24
31009 M$=L$&CHR$(162)&CHR$(2
40)&CHR$(183)&CHR$(200)&CHR$(
LEN(STR$(R)))&STR$(R)&CHR$(
179)
31010 FOR C=3 TO 30 :: CALL
GCHAR(R,C,6):: CALL HCHAR(R,
C,42):: IF F=# AND 6=32 THEN
31013
31011 F=1 :: IF FF=1 THEN 31
012 ELSE CC=C-2 :: FF=1
31012 A$=A$&CHR$(6)
31013 NEXT C :: IF CC=# THEN
CC=1 :: A$=""
31014 PRINT #1:M$&CHR$(200)&
CHR$(LEN(STR$(CC)))&STR$(CC)
&CHR$(182)&CHR$(181)&CHR$(19
9)&CHR$(LEN(A$))&A$&CHR$(#)
31015 L=L+10 :: F,FF,CC=# ::
M$,A$="" :: GOSUB 31018 ::
NEXT R
31016 PRINT #1:L$&CHR$(134)&

```

```

CHR$(201)&L$&CHR$(#):: GOSUB
31018
31017 PRINT #1:CHR$(255)&CHR
$(255):: CLOSE #1 :: SUBEXIT
31018 L1=INT(L/256):: L2=L-2
56#L1 :: L$=CHR$(L1)&CHR$(L2
):: L=L+10 :: RETURN
31019 PRINT #1:L$&CHR$(157)&
CHR$(200)&CHR$(4)&"CHAR"&CHR
$(183)&CHR$(200)&CHR$(LEN(STR
$(CH)))&STR$(CH)&CHR$(179)&
CHR$(199)&CHR$(16)&CH#&CHR$(
182)&CHR$(#):: RETURN
31020 SUBEND

```

Next, Enter MERGE DSK1.  
HEXCODES to merge in those  
DATA statements. Then save  
the program by SAVE  
DSK1.GRAFWRITER,MERGE

Now, load any program  
which has a screen you would  
like to copy. Run the  
program to the point where  
the screen display is ready,  
then break it with FCTN 4.  
Put in a temporary line  
going to itself, such as  
1001 GOTO 1001, and run the  
program again to be sure you  
found the right place. Then  
replace that temporary line  
with CALL GRAFWRITER :: STOP

Put in the disk  
containing the Grafwriter  
program and enter MERGE  
DSK1.GRAFWRITER. Then RUN  
the program. When it stops,  
type NEW, then MERGE DSK1.P6  
and then RUN!

Now for a Tigercub chall-  
enge that I can't answer!  
Can one of you assembly  
programmers tell me how to  
PEEK out of Extended Basic  
for screen color and charac-  
ter set colors, so I can  
reproduce them in that  
program?

And, thanks to Jerry Glaze  
in the Southern Nevada U6  
newsletter, by way of the  
Tidewater newsletter - you  
don't need SIZE with DISPLAY  
AT - just a semicolon!  
100 DISPLAY AT(12,1):RPT\$(#\*  
",2B):: DISPLAY AT(12,1):"SE  
E?";

MEMORY FULL! - Jim Peterson



```

ING 250:NN+3
470 DISPLAY AT(X+6,1):" C
hoice?" :: ACCEPT AT(X+6,16)
SIZE(-3)VALIDATE(DIGIT):K
480 IF FLAG=1 THEN 500
490 IF K=NN+2 THEN 840 ELSE
IF K=NN+3 THEN CLOSE #1 :: N
M=0 :: GOTO 190
500 IF K<>NN AND K<>NN+1 THE
N 590
510 IF K=NN THEN CALL CLEAR
:: CLOSE #1 :: END
520 DISPLAY AT(X+5,12)SIZE(1
2):" #?" :: ACCEPT AT(X+5,15
)SIZE(2)VALIDATE(DIGIT):KD :
: IF KD<1 OR KD>NN THEN 520
530 IF V(KD,1)>0 THEN 550
540 FOR J=1 TO 10 :: DISPLAY
AT(11,1):" " : " PROTECTED -
CANNOT DELETE": " " :: DISPL
AY AT(12,1):" " :: NEXT J ::
GOTO 570
550 DISPLAY AT(X+6,1)SIZE(27
)BEEP:" Verify - Delete ";P6
$(KD);" " :: DISPLAY AT(X+6,
28)SIZE(1):"Y" :: ACCEPT AT(
X+6,28)SIZE(-1)VALIDATE("YN"
):Q$ :: IF Q$>"Y" THEN 570
560 DELETE D$&P6$(KD)
570 CLOSE #1
580 CALL VCHAR(1,3,32,672)::
NN=0 :: X=0 :: FLAG=0 :: GO
TO 260
590 IF K<1 OR K>127 OR LEN(P
6$(K))=0 THEN 430
600 IF ABS(V(K,1))=5 OR ABS(
V(K,1))=4 AND V(K,2)=254 THE
N 640
610 DISPLAY AT(12,1)ERASE AL
L:"Print to ? S": "(P)rinte
r?":(S)creen?" :: ACCEPT AT
(12,12)SIZE(-1)VALIDATE("PS"
):Q$ :: IF Q$="S" THEN PP=0
:: GOTO 630
620 DISPLAY AT(12,1)ERASE AL
L:"PRINTER? PIO" :: ACCEPT A
T(12,10)SIZE(-18):P$ :: OPEN
#3:P$ :: PP=3
630 CALL CLEAR :: CALL SCREE
N(16):: ON ABS(V(K,1))GOTO 6
80,690,750,760
640 CLOSE #1 :: IF SEG$(P6$(
K),LEN(P6$(K)),1)!="*" THEN D
ISPLAY AT(12,1)ERASE ALL:"RE
TURN TO BASIC AND LOAD BY:"
TYPING DLD ";D$&P6$(K):: STO
P
650 CALL PEEK(-31952,A,B)::
CALL PEEK(A#256+B-65534,A,B)
:: C=A#256+B-65534 :: A=D$&

```

```

P6$(K):: CALL LOAD(C,LEN(A$)
)
660 FOR I=1 TO LEN(A$):: CAL
L LOAD(C+I,ASC(SEG$(A$,I,1))
):: NEXT I :: CALL LOAD(C+I,
0)
670 CALL VCHAR(1,3,32,672)::
CALL SCREEN(8):: FOR S=0 TO
14 :: CALL COLOR(S,2,1):: N
EXT S :: DISPLAY AT(12,2):"L
OADING ";A$ :: GOTO 900
680 OPEN #2:D$&P6$(K),INPUT
,FIXED :: GOTO 700
690 OPEN #2:D$&P6$(K),INPUT
700 LINPUT #2:M$ :: PRINT #P
P:M$ :: IF EOF(2)THEN 730
710 CALL KEY(0,K,S):: IF S=0
THEN 700
720 CALL KEY(0,K2,S2):: IF S
2<1 THEN 720 ELSE 700
730 CLOSE #1 :: CLOSE #2 ::
PRINT " >>>press any key<<
" :: IF Q$="P" THEN CLOSE #
3
740 CALL KEY(0,K,ST):: IF ST
<1 THEN 740 ELSE 580
750 OPEN #2:D$&P6$(K),INPUT
,INTERNAL,FIXED :: J=0 :: GO
TO 770
760 OPEN #2:D$&P6$(K),INPUT
,INTERNAL :: J=0
770 IF EOF(2)=1 THEN 730 ::
J=J+1 :: INPUT #2:M$ :: IF L
EN(M$)=0 THEN 790
780 PRINT #PP:M$ :: GOTO 820
790 FOR Y=1 TO 8 :: @E=ASC(
SEG$(M$,Y,1)):: IF @E<32 OR @
E>127 THEN 810
800 NEXT Y :: GOTO 780
810 RESTORE #2 :: FOR X=1 TO
J-1 :: INPUT #2:M$ :: NEXT
X :: INPUT #2:M :: PRINT #PP
:M
820 CALL KEY(0,K,S):: IF S=0
THEN 770
830 CALL KEY(0,K2,S2):: IF S
2<1 THEN 830 ELSE 770
840 DISPLAY AT(24,1):"PRINTE
R NAME? PIO" :: ACCEPT AT(24
,15)SIZE(-14):PP$ :: OPEN #2
:PP$ :: PRINT #2:SEG$(D$,1,4
)&" - Diskname= "&M$
850 PRINT #2:RPT$("*",28):"A
vailable";358-VT;"Used";VT
:RPT$("**",28)
860 PRINT #2:"FILENAME SIZE
TYPE":RPT$(" ",28)
870 FOR P=1 TO NN-1 :: PRINT
#2:P6$(P);TAB(15);V(P,3);TA
B(20):T$(ABS(V(P,1)));TAB(25

```

```

);V(P,2):: NEXT P :: CLOSE #
2
880 DISPLAY AT(12,3)ERASE AL
L:"(P) to print again":(R
) to rescane":(Q) to quit"
890 ACCEPT AT(15,4)VALIDATE(
"PQR")SIZE(-1)BEEP:Q$ :: IF
Q$="P" THEN 840 :: CLOSE #1
:: NN=0 :: IF Q$="R" THEN 19
0 ELSE END
900 RUN "DSKX.1234567890"

This version turns off
the Quit key, restarts
itself rather than crashing
on an I/O error, and has
pre-scan for faster
start-up. It displays disk
name, sectors available and
sectors presumably used - it
also totals up actual
sectors used and sounds a
warning if any sectors are
not accounted for.

It lists up to 127
programs and files by
number, filename, number of
sectors, program or file
type, file record length,
and write-protection. It
will stop for menu selection
on any keypress or at the
end of each screen,
continuing on Enter. It
will load and run any
program that can run from
Extended Basic, displaying
its filename while loading.
If the filename ends in an
asterisk, it will warn you
to return to Basic. It will
delete any unprotected
program or file, after first
requiring verification by
filename, or will inform you
if the file is protected.
It will read any readable
file, including internal
numeric, and list it to
screen or printer. It will
dump a catalog of the disk
to your printer, and it will
offer the option of quitting
or rescanning the disk or
another disk. And it's
free, I don't even want a
freeware donation - but I
would appreciate if you
would take a look at my
catalog and see if,

```

```

somewhere among those 140
programs, there might be
something you would be
willing to pay $3 for? The
Menu Loader is included as a
bonus on every disk I sell'

100 CALL CLEAR :: RANDOMIZE
:: DISPLAY AT(3,4):"TIGERCUB
MATH PUZZLE"
110 DISPLAY AT(6,1):"Insert
+, -, * (multiply) OR / (div
ide) between the digits
to equal the total": "Type
0 to give up"
120 DISPLAY AT(12,1):"Level
1 or 2?" :: ACCEPT AT(12,15)
VALIDATE("12"):L$
130 T,X=INT(9#RND+1):: M$=ST
R$(X):: Z$=M$&" "
140 FOR J=1 TO 4 :: Y(J)=INT
(9#RND+1):: Z=INT(4#RND+1)::
ON Z GOSUB 240,250,260,270
:: Z$=Z$&STR$(Y(J))&" " :: N
EXT J
150 IF L$="1" AND T<>INT(T)T
HEN 130 :: Z$=Z$&" "&STR$(T)
160 DISPLAY AT(12,1):Z$ :: D
ISPLAY AT(18,1):" " :: DISPL
AY AT(20,1):" " :: DISPLAY A
T(22,1):" "
170 P=2 :: FOR J=1 TO 4 :: A
CCEPT AT(12,P)VALIDATE("Q+*
/")SIZE(1):S$
180 IF S$="Q" THEN 200 ELSE
IF S$="+" THEN X=X+Y(J)ELSE
IF S$="-" THEN X=X-Y(J)ELSE
IF S$="*" THEN X=X*Y(J)ELSE
X=X/Y(J)
190 P=P+2 :: NEXT J :: IF X=
T THEN 230 :: DISPLAY AT(18,
1):"WRONG!"
200 DISPLAY AT(20,1):"ANSWER
IS ";M$
210 DISPLAY AT(22,1):"PRESS
ANY KEY"
220 CALL KEY(0,K,ST):: IF ST
<1 THEN 220 :: GOTO 130
230 DISPLAY AT(18,1):"RIGHT!
" :: GOTO 210
240 M$=M$&" "&STR$(Y(J)):: T
=T+Y(J):: RETURN
250 M$=M$&"-"&STR$(Y(J)):: T
=T-Y(J):: RETURN
260 M$=M$&"*"&STR$(Y(J)):: T
=T*Y(J):: RETURN
270 M$=M$&"/"&STR$(Y(J)):: T
=T/Y(J):: RETURN

Enjoy!

Jim Peterson

```

# A HANDY DANDY TI-WRITER USERS REFERENCE GUIDE

SUBMITTED BY BOB STEPHENS

The following handy TI-WRITER commands are reprinted for the June issue of the 99'er News published by the TI Users Group of Will County, Romeoville, Il. This puts the most used commands on one page for handy access at your computer.

```

=====
EDITOR COMMAND |FCTN|CTRL| EDITOR COMMAND |FCTN|CTRL| EDITOR COMMAND |FCTN|CTRL|
=====
Back tab      |   |   | T | Ins. Blank line | 8 |   | O | Quit |   | = |   |
Beginning/line |   |   | V | Insert character | 2 |   | G | Reformat |   |   | 2orR
Command/escape |   | 9 | C | Last paragraph |   | 6orH| Right arrow |   | D |   | D
Delete character |   | 1 | F | Left arrow |   | S | S | Roll down |   | 4 |   | A
Del. end of line |   |   | K | Left margin rel. |   |   | Y | Roll up |   | 6 |   | B
Delete line |   | 3 | N | New page |   | 9orF| Screen color |   |   | 3
Line #'s(on/off) |   | 0 |   | New paragraph |   | 8orM| Tab |   | 7 |   | I
Down arrow |   | X | A | Next paragraph |   | 4orJ| Up arrow |   | E |   | E
Duplicate line |   |   | S | Next window |   | 5 |   | Word tab |   | 7orW
Home cursor |   |   | L | Oops! |   | 1orZ| Word wrap/fixed |   |   | 0
=====
Load files: LF (enter) DSK1.FILENAME (load entire file)
             LF (enter) 3 DSK1.FILENAME (merges filename with data in memory
             after line 3)
             LF (enter) 3 1 10 DSK1.FILENAME (lines 1 thru 10 of filename are
             merged after line 3 in memory)
             LF (enter) 1 10 DSK1.FILENAME (loads lines 1 thru 10 of filename)
=====
Save files: SF (enter) DSK1.FILENAME (save entire file)
            SF (enter) 1 10 DSK1.FILENAME (save lines 1 thru 10)
=====
Print Files:PF (enter) PIO (prints control characters and line numbers)
             PF (enter) C PIO (prints with no control characters)
             PF (enter) L PIO (prints 74 characters with line numbers)
             PF (enter) F PIO (prints fixed 30 format)
             PF (enter) 1 10 PIO (prints lines 1 thru 10)
NOTE: The above assumes PIO, DSK1.FILENAME, and RS232 are also valid!
      To cancel the print command press FCTN 4.
=====
Delete file:DF (enter) DSK1.FILENAME
=====
Setting Margins and Tabs: (16 tabs maximum)
      L - Left margin      R - Right margin      I - Indent      T - Tab
      Use ENTER to execute or COMMAND/ESCAPE to terminate command.
=====
Recover Edit: RE (enter) Y or N
=====
Line move: M (enter) 2 6 10 (moves lines 2 thru 6 after line 10)
           M (enter) 2 2 10 (moves line 2 after line 10)
=====
Copy:      same as move except use C instead of M.
=====
Find String: FS (enter) /string/ (will look for string in entire file)
            FS (enter) 1 15 /string/ (will look for string in lines 2 thru 15)
=====
Delete:    D (enter) 10 15 (deletes lines 10 thru 15 in memory)
=====

```



# REVIEW OF DISKETTE SECTOR LAYOUT

By Homer Crabtree

I know you all understand what is in each sector of a diskette. I am writing this only for reference.

## BYTE | DESCRIPTION OF SECTOR ZERO

0 - 9	DISK NAME ( Unused spaces will contain >20)
10 - 11	NUMBER OF INITIALIZED SECTORS   (SSSD=>160, SSDD or DSDD=>200, DSDD=>5a0)
12	NUMBER OF SECTORS PER TRACK   (Single density=>09, double density=>18)
13 - 15	"DSK" This means that the disk has been init
16	PROTECTION >50=protected, >20 not Protected.
17	NUMBER OF TRACKS PER SIDE
18	NUMBER OF SIDES INITIALIZED
19	DENSITY >01=single, >02=double
20 - 55	NOT USED (yet)
56- 255	SECTOR IN USE BIT MAP   56 - 100 are used for SSSD (360 sectors)   56 - 145 for SSDD or DSDD (720 sectors)   56 - 235 for DSDD (1440 sectors)

## DESCRIPTION OF SECTOR ONE

Sector one is a table with two byte entries. These entries contain the sector numbers of the file descriptors. The order they are placed in this table is the same order they appear in a catalog.

Bytes >00->01 contains the sector number of the first name in a catalog.

Bytes >02->03 contains the sector number of the second name in a catalog.

Bytes >04->05 etc.

.....

.....

There is enough room in this table for 127 entries. Therefore a maximum of 127 files can be stored on each diskette.

## MORE ON DISKO

I thought Advaced Diagnostics was the only way to see fractured files. DISKO option 2 is like FF (find file). Use this option to compute the DATA CHAIN POINTERS and OFFSETS.

Make the modification reported in August BYTEMONGER so you can scroll above sector 360, if you have double sided disk drives.

[ Editors note: Our thanks to Homer for his capable and thorough presentation on diskette layout at the August meeting. For the few of us who don't always remember everything, Homer has provided these handy notes. For completeness, the modification is included below.

Load the OP code file into the E/A Editor and go to record #97. The numbers following the 3rd 'B' tag should be >0167. Change these to >02CF (720 base 10) or >059f (1440 for DSDD). Then change the '7' tag (the 6th from the last non-blank character on that line ) to an '8' so that the checksum won't be checked.

Select the SAVE option of the Editor, respond N to the 'Variable-length?' prompt and save the file (name different than original). You should now be able to load that file and access all of the sectors with the FCTN PROCEED key.

## DESCRIPTION OF DIRECTORY ENTRY

NOTE: File descriptor, file header, directory entry, catalog entry are used interchangeably, and contain the following:

### BYTE | DESCRIPTION

0 - 9	NAME OF FILE
10 - 11	NOT USED (all zeros)
12	FILE ATTRIBUTES (bit#) (if on) (if off)     0   PROGRAM DATA FILE   1   INTERNAL DISPLAY   2   RESERVED RESERVED   3   PROTECTED NOT PROT.   4-6   RESERVED RESERVED   7   VARIABLE FIXED
13	NUMBER OF RECORDS PER SECTOR=INT(256/REC LN)   >00 FOR PROGRAMS
14 - 15	NUMBER OF ADDITIONAL SECTORS USED
16	NUMBER OF BYTES IN LAST RECORD (variable only)
17	RECORD LENGTH
18 - 19	NUMBER OF FIXED LENGTH RECORDS or   NUMBER OF SECTORS USED BY VARIABLE LENGTH FILE   NOTE: bytes are reversed. >0500 = >0005
20 - 27	NOT USED (all zeros)
28- 256	DATA CHANGE POINTERS   These three byte entries point to blocks   of data that make up the file, and   contains a count of sectors used thus far.   The second byte is made up of two nibbles   and should be arranged as follows:   If the first entry was >22 >60 >00,     2 2 6 0 0 0   —     —                     Highest sec.           offset from   v v v v v beginninf of   SECTOR-> 022 006 <-file.

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

