



JULY 1985 Vol. 3 No. 7

The July meeting will be held on Thursday, July 18th 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 begins at 7:30 PM. Please remember to sign in.

The program this month is a Fun and Swap Night. Everyone should bring their equipment to get the maximum enjoyment out of your computer.

We will need members to serve on the nominating committee for the coming elections. Anyone interested in serving on the committee or would like to hold an office, speak up at the coming meeting. In order to vote in the September elections you must be a current member.

JUNE WINNERS

| | |
|----------------|--------------------|
| Scott Chrisman | Cassette Box |
| Dominic Sedits | Joystick Extension |
| Norm Sorkin | 10 Dicks |
| Dan Fedak | Joystick "Y" |
| Russ Cook | Cassette Interface |

CALENDAR OF EVENTS:

July 18 Meeting- Fun and Swap Night
July 25 Board Meeting
August 15 Meeting- Advantages of Extended Basic over Basic
August 22 Board Meeting
September 20 Meeting- Elections
September 27 Board Meeting

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 |

TIPS FROM THE TIGERCUB

#24

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) I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

TIJ
UDS:TDFIFA
BJIF
JDIF
SURS
STSA
SFBF
TRA

The above is a long division problem in the proper format, with each numeral replaced by a letter. Can you solve it?

My program TC-41 Long Division Cryptograms, will generate an infinite number of such puzzles for you, and help you to solve them - and it only costs \$3.00. It took me a week to program, and I've sold 12 copies in 2 years! Doesn't anyone like to exercise their brains anymore?

TIGERCUB CHALLENGE

```
100 FOR J=1 TO 7 :: READ M$
:: PRINT M$ :: NEXT J
30000 DATA AAAAAAAAAAAAAAAAAA
AAAAAAAAAAA,BBBBBBBBBBBBBB,BB
BBBBBBBBBBBB,CCCCCCCCCCCCCC,
DDDDDDDDDDDDDD
30010 DATA "TESTING",,,,,,,
,,,,,,,""TEST
ING""
>RUN
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBB,BBBBBBBBBBBBBB
CCCCCCCCCCCCCC
DDDDDDDDDDDDDD
"TESTING"
,,,,,,,""TESTING""
$ READY $
```

Can you run this program and get these results? You won't even be able to key in that last DATA item! So, how was this programmed? No, there are no redefined characters!

Do you need something educational? Here is a little routine to give the plural endings for most words. I will leave it to you to develop further - and see if you can teach the computer the plurals of PANTS, TOOTH, MAN, FUNGUS, DATA and the other inconsistencies of the English language.

```
100 REM PLURAL ENDINGS
    by Jim Peterson
110 INPUT W$
120 Z$=SE6$(W$,LEN(W$),1)
130 Y$=SE6$(W$,LEN(W$)-1,2)
140 ON POS("EFHSXYZ",Z$,1)+1
    GOTO 270,150,190,180,250,25
    0,220,250
150 IF SE6$(W$,LEN(W$)-2,2)<
    >"IF" THEN 270
160 PL$=SE6$(W$,1,LEN(W$)-2)
    &"VES"
170 GOTO 280
180 IF (Y$="CH")+(Y$="SH")TH
    EN 250 ELSE 270
190 IF (Y$<>"LF")$(Y$<>"RF")
    $(Y$<>"AF")$(W$<>"HDOF")THEN
    270
200 PL$=SE6$(W$,1,LEN(W$)-1)
    &"VES"
210 GOTO 280
220 IF (Y$="AY")+(Y$="EY")+
    (Y$="OY")+(Y$="UY")THEN 270
230 PL$=SE6$(W$,1,LEN(W$)-1)
    &"IES"
240 GOTO 280
250 PL$=W$&"ES"
260 GOTO 280
270 PL$=W$&"S"
280 PRINT PL$
290 GOTO 110
```

If you want to turn that into a quiz, change line 110 to READ W\$, change line 280 to PRINT W\$;" PLURAL?":. Add lines - 281 INPUT Q\$ 282 IF Q\$<>PL\$ THEN 285 283 PRINT : "RIGHT!": 284 GOTO 110 285 PRINT : "WRONG! PLURAL OF ";W\$;" IS ";PL\$: 300 DATA BOX,WATCH,WIFE,BOY (And as much more as you want)

Just one more optional refinement to my Menu Loader. If you want to use a filename ending in an asterisk for those Basic programs which will not run in XBasic, this change will keep you from loading and crashing them.

```
420 CLOSE #1 :: IF SE6$(P6$(K),LEN(P6$(K)),1)="#" THEN D
    ISPLAY AT(12,1)ERASE ALL:"RE
```

TURN TO BASIC AND LOAD BY:" TYPING OLD DSK1.&P6\$(K):: S TOP

The idea of a program that writes a program has stirred up a little interest, so here's another. This routine will aid you in formatting your screen text into neat 28-column lines, and will save the text in program lines of DATA statements. When you are ready to save, type @@@ and enter as the last line, then NEW and MERGE DSK1.LINEFILE

100 !LINERITER

```
- by Jim Peterson
130 CALL CLEAR :: OPEN #1:"D
    SK1.LINEFILE",VARIABLE 163 :
    : LN=30000
140 FOR R=1 TO 24 :: DISPLAY
    AT(R,1)SIZE(1):" " :: ACCEP
    T AT(R,0)SIZE(-2B):A$ :: IF
    A$="@@" THEN 180 :: B$=B$&C
    HR$(200)&CHR$(LEN(A$))&A$
150 X=X+1 :: IF X/4=INT(X/4)
    THEN 160 ELSE B$=B$&CHR$(179
    ):: GOTO 170
160 GOSUB 210 :: LN=LN+10
170 NEXT R :: X=0 :: CALL CL
    EAR :: GOTO 140
180 IF B$="" THEN 200 :: IF
    SE6$(B$,LEN(B$),1)=CHR$(179)
    THEN B$=SE6$(B$,1,LEN(B$)-1)
190 GOSUB 210
200 PRINT #1:CHR$(255)&CHR$(
    255):: CLOSE #1 :: END
210 PRINT #1:CHR$(INT(LN/256
    ))&CHR$(LN-256*INT(LN/256))&
    CHR$(147)&B$&CHR$(0):: B$=NU
    L$ :: RETURN
```

Here's something for "JET" and Danny and Gene and all the rest of my friends in Alabama - and in all the rest of Dixie. You've never seen fireworks quite like these before!

```
100 CALL CLEAR :: PRINT TAB(
    5);"ALABAMA 4th of JULY": :
    : : : : "programmed by
    Jim Peterson" :: FOR D=1 TO
    200
110 NEXT D :: RANDOMIZE
120 DIM S$(12),A$(16),S(16),
```

```

SX$(15)
130 DATA 196,220,247,262,294
,330,349,392,440,494,523,587
,659
140 FOR J=4 TO 16 :: READ S(
J):: NEXT J :: FOR SET=2 TO
14 :: CALL COLOR(SET,1,1)::
NEXT SET :: CALL SCREEN(2)
150 DATA 00,10,24,3C,42,5A,6
6,7E,81,99,AS,B0,C3,DB,E7,FF
160 FOR J=1 TO 16 :: READ A$(
J):: NEXT J
170 FOR CH=40 TO 136 STEP 8
:: FOR L=1 TO 4 :: X=INT(16*(
RND+1)):: B$=B$&A$(X):: C$=A$(
X)&C$ :: NEXT L
180 SX$(CH/8-4)=B$&C$ :: CAL
L CHAR(CH,B$&C$):: GOSUB 350
:: B$=NULL$ :: C$=NULL$ :: NE
XT CH
190 FOR J=1 TO 12 :: FOR L=1
TO 6 :: X$=CHR$(INT(13*(RND+
5))&8):: B$=B$&X$&X$ :: C$=X$
&X$&C$ :: NEXT L
200 S$(J)=B$&C$ :: B$=NULL$
: C$=NULL$ :: NEXT J :: CALL
MAGNIFY(2)
210 FOR J=1 TO 12 :: DISPLAY
AT(J,3):S$(J):: NEXT J :: X
=1 :: FOR J=13 TO 24 :: DISP
LAY AT(J,3):S$(J-X):: X=X+2
:: NEXT J :: CALL DELSPRITE(
ALL):: FOR D=1 TO 200 :: NEX
T D
220 DATA 1,11,7,1,9,7,2,7,4,
2,4,7,1,7,4,1,4,8,1,4,9,1,4,
10,2,11,7,2,7,11,2,11,7,2,9,
4
230 DATA 2,12,5,2,5,12,3,12,
7,1,11,7,3,12,5,1,11,7,1,12,
5,1,13,4,1,14,5,1,15,10
240 DATA 6,16,7,1,14,9,1,11,
7,6,14,4,1,11,7,1,9,4,6,11,6
,1,8,6,1,9,7,6,7,4
250 DATA 1,11,7,1,13,4,2,14,
9,2,16,11,3,15,4,1,14,9,2,12
,10,4,14,10,2,12,7,6,15,10,2
,12,8
260 DATA 6,15,6,1,11,6,1,13,
4,2,14,9,2,16,14,3,15,11,1,1
4,9,2,12,10,2,13,7,3,14,10,1
,12,10
270 DATA 2,11,7,2,9,4,3,14,9
,1,9,5,2,9,4,4,8,4,2,9,4,6,7
,4,2,9,4,6,8,4,2,12,5
280 DATA 2,11,7,2,9,4,3,14,7
,1,16,7,2,15,10,4,14,9,2,9,4
,6,7,4,2,9,4,6,8,4,2,12,10
290 DATA 2,11,7,2,9,4,3,16,1
1,1,14,9,2,15,4,2,14,7,2,14,

```

```

9,6,14,11
300 FOR N=1 TO 96 STEP 3 ::
READ T,A,B :: CALL COLOR(A-2
,A-2,1):: CALL COLOR(B-2,B-2
,1):: FOR TT=1 TO T :: CALL
SOUND(-999,S(A),0,S(B),5)::
NEXT TT
310 CALL COLOR(A-2,1,1):: CA
LL COLOR(B-2,1,1)
320 NEXT N :: RESTORE 220 ::
FOR N=1 TO 252 STEP 3 :: RE
AD T,A,B :: CALL COLOR(A-2,A
-2,1):: CALL COLOR(B-2,B-2,1
):: FOR TT=1 TO T :: CALL SO
UND(-999,S(A),0,S(B),5):: ME
XT TT
330 CALL COLOR(A-2,1,1):: CA
LL COLOR(B-2,1,1)
340 NEXT N :: FOR J=5 TO 30
:: CALL SOUND(-999,S(A),J,S(
B),J):: NEXT J :: RESTORE 22
0 :: FOR CH=40 TO 136 STEP 8
:: GOSUB 350 :: NEXT CH ::
GOTO 190
350 CALL MAGNIFY(1):: CALL S
PRITE(0CH/8-4,CH,13*(RND+3,20
0,128,-30,RND*20-RND*20):: R
ETURN

```

The Home Computer Magazine, Vol. 4 No. 3, had a program called Elementary Addition and Subtraction, which generates random numbers between 1 and 5 for elementary math practice.

The first time I tried it, it asked me for the answer to 1 + 1. When I answered correctly, it produced another random problem - 1 + 1 again!

This is known as the idiotic computer syndrome, and it helps us to remember that our computers are still no smarter than their programmers!

Fortunately, this bit of idiocy is easy to cure. Try this -

```

100 RANDOMIZE
110 X=INT(5*(RND+1))
120 IF X=X2 THEN 110
130 X2=X
140 PRINT X;
150 GOTO 110

```

Do you see how it works? The first time you

get a number, X2 will equal 0 because it has never been given a value. X will be selected as a number between 1 and 5. Let's suppose it is 2. Line 120 compares it with X2; 2 is not equal to 0, so the program continues to line 130, where X2 now picks up the value of 2, then on to print the value, and back to 110. Now, suppose that the random factor in line 110 picks 2 again. Line 120 finds that 2=2, X=X2, and sends the program back to 110 to pick a different number.

If you want to avoid a repeat until after two times, change line 120 to read 120 IF (X=X2)+(X=X3)THE N 110 and add a line 125 X3=X2 .

For a longer series without repeating, it might be better to use this method.

```

100 A$="ABCDEFGHJIJ"
110 FOR J=1 TO 10
120 RANDOMIZE
130 Y=INT(RND*LEN(A$)+1)
140 X=ASC(SEG$(A$,Y,1))-64
150 A$=SEG$(A$,1,Y-1)&SEG$(A
$,Y+1,LEN(A$))
160 PRINT X
170 NEXT J
180 GOTO 100

```

That will give you a random series of 1 through 10 and then repeat with a different random series. Adjust the number of letters in the string A\$, and the corresponding "TO" value in 110, for whatever you require.

Several newsletters recently have published articles on the "program that you never run" - because it consists entirely of REM statements!

For instance, you can keep a list of the members of your users group, using their membership number for the program line number,

followed by REM (or ! in Basic) and their name and address. For a printed list, just LIST the program to the printer. To change someone's address, or to delete a deadbeat who doesn't pay his dues, just edit the program. You can also LIST the program to disk to create a DIS/VAR 80 file which you can then load into TI-Writer and use its editing features, FindString, etc.

The same method can give you a tickler file, or appointment calendar, which is just as good as some rather complex disk filing programs written for this purpose. Just use the month number (1-12) and date (always in two digits, 01-31) for the line number -

```

1000 !buy birthday pres
sent for wife!
1009 !wife's birthday!
1010 !apologize to wife for
forgetting birthday

```

You can schedule several things in one program line -

```

1011 !get haircut/change oil
in car/pinch secretary.....
- but it might be better
to add an extra digit (0-9)
to the line number and
schedule separately -

```

```

10110 !get haircut
10111 !change oil

```

Then, if something doesn't get done, just use the REDO key to change the line number and reschedule it for another date. You can print out a list of the day's chores by simply LIST "PI0:7010-7019 (did you know you could do that?)

MEMORY FULL IN LINE 470

- Jim Peterson

CARTRIDGE SCREEN DUMP

This is the **Screen Dump and Switch** I told you about at the last couple of meetings. If you recall I mentioned that I had heard of a screen dump that would also dump a cartridge's screen by just pushing a button. WELL THIS IS IT Thanks to the **Houston Users Group (HUG)** and their March newsletter.

I called Texas and talked to **Cecil** about the hardware instalation and **Danny Michaels** Screen Dump program.

We also talked their **TIBBS** system. I've not loqged on there yet, but more on that later. Lets get back to the screen dump.

Cecil was very helpful and what we have to show you is due to his help.

I must note that the interrupt switch only works with the **Danny Michael** Screen Dump.

You will get a lot more information from the documentation that is on the disk you get with the screen dump.

IF YOU HAVE ANY PROBLEMS GIVE ME A CALL. I'LL TRY TO HELP IF I CAN. I got mine working in less than 1/2 an hour.

NORM

LIBRARY NOTES

SCREEN DUMP

HUG Library #1380

This program is an assembly language screen dump which will dump BASIC, Extended BASIC and COMMAND MODULE screens to your printer. It requires Memory Expansion and the XB-Module for XB screens and the E/A Basic for BASIC screens. For dumping Command Module screens, an interrupt switch is required (see article in this newsletter). The interrupt switch may also be used for dumping the other types of screens as well.

The program comprises 330 sectors of disk space and includes the source codes and 4 pages of printed documentation. It will dump either normal size, double size (6XB), rotation of 90 degrees, tab settings for spacing and inverted image. The documentation is easy to understand and gives you samples of CALL LINK statements to use to get the different types of printout. The program takes only about a minute to dump a complete screen to your printer.

The only drawbacks I see with this program is that it does not have shadowing (the colors are either black or white) and it will not print sprites (no screen dump will do that at this point). The program was written by **Danny Michael** of Florence, Alabama and I must congratulate him on such a great program. Anyone who has a printer capable of screen dumps should have this program in their library.
-Bill Knecht

For those who have tried **BANNER (2000-2)** from the library, there is a revised version, **BANNER II**. **BANNER** is limited to one size letter—**BIG**—and on my printer paused 4-6 seconds between each line. **BANNER II** eliminates the pauses and lets you vary the height and width of letters. Didn't have time and printer paper to test all variations so let me know of any problems and I'll try to fix it.

Another program, **BANNERMAKER**, has been added. It will make letters from 5/8 inch to 6 1/2 inches high.

If you tried to play **BATTLESTAR (1000-10)** and had problems, try again. Some corrections have been made.

An assembly language disassembler written in **TI-BASIC** has been added. It requires a Mini-memory module and can list from the Mini-memory ROM and RAM as well as the **TI99/4A** memory.

Russ Cook

This article comes to us from 99 HOCUS, June, 1985 from the Milwaukee area.

```
SCORE      145
MEN        2
HIGH SCORE 209
```



FORTH Editing Tricks

Did you know about the DOPS key in the Forth Editors? In case you ever tried to insert, using the FCTN 2 key and mistakenly hit FCTN 3 and erased the entire line, don't despair, FCTN 8 (insert line) brings it right back since it's still sitting there in memory. This usefull key can be utilized for moving lines, duplicating lines and even portions of lines, even to other screens. Using FCTN 3 to erase a line or FCTN 7 to erase a portion of a line, saves it in a memory buffer from which FCTN 8 will return it. I've included a little Forth game I wrote to utilize some speech and arcade sound routines from our Forth data disks. It'll run even if you don't have the Speech Synthesizer but the computer just won't talk to you. You can test the editing key on this program and eliminate a whole screen. Do it in this order:

```
75 78 SCOPY <cr>
76 75 SCOPY <cr>
78 76 SCOPY <cr> -this switches lines 75 & 76
```

now use FCTN 3 (erase) on line 1 Scr 77
 FCTN 6 to return to Scr 76 cursor to line 12
 FCTN 8 (insert) and the line appears here
 use the same method to move lines 2, 4 & 5 from Scr 77 to lines 13 to 15 Scr 76 and you've eliminated Scr 77
 Always be careful however inserting lines, so you don't wipe out something important on line 15 that disappears in this case the --> is no longer needed. When using the FCTN 8 to insert a line from the memory buffer, it doesn't forget the line. You can continue to insert the line to your heart's content. Thus if you need a number of similar lines, just erase the one and then insert as many as you need before going back and making the minor editing changes in each. To copy a line, merely erase it and re-insert it. Since it's still in memory you can then copy it any place as often as you wish.

In BODY-SNATCHERS your gun emplacement should move horizontally back and forth, if it begins to descend off the screen when moving to the left, you evidently have one of the 'buggy screens' versions of Forth. I've seen 4 different versions already. Correct lines are:
 Screen 58 lines 9 & 10
 9- VOPMOZ @ 4 < IF SMTN @ 0 VFILL 300 SATR ! ENDIF
 10-- GATR 20 @ DO DUP >R D000 SP@ R) 2 VMBW DROP 4 + LOOP DROP
 Screen 59 Line 9
 9--- 4 & SMTN + >R @ SLA SWAP @0FE' AND DR SP@ R) 2 VMBW DROP ;

```
SCR # 70
0 ( BODY-SNATCHERS Scr 1 by Gene Hutz )
1 BASE->R HEX
2 : SPEAK 40 9400 C! 50 9400 C! ;
3 : s 4 @ DO 4 SRC DUF 4 SRC @F00 AND 4000 OR SWPB 9400 C!
4 LOOP SFEAR DROP ;
5 : "OVER" 2E29 s 4E0A s 7170 s 556E s 71BE s 4786 s 145C s
6 4FE2 s 1745 s 1
7 : "NICE" 49A5 s 71BE s 2365 s 2B1E s ;
8 : "HIGH" 7DD8 s 71BE s 2FFC s 35AE s 1BAC s 35AE s ;
9 : "GET" 487D s 30D6 s 379C s 31A0 s 71BE s ;
10 : "GOOD" 30FA s 30 @ SPRPAT 7:5E s 34 @ SFRPAT
11 31A0 s 38 @ SPRPAT 4C7C s 3C @ SPRPAT ;
12 : "PLEA" 5093 s 5231 s 26AD s 145C s 637C s ;
13 : "LOTR" 71F4 s 61C6 s 57E5 s 3A32 s 664E s 4ADA s ;
14 : wait 400 @ DO NOP LOOP ;
15 -->
```

```
SCR # 71
0 ( BODY-SNATCHERS Scr 2 )
1 8400 CONSTANT SOUND @ VARIABLE CMD DECIMAL
2 : FCODE >R 111B60B. R> M/ SWAP DROP 10 / ; HEX
3 : TONE FCODE DUP @F AND CMD C! 4 SRL @3F AND CMD 1+ C!
4 CASE 1 @F @20 ENDOF 2 @F @A0 ENDOF 3 @F @C0 ENDOF ENDCASE
5 CMD @0 DR CMD C! CMD 1+ @0 SOUND CMD @0 SOUND C! C! ;
6 : VOL @F AND CMD C! CASE 1 @F @90 ENDOF 2 @F @B0 ENDOF
7 3 @F @D0 ENDOF ENDCASE CMD @0 OR DUP CMD C! SOUND C! ;
8 DECIMAL
9 : UP 330 220 DO 1 1 TONE 4 +LOOP ;
10 : DD 330 220 DO 1 550 1 - TONE 4 +LOOP ;
11 : SLI 1 1 VOL 5 @ DO UP DD LOOP 1 15 VOL ;
12 : ALA 1 1 VOL 5 @ DO UP DD LOOP 1 15 VOL ;
13 : SPA 2 274 TONE 3 276 TONE 2 2 VOL 3 2 VOL SLI
14 2 15 VOL 3 15 VOL ; ; AT GC70XY ;
15 -->
```

```
SCR # 72
0 ( BODY-SNATCHERS Scr 3 )
1 HEX @ VARIABLE SCO @ VARIABLE HYS @ VARIABLE FLG
2 @ VARIABLE MEN A VARIABLE FAL @ VARIABLE SHL RANDOMIZE
3 : SS0 2000 SEET 2 MAGNIFY A SCREEN C 1 10 COLCR 1 1 4 COLOR
4 AAFF FFFF FFFF FFFF B0 C-#F @ 16 20 B0 MCHAR 3 #MOTION
5 0101 0303 0703 0707 20 SPCHAR 0707 0707 0505 21 SPCHAR
6 0000 0000 0000 0000 22 SPCHAR 0000 0000 0000 23 SPCHAR
7 1F12 3F7F FFFF 77F7 24 SPCHAR F737 0800 0800 25 SPCHAR
8 FB4B FCFE FFFF EEEF 26 SPCHAR EFEC 1000 1000 27 SPCHAR
9 0000 000B 0404 070F 28 SPCHAR 3F7F DEFF FFBF 29 SPCHAR
10 0000 0020 0404 00E0 2A SPCHAR FBFC 5FE FEFA F206 2B SPCHAR
11 0404 0505 0703 0303 2C SPCHAR 0302 0206 0000 2D SPCHAR
12 2020 A0A0 E0C0 00C0 2E SPCHAR C040 4060 0000 2F SPCHAR
13 1F12 0040 DECC 51C6 30 SPCHAR 645D 1903 0B10 31 SPCHAR
14 FB4B 0002 780B 8A63 32 SPCHAR 26BA 98C0 101B F00B 33 SPCHAR
15 -->
```

```
SCR # 73
0 ( BODY-SNATCHERS Scr 4 )
1 1F12 0040 CCC0 50C2 34 SPCHAR C01D 1802 001B 0710 35 SPCHAR
2 FB4B 0002 3C0B 0A43 36 SPCHAR 03B8 1840 001B E00B 37 SPCHAR
3 1B10 0000 40F4 00C0 38 SPCHAR 8010 1800 0010 010C 39 SPCHAR
4 DB0B 0000 0229 0003 3A SPCHAR 0108 1800 000B E0C2 3B SPCHAR
5 1110 0000 00C0 00B0 3C SPCHAR 8000 2030 0000 0026 3D SPCHAR
6 BB0B 0000 0003 0001 3E SPCHAR 0100 040C 0000 0054 3F SPCHAR
7 A0 A0 F 2B 1 SPRITE @ @ 1 MOTION 4 2 AT ." SCORE " SCO ?
8 4 3 AT ." MEN " MEN ? ;
9 : CLR @ @ AT CLS ;
10 : MT> 70 RND 30 - FAL @ RND 2+ @ MOTION ;
11 : SS1 A0 2 D 24 @ SPRITE 70 RND 30 - 1 @ MOTION SPA MT ;
12 : ?HI SCO @ HYS @ OVER < IF HYS < "HIGH" wait ELSE
13 "NICE" wait THEN 4 4 AT ." HIGH-SCORE "
14 HYS ? ;
15 -->
```

```

SCR # 74
0 ( BODY-SNATCHERS Scr 5 )
1
2 : G4 ?HI A C AT ." FINI" "OVER" wait wait TEXT ABORT ;
3 : G5 FFFF MEN +! A 3 AT MEN ?
4   MEN @ @= IF G4 THEN ;
5
6 : G2 @ SPRGET SWAP DROP ;
7
8 : G1 @ @ @ MOTION @ SPRGET F + @ 2C 2 'SPRITE
9   @ F@ @ MOTION @ F@ 2 MOTION BEGIN 62 10 < END 2 DELSPR G5 ;
10
11 : HYT @ SPRGET SWAP DROP A@ > IF "GDT" G1 SSI THEN ;
12 -->
13
14
15

```

```

SCR # 75
0 ( BODY-SNATCHERS Scr 6 )
1 : Gd 1 SML +! SML @ A = IF @ SHL ! "LOTR" G5 THEN ;
2 : Gc 2 SPFGT SWAP DROP @ SPRGET SWAP DROP > ;
3 : Gb @ SPFGT B@ SWAP - SCD +! DROP @ 2 SPRCOL @ @ SPRCOL
4   @ @ 2 MOTION 1 1 2 SPRPUT @ @ @ MOTION wait ALA @ FLG !
5   "GOOD" A 2 AT SCD ? wait ;
6 : Ga 2 @ @ @ COINC IF Gb SSI THEN ;
7 : G9 HONK @ @ 1 MOTION 1 SPRGET A 2@ 2 SPRITE 1 FLG !
8   @ 9@ 2 MOTION SEE:N Gc WHILE Ga REPEAT 2 DELSPR FLG @ 1 =
9   IF Gd @ FLG ! THEN ;
10 : G8 @ @ AT 1 JOYST DROP @ @ @ 1 MOTION 12 = IF G9 THEN ;
11 : GAM SSI BEGIN 5 FAL +! 2@ @ DO MT> G8 HYT LOOP AGAIN ;
12
13
14 -->
15

```

```

SCR # 76
0 ( BODY-SNATCHERS Scr 7 )
1 : DOCU 8 2 AT ." BODY-SNATCHERS" CR CR
2   " ARCADE ACTION SOFTWARE " CR CR
3   " Move your gunner with the arrow keys "
4   " or joysticks (1) and fire with '0' "
5   " or the fire button." CR CR
6   " You have 5 gunners to stop the alien "
7   " Body-Snatchers from body-snatching." CR CR
8   " Everyone getting thru snatches a body "
9   " and costs you a gunner." CR CR
10  " Every 10 misses also costs you a one, "
11  " so don't waste your shots." CR CR
12  " The higher the alien when you hit him "
13  " the higher the points scored." CR CR
14  " Press Enter To Start" "PLEA" wait CR KEY D = IF
15  CLR ELSE ABORT THEN ; -->

```

```

SCR # 77
0 ( BODY-SNATCHERS Scr 8 )
1 : RLN CLR DOCU @ SCC ! 5 MEN ! @ SHL ! A FAL !
2   GRAPHICS S2@ DECIMAL GAM ;
3
4 RUN
5 R->BASE
6
7
8
9
10
11
12
13
14
15

```

This article comes to us from Pug Peripheral by way of Akron Area TI Users Group, June, 1985.

LOTTERY

Listed below is a short program, for picking 6 random numbers. The problem with random number generators is that they often pick duplicate numbers. This program does away with this problem. Lines 150-170 stores numbers 1-40 in B\$ array, lines 220-240 is a normal random number generator, line 250 checks to see if the number chosen is a null string in B\$ array, line 270 replaces a chosen random number with a null string in B\$ array. No two numbers selected are the same in a set of 6 numbers.

By Tom Hare

```

100 REM RANDOM NUMBER GENERATOR FOR 6 NUMBERS 1 TO 40
110 DIM B$(40)
120 CALL SCREEN(12)
130 CALL CLEAR
140 RESTORE 180
150 FOR A=1 TO 40
160 READ B$(A)
170 NEXT A
180 DATA 1,2,3,4,5,6,7,8,9,10
190 DATA 11,12,13,14,15,16,17,18,19,20
200 DATA 21,22,23,24,25,26,27,28,29,30
210 DATA 31,32,33,34,35,36,37,38,39,40
220 RANDOMIZE
230 FOR I=1 TO 6
240 R=INT(RND*40)+1
250 IF B$(R)="" THEN 240
260 PRINT R;" ";
270 B$(R)=""

```

Continued on page 10.

TIPS FROM THE TIGERCUB

#25

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) I will send you my descriptive catalog for a dollar, which you can then deduct from your first order.

Many of the users groups are taking a summer break, so I thought I would do the same. I'm going to mail out the July and August issues of the Tips in June (imagine, a TI publication

AHEAD of schedule!!) and then go fishing. However, if anyone should by any chance decide to send me an order during the summer, they will still get my same-day service.

It seems that I had better clear up a few misunderstandings. The "freeware" offers I have mentioned in past Tips are NOT available from me - send your disk and returnable mailer AND RETURN POSTAGE to the author of the program.

And, my copyrighted Tigercub Software programs are NOT freeware. They can only be legally obtained by mail order from me - if you copy them from anyone else, you are stealing!

As for the programs which I write and publish or distribute without copyright, they are also not Freeware, they are FREE. I don't want to be paid for them, and I don't think anyone else should be paid for them.

Some users groups are putting my copyrighted programs, and those of other programmers, in their software library, "for use but not copying" or "for review and evaluation only". Who do you think you're kidding? I know I won't sell any software to members of pirate clubs, so why should I support them?

If you didn't solve the Long Division Puzzle in Tips #24, try dividing 230709 by 835. As for the solution to the Tigercub Challenge, it was right on the same page! Try creating those DATA statements with the LINewriter routine. I don't know why it works, but it does.

I've been asked to print more information on the "program that writes a

program". I don't have room for a detailed account, but here are the basics. If you tried my TOKENLIST routine in Tips #23 you already have a list of the token codes you will need.

I won't go into the way that the computer squishes a program line number into only two characters, but you can accomplish it with DEF L%=CHR\$(INT(LN/256))&CHR\$(LN-256*INT(LN/256)), where LN has been predefined as the value of the line number.

If you need to refer to a program line in a statement, as in GOTO 500, use DEF R%=CHR\$(201)&CHR\$(INT(RN/256))&CHR\$(RN-256*INT(RN/256)), RN being the line number.

To print a statement or command, simply print its token character. For instance, the token for DATA is 147, so you would print CHR\$(147). Note that all the punctuation marks used in programming, such as (and +, are also represented by token codes which are NOT the same as their keyboard ASCII value.

To print a variable name, either numeric or string, just enclose it in quotes, "A" or "A\$".

To print a value, or an unquoted string (as in a DATA statement), or the word which follows a CALL, you must print CHR\$(200) followed by a token giving the number of characters to follow, such as CHR\$(5) for a 5-character word such as CLEAR, then the value in quotes. For instance, the token for CALL is 157, so CALL CLEAR is CHR\$(157)&CHR\$(200)&CHR\$(5)&"CLEAR".

You can simplify that by predefined DEF U\$(V\$)=CHR\$(200)&CHR\$(LEN(V\$))&V\$, and then simply print CHR\$(157)&U\$("CLEAR").

A quoted string is handled in the same way

except that it is preceded by token 199 instead of 200, so you can predefine it as DEF W\$(V\$)=CHR\$(199)&CHR\$(LEN(V\$))&V\$ - the computer will take care of the quote marks.

Each program line must end with CHR\$(0), and the last record you print must be CHR\$(255)&CHR\$(255).

A MERGE format file is D/V 163, so open the file with OPEN #1:"DSK1.MERGEFILE", VARIABLE 163.

Don't print more than 163 characters in a record or the computer will blow its mind! You can print multiple-statement XBasic lines, but be sure to use the double-colon token CHR\$(130) as the separator, not two of the CHR\$(181) colon tokens.

Any errors you make will usually not show up until you try to MERGE or use the program you have created. I/O ERROR 25 means that you forgot the final 255 & 255; DATA ERROR or SYNTAX ERROR probably means that you left off a CHR\$(0) or gave the wrong count of characters after CHR\$(200).

Here's a bit of psychedelic blues - -

```
100 REM - FRANKIE & JOHNNIE
      by Jim Peterson
110 DIM S(12)
120 CALL SCREEN(2)
130 FOR R=1 TO 12
140 CALL COLOR(R+1,1,1)
150 FOR T=R TO 25-R
160 CALL HCHAR(T,R,32+R*8,34-2*R)
170 NEXT T
180 NEXT R
190 DATA 262,294,311,330,349
    ,392,440,494,523,587,40000
200 FOR N=1 TO 11
210 READ S(N)
220 NEXT N
230 FOR J=1 TO 110 STEP 2
240 CALL COLOR(A+1,1,1)
250 READ T,A
260 CALL COLOR(A+1,A+2,A+2)
```

```

270 FOR TT=1 TO T
280 CALL SOUND(-999,S(A),0)
290 NEXT TT
300 NEXT J
310 RESTORE 330
320 GOTO 230
330 DATA 2,1,2,2,2,4,2,7,1,1
1,1,7,2,6,4,4,2,1,1,13,1
340 DATA 2,1,2,2,2,4,2,7,1,1
1,1,7,2,6,4,4,12,1
350 DATA 1,11,3,1,2,5,2,6,2,
7,2,9,1,11,1,9,2,10,4,7,1,9,
1,11,7,9
360 DATA 4,7,2,8,2,9,1,11,3,
9,1,11,1,9,4,8,2,7,6,6
370 DATA 4,4,1,11,3,4,4,3,16
,2,1,11,4,7,2,6,4,7,4,6,20,1
,8,11

```

You can too have a blank space in your disk filenames! Just use FCTN V for the blank, instead of the space bar. You can even have a diskfull of 10 programs with invisible filenames consisting of 1 to 10 of those FCTN V's.

However, those invisible characters can do strange things when you list your disk catalog to a printer.

If you want to INPUT a string with leading and/or trailing blanks, just enclose the whole works in quotation marks. Try this -

```

100 INPUT A$ !type TEST
110 PRINT A$;LEN(A$)
120 INPUT A$ !type " TEST "
130 PRINT A$;LEN(A$)
140 GOTO 100 !you can even
input a blank string of 136
characters

```

I really shouldn't tell you this, but if you want to make it difficult for someone to LIST your program, just insert a garbage line, every 5th line or so until you run out of memory, consisting of REM followed by 4 or 5 lines of random characters typed with the CTRL key held down.

Here's a program that

```

can actually read your mind!
100 CALL CLEAR
110 PRINT "TIGERCUB MIND READER PROGRAM":
120 PRINT "I'll bet you a dollar I can guess what you are thinking.":
130 GOSUB 440
140 PRINT "And I'll bet another dollar I can tell if what you are thinking is correct.":
150 GOSUB 440
160 PRINT "And I'll bet another dollar I'm right BOTH times.":
170 GOSUB 440
180 PRINT "And I'll bet one more dollar I can guess what you'll be thinking a minute from now.":
190 GOSUB 440
200 PRINT "OK....":
210 GOSUB 480
220 PRINT "You're thinking that a computer can't possibly know what you are thinking.....right?":
230 GOSUB 480
240 PRINT "So I told you what you were":"thinking.....right?":
250 GOSUB 480
260 PRINT "You owe me a buck.":
270 GOSUB 480
280 PRINT "And you're absolutely right..I can't read your mind.":
290 GOSUB 480
300 PRINT "So I told you correctly that":"what you were thinking was":"correct.....right?":
310 GOSUB 480
320 PRINT "You owe me another buck.":
330 GOSUB 480
340 PRINT "So I was right BOTH times...right?":
350 GOSUB 480
360 PRINT "That makes three bucks you owe me.":
370 GOSUB 480
380 PRINT "And now it's a minute later":"and you're thinking you've":"been played for a sucker....":"...right?":

```

```

390 GOSUB 480
400 PRINT "...so you owe me four bucks.":
410 GOSUB 480
420 PRINT "NEVER NEVER bet against a computer!! "
430 END
440 PRINT "Want to bet? Type Y(Yes)":
450 CALL KEY(3,K,ST)
460 IF (ST=0)+(K<>89)THEN 450
470 RETURN
480 FOR D=1 TO 800
490 NEXT D
500 RETURN

```

Since the manual doesn't mention it, some folks don't know that you can use IMAGE and PRINT USING for output to the printer. Try this -

```

100 OPEN #1:"PIO"
110 INPUT "NAME? ":N$
120 INPUT "AMOUNT? ":A
130 PRINT #1,USING "#####
#####"
.##":N$,A
GOTO 110

```

Of course, you could also add a line -

```

105 IMAGE "#####
####.##"

```

And change line 130 to

```

130 PRINT #1,USING 105:N$,A

```

John Taylor has written the most complete and versatile SPRITE BUILDER utility program that I have ever seen. It has 22 different options available with a single key press, including rotation and animation. And along with it comes a diskfull of preprogrammed sprites designed by a professional artist. This is being distributed as Freeware. Send two single-sided or one double-sided disks to John Taylor, 2170 Estaline Drive, Florence AL 35630, in a returnable mailer WITH RETURN POSTAGE, at least - and I hope you'll also include something more!

Attention, assembly programmers! Fred Hawkins of the Lehigh UG is trying to coordinate a project of documenting the operating system by breaking the console ROM down to pages of 256 bytes so that each individual or group can work on just one page. Only those who participate will share in the results! All this is far beyond me, but if you want in, send an SASE and a SSSD disk with return postage and mailer to Fred Hawkins, 1020 N 6th St, Allentown PA 18102 - soon!

If you have a program on disk which is so long that you must type CALL FILES(1) before you can load it, add several program lines to it consisting of REM and any key you want to hold down for 5 lines. Then SAVE it back to the disk; it will now be in INT/VAR 254 format and will load without CALL FILES(1). If you then need sometime to make a cassette copy, just delete those lines and SAVE it back to disk again.

If a program loads, but gives you a MEMORY FULL IN LINE ... when you try to run it, it has used up all available memory while reading DATA into arrays or performing other internal calculations. If it runs for some time and then gives you the MEMORY FULL message, it is because you have repeatedly jumped out of a FOR...NEXT loop with an IF...THEN...GOTO before the loop is completed. This rarely happens but it can, especially when you repeatedly jump out of the innermost of several nested loops.

MEMORY FULL

Jim Peterson

This article comes to us from 99 HOCUS, June, 1985 from the Milwaukee area.

DON'T HESITATE TO TRANSLITERATE!

Many members of our User's Group and other TI-99/4A owners with TI-WRITER avoid using the Transliterate Command like it was a disease or a foreign language. As thick as the TI-WRITER Manual is, which intimidates some, this command is certainly one subject that deserves further explanation. These tips and applications are probably not the only points to be made about transliteration, and I hope others will come about as a result of this article. To keep it short, no examples are included, but contact me at the meeting if you have any questions.

1) Contrary to several published reports, the Transliterate Command does indeed work, and works well if each instance of its use is on a separate line (with a carriage return afterward). The most common reason seen for failure of this command is the lack of the leading period, which is required for all format commands. Also, you must use commas, not spaces, when a multiple-character "conversion" is needed. One user even gets it to change printer configuration (instead of the SPECIAL CHARACTER MODE, which does not seem to work), but he uses the ampersand "&" between the multiple characters needed.

2) Use the Transliterate Command to transliterate a character to itself if you want to cancel that conversion later in your text.

3) Be aware that any time you use the Transliterate Command to cause one character to produce a set of symbols all at one time, if you are using the Fill and Indent Mode, the Text Formatter will treat those symbols as one character! As examples, consider the sequence given on page 107 of the TI-WRITER Manual to print a tilde over an "n" or the case of using one character to produce a set of ellipses marks (for a partial quotation).

4) The instance described in item #3 can be used to your advantage when you have run out of positions when composing a Header or Footer --- just use one "complex" Transliterate Command to specify the additional spaces or other characters needed.

5) The Transliterate Command can also be used to increase the spacing between a Header or Footer and the text and/or the end/start of the page, by using a line feed appropriately in the Header or Footer, preceded by a proper Transliterate Command. However, in this instance, do not use underline or overstrike symbols in your text.

6) Another caution to keep in mind is not to convert any of the "reserved" characters used by TI-WRITER for its own special functions (the caret "^", the ampersand "&" or the at-sign "@"), as well as the underline. You will not get any error message, but your results will be strange!

7) The suggestion by the TI-WRITER Manual at pages 106 and 127 to use two at-signs or ampersands together when you want to print one of these symbols did not seem to work. Only transliterating did the trick.

8) I haven't tested this for the entire set of available characters, but it appears that the Transliterate Command will work with any valid ASCII code as its first parameter (even ASCII 0 to 31), although the second parameter should be a character your printer is capable of printing/processing.

9) It's a good idea to use the Find String Command to check your document for instances where you may have used a character in the text before you use that same character in a transliteration.

-----Abdallah Clark



Cuyahoga Falls, Ohio 44223

P.O. Box 3201

SUMMIT 99ers USERS GROUP

```
280 NEXT I
290 PRINT
300 PRINT "PRESS ENTER FOR ANOTHER SET FCTN <CLEAR> TO STOP"
310 PRINT
320 CALL KEY(O,X,STATUS)
330 IF X=13 THEN 140
340 IF STATUS=0 THEN 320
```