TIPS FROM THE TIGERCUB

No. 65

Tigercub Software
156 Collingwood Ave.
Columbus, OH 43213

****

My three Nuts & Bolts disks, each containing 100 or more subprograms, have been reduced to $5.00. I am out of printed documentation so it will be supplied on on disk.

My TI-PD library now has well over 500 disks of fairware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename, Basic programs converted to XBasic, etc. The price is just $1.50 per disk(!), post paid if at least eight are ordered. TI-PD catalog #5 and the latest supplement is available for $1 which is deductible from the first order.

It is a bit of a nuisance to have to hit Enter after inputting a single character such as Y or N for "yes" or "no". CALL KEY accepts a single character without Enter, but has no blinking cursor to tell you that it is waiting. I should have had this one in my Nuts & Bolts years ago - the CALL KEY WITH CURSOR subprogram! R is the row, C is the TAB position, V$ is the validation string, such as "YyNn", and the character selected is returned in K$.

```
30000 SUB CALLKEY(R,C,V$,K$)
30001 CALL HCHAR(R,C+2,30)::
 FOR T=1 TO 3 :: CALL KEY(0,
K,S):: IF S<>0 THEN 30004
30002 NEXT T :: CALL HCHAR(R
,C+2,20):: FOR T=1 TO 3 :: C
ALL KEY(0,K,S):: IF S<>0 THE
N 30004
30003 NEXT T :: GOTO 30001
30004 IF POS(V$,CHR$(K),1)=0
 THEN 30001 ELSE K$=CHR$(K)
30005 SUBEND
```

And for a demonstration of the use of that subprogram, here is a little game that no one will ever play to the *end* -

```
100 DISPLAY AT(3,6)ERASE ALL
:"THE ULTIMATE TEST":"":" An
swer the question with a num
ber according to whether the
 number or color shown,"
110 DISPLAY AT(8,1):"or the
note sounded, was 1stor 2nd
or 3rd, etc."
120 DISPLAY AT(23,6):"PRESS
ANY KEY" :: DISPLAY AT(23,6)
:"press any key" :: CALL KEY
(0,K,SS):: IF SS=0 THEN 120
ELSE CALL CLEAR
130 DATA 2,BLACK,3,GREEN,5,B
LUE,9,RED,12,YELLOW,14,PURPL
E
140 FOR J=1 TO 6 :: READ C(J
),C$(J):: CT$=CT$&CHR$(J)::
N$=N$&CHR$(J+48):: NEXT J ::
 T=2 :: DL=500 :: V$="12"
150 RANDOMIZE :: T$,NN$=CT$
:: FOR J=1 TO T :: X=INT(RND
*LEN(T$)+1):: X$=SEG$(T$,X,1
):: T$=SEG$(T$,1,X-1)&SEG$(T
$,X+1,255):: Y(J)=ASC(X$)
160 X=INT(RND*LEN(NN$)+1)::
X$=SEG$(NN$,X,1):: NN$=SEG$(
NN$,1,X-1)&SEG$(NN$,X+1,255)
:: S(J)=ASC(X$):: NEXT J ::
FOR J=1 TO T
170 Z(J)=INT(89*RND+10):: FO
R K=1 TO J-1 :: IF Z(J)=Z(K)
THEN 170
180 NEXT K :: NEXT J :: CALL
 CLEAR :: CALL COLOR(3,16,1,
4,16,1)
190 FOR J=1 TO T :: CALL SCR
EEN(C(Y(J))):: CALL SOUND(-9
99,110*S(J),0):: DISPLAY AT(
12,12):Z(J):: FOR D=1 TO DL
:: NEXT D :: NEXT J
200 CALL CLEAR :: CALL SCREE
N(16):: CALL COLOR(3,2,1,4,2
,1):: X=INT(3*RND+1):: W=INT
(T*RND+1):: ON X GOTO 210,23
0,210
210 IF X=1 THEN Q$=C$(Y(W))E
LSE IF X=3 THEN Q$=STR$(Z(W)
)
220 DISPLAY AT(12,1):"WHICH
WAS ";Q$ :: GOTO 240
230 CALL SOUND(1,30000,30)::
 DISPLAY AT(12,1):"WHICH WAS
?" :: FOR D=1 TO 200 :: NEXT
 D :: CALL SOUND(500,110*S(W
),0)
240 CALL CALLKEY(12,20,V$,K$
):: Q=ASC(K$)-48
250 IF Q=W THEN DISPLAY AT(1
5,12):"RIGHT!" ELSE DISPLAY
AT(15,12):"WRONG!"
260 IF Q=W THEN DL=DL-50 ELS
E DL=DL+50
270 IF DL<100 THEN DL=500 ::
 T=T+1 :: V$=SEG$(N$,1,T)
280 GOTO 150
290 SUB CALLKEY(R,C,V$,K$)
300 CALL HCHAR(R,C+2,30):: F
OR T=1 TO 3 :: CALL KEY(0,K,
S):: IF S<>0 THEN 330
310 NEXT T :: CALL HCHAR(R,C
+2,20):: FOR T=1 TO 3 :: CAL
L KEY(0,K,S):: IF S<>0 THEN
330
320 NEXT T :: GOTO 300
330 IF POS(V$,CHR$(K),1)=0 T
HEN 300 ELSE K$=CHR$(K)
340 SUBEND
```

I have warned repeatedly over the years, in these Tips and in Micropendium and elsewhere, that printing program listings through the Funlweb Formatter usually results in garbled listings that cannot be keyed in correctly - but I still see the garbled listings published. Here is a fix to the Funlweb FO file that will partially solve the problem -

Boot DSKU. Select 1. File Utilities. Select 5. Find String. Enter filename FO and the drive number. Enter H for hex. Enter the string 2A23214026 . Enter replace string 7C2321605C . When the string is found, enter R for replace, then CTRL W, hit Enter twice to accept the defaults. Thereafter, use FCTN Z instead of & to underline, FCTN C instead of @ to double-strike, and FCTN A instead of % to call a value added file. I don't know why Texas Instruments didn't do that in the first place, and I wonder why the McGoverns didn't make that fix.

Now, can anyone tell me how to replace the ^, which tends to disappear, and the period, which will make the whole line disappear if it happens to be at the beginning of the line?

If you are one of the few who are still interested in recreational computing - the use of the computer to solve puzzles and math problems just for the fun of it - you might be interested in Recreational and Educational Computing, published 8 times a year at 909 Violet Terrace Clarks Summit PA 18411. The annual subscription is $27. Program listings are in dialects of Basic other than TI but usually not hard to convert.

That is where I found this ridiculously short, simple and fast card shuffling routine.

```
100 DIM C(52)
110 FOR X=1 TO 52 :: C(X)=X
:: NEXT X
120 FOR X=52 TO 1 STEP -1 ::
 I=INT(RND*X+1)
130 T=C(I):: C(I)=C(X):: C(
)=T :: NEXT X
```

In the same place, I rea a routine to extract a roo to 16-digit accuracy instea of the 8 digits availabl on a PC from the basic for mula ROOT=NUMBER^(1/POWER) We don't need it - our obs lete 16k 16-bit computer c give us 14-digit accura from the basic formula!

The same publication ga me the idea for this litt game -

```
100 DISPLAY AT(3,6)ERASE A
:"THE GAME OF N":"":"You
 the computer will take
rns adding to a num- ber
```

```
reach a goal."
110 DISPLAY AT(8,1):"If you
reach the goal, you  win. Yo
u get to go first andyou sho
uld be able to win  almost
every time."
120 RANDOMIZE :: N=INT(RND*1
5)+15 :: R=INT(4*RND+3):: S=
R+1 :: D=N-INT(N/S)*S :: T=0
130 DISPLAY AT(13,1):"The go
al is";N;"":"Maximum input i
s";R :: DISPLAY AT(19,1):RPT
$(" ",28*6)
140 DISPLAY AT(17,1):"Your n
umber?" :: ACCEPT AT(17,14)S
IZE(1)VALIDATE(DIGIT):A :: I
F A<1 OR A>R THEN DISPLAY AT
(15,1):"" :: GOTO 130
150 T=T+A :: DISPLAY AT(21,1
):"Total is";T :: IF T=N THE
N DISPLAY AT(23,1):"YOU WIN!
" :: GOSUB 190 :: GOTO 120
160 IF N-T<S THEN P=N-T :: T
=T+P :: DISPLAY AT(19,1):"Co
mputer adds";P :: DISPLAY AT
(21,1):"Total is";T :: DISPL
AY AT(23,1):"COMPUTER WINS!"
 :: GOSUB 190 :: GOTO 120
170 IF T=0 THEN P=D ELSE IF
(N-T)/S=INT((N-T)/S)THEN P=I
NT(R*RND+1)ELSE Y=N-T :: P=Y
-INT(Y/S)*S
180 T=T+P :: DISPLAY AT(19,1
):"Computer adds";P :: DISPL
AY AT(21,1):"Total is";T ::
GOTO 140
190 DISPLAY AT(24,8):"PRESS
ANY KEY" :: DISPLAY AT(24,8)
:"press any key" :: CALL KEY
(0,K,S):: IF S=0 THEN 190 EL
SE T=0 :: RETURN
```

REC also printed a puzzle which seemed so simple that I could not see why. It goes like this -

A game show host shows you three curtains. Behind one is a new car, behind the other two are goats. You choose one. The host, who can peek behind the curtain, opens one of those you did not pick, and shows a goat. Then he offers to let you change your choice. Should you switch, stand pat, or does it make no difference?

You now have a 50-50 bet, so it makes no difference, right? But some very distinguished mathematicians were saying you should switch, so I wrote this computer simulation to prove them wrong. Key it in, run it, and be surprised. Do figures lie? Do computers lie? Is there something wrong with my simulation?

```
100 CALL CLEAR
110 DATA CAR BEHIND,A PICKS,
HOST SHOWS,A WINS,B WINS,C W
INS
120 FOR J=1 TO 3 :: READ M$
:: DISPLAY AT(J,1):M$ :: NEX
T J :: FOR J=12 TO 14 :: REA
D M$ :: DISPLAY AT(J,1):M$ :
: NEXT J
130 FOR J=1 TO 1000 :: RANDO
MIZE :: X=INT(3*RND+1):: DIS
PLAY AT(1,13):X !RANDOMLY PL
ACE CAR
140 A=INT(3*RND+1):: DISPLAY
 AT(2,13):A !PLAYER CHOOSES
150 D=INT(3*RND+1):: IF D=X
OR D=A THEN 150 :: DISPLAY A
T(3,13):D :: ! HOST PICKS CU
RTAIN WITH GOAT
160 IF A=X THEN AA=AA+1 :: D
ISPLAY AT(12,7):AA ! A DOES
NOT SWITCH
170 B=INT(3*RND+1):: IF B=A
OR B=D THEN 170
180 IF B=X THEN BB=BB+1 :: D
ISPLAY AT(13,7):BB ! B SWITC
HES
190 C=INT(3*RND+1):: IF C=D
THEN 190
200 IF C=X THEN CC=CC+1 :: D
ISPLAY AT(14,6):CC ! C CHOOS
ES RANDOMLY
210 NEXT J
```

Here is an improved version of a program that was in a Tips long ago, to strip out the extra blanks from a Filled and Adjusted Funlweb Formatter file -

```
100 DISPLAY AT(3,6)ERASE ALL
:"TIGERCUB UNFILLER";"":" To
 remove extra spaces from":"
a TI-Writer text which has":
"been Filled and Adjusted by
"
110 DISPLAY AT(8,1):"the For
matter, prior to":"reformatt
ing."
120 DISPLAY AT(15,1):"Input
file? DSK" :: ACCEPT AT(15,1
6):IF$ :: OPEN #1:"DSK"&IF$,
INPUT
130 DISPLAY AT(17,1):"Output
 file? DSK" :: ACCEPT AT(17,
17):OF$ :: OPEN #2:"DSK"&OF$
140 LINPUT #1:M$ :: P=1
150 X=POS(M$," ",P):: IF X=P
 THEN P=P+1 :: GOTO 150
160 X=POS(M$,"  ",P):: IF X=
0 THEN PRINT #2:M$ :: GOTO 1
80
170 M$=SEG$(M$,1,X)&SEG$(M$,
X+2,255):: GOTO 160
180 IF EOF(1)<>1 THEN 140 ::
 CLOSE #1 :: CLOSE #2
```

While a program is running, the computer periodically pauses for a fraction of a second to do a "garbage collection", getting rid of information it no longer needs, to make room in memory. If this pause occurs at a critical moment in program execution, it can cause problems. Thanks to the Sydney User Group in Australia, *here is a CALL LOAD which* will force a garbage collection just before that critical point -

```
CALL LOAD(-31885,144,"",-318
58,81,169,152,0)
```

Here is a neat one from Bruce Harrison. Key it in, (you can skip the lines that start with an asterisk) and assemble it, then use ALSAVE to imbed it in any program that opens a disk file. Put CALL LINK("DEVICE",DEV$) at the beginning of the program and change any line reading OPEN #1:"DSK1.FILENAME" - or whatever - to read -

```
OPEN #1:DEV$&".FILENAME"
```

(don't forget the period before the filename!). Now you can load the program from any drive and it will open the file on that same drive!

```
* STRING ASSIGN DEVICE NAME
* PLACES DEVICE NAME IN AN
*  XBASIC STRING
* HARRISON SOFTWARE
* 8 OCTOBER 1990
* FOR USE WITH ALSAVE AND XB
* TAKES ONLY 42 BYTES MEMORY
STRAS6 EQU  >2010
WS     EQU  >20BA
       DEF  DEVICE
DEVICE
* USE OUR WORKSPACE
       LWPI WS
* GET THE CRU BASE IN R12
       MOV  @>83D0,R12
* GET ROM ADDRESS FOR DEVICE
*  IN R2
       MOV  @>83D2,R2
* ENABLE THE ROM
       LDCR @ONES,0
* ADDING 4 PUTS US AT THE
*  LENGTH BYTE
       AI   R2,4
* FIRST PARAMETER
       LI   R1,1
* NOT AN ARRAY VARIABLE
       CLR  R0
* ASSIGN DEVICE NAME TO A
*  STRING
       BLWP @STRAS6
* CLEAR CRU, DISABLE ROM
       LDCR R0,0
* LOAD GPL WORKSPACE
       LWPI >83E0
* RETURN TO GPL INTERPRETER
       B    @>006A
* WORD TO TURN ON ROM IN CRU
ONES   DATA >0101
       END
```

Getting short on memory, so more next time.

Jim Peterson

LETTER HEADER
by Dave Swartz

At the recent Rocky Mountain 99'ers Jamboree I was discussing word processing with a new member (yes, we do have new members) and the point was brought up about using word processor files to hold routine information used frequently, i.e. letterheads and return addresses that are printed on almost every letter we write. My friend said he had never heard of the procedure, but thought it would be a good idea. With his remarks in mind and not knowing how many others there may be who do not know of the procedure, this article has been written to help spread the word. The procedure works equally well with TI Writer or FunnelWeb Writer.

The address data as well as control codes can be encoded so you can set up your format and printer one time and have it available with a few key presses.

Usually, I type documents Emphasized (E) and Double Strike (G) so as to get nice black copy for the copy machine. I use the CTRL U, FCTN R, CTRL U codes to set this up. Another CTRL U code is used to set up 12 characters per inch (M), as I feel this makes a better looking document. I use an Epson LQ-850 printer, so use standard Epson codes.

The first line looks like this on the monitor:

```
0001  'bE'bG'bMcr
```

Next I set up the format of the letterhead or return address I am going to use. The rest of the set up looks like this on the monitor:

```
0002  .AD;FI;LM 8;RM 88;PL 64cr
0003  .CE 4cr
0004  ROCKY MOUNTAIN 99'ERScr
0005  P.O. Box 31846cr
0006  Aurora, CO 80041cr
0007  U.S.A.cr
0008  cr
0009  .IN +56cr
0010  xx Nov 1990cr
0011  cr
0012  .IN +0cr
0013  (Type in address for business
      letter. May be several lines.)cr
0014  cr
0015  Dear xxxxcr
0016  cr
0017  .IN +5 (Normal paragraph indent.)cr
0018  (Text)cr
```

This file is saved as DSKx.HEADING on the disk on which I compose and save my letters.

For personal letters where I want my return address on the right side of the paper, I make changes as shown below:

```
0001  'bE'bG'bMcr
0002  .AD;FI;LM 8;RM 86;PL 64cr
0003  .IN +56cr
0004  (Street address)cr
0005  (City, state and zip)cr
0006  xx Nov 1990cr
0007  cr
0008  .IN +0cr
0009  Dear xxxxcr
0010  cr
0011  .IN +5cr
0012  (Text)cr
```

I have used parameters for 12 characters per inch instead of 10 per inch. If your printer does not have a 12 CPI capability, you should modify the commands for Right Margin and Indent to reflect your printer's capabilities.

With this system, all you have to do is load your D/V 80 "HEADING" file, put in the date and addressee, and start your letter. It saves much time and typing and eliminates errors in addition to formatting your letters.

In lines 0001 of the above programs, the symbol "'b" represents the ESCAPE code. Key in CTRL U, FCTN R, CTRL U, to generate the escape code. Then use the character shown in the printer handbook for the function you wish to employ. The symbol "cr" indicates the CARRIAGE RETURN (ENTER) key should be pressed.

My thanks to Jack Sughrue and his "PLUS!" program which facilitated the printing of the printer control symbols.

THE CYBERNETIC MOTHER GOOSE

Old Mother Hubbard
Went to the cupboard
'Twas disks with old files she went for.
When she got there
The disks were all bare;
A magnet on the cupboard door.

Georgie Porgie, pudding and pie,
Had not backed up since last July.
When at last his disk did crash,
Georgie Porgie got a rash.

Jack and Jill went up the hill
To buy a ream of paper.
"Though it's bland, we'll write by hand,
The software still is vapor."

There was a graphic man and he made a graphic box,
He found a graphic program, he gave up Paradox;
He pulled a graphic menu, to start a graph' routine,
And they interfaced together on a little graphic screen.

Hey, diddle, diddle!
Must you always fiddle?
The estimate's over the moon;
The users still wait to see it work,
And the programmer keeps saying "soon".

High-res mice!
High-res mice!
See how they roll!
See how they roll!
Drag the cursor up to the top line
Pull down a menu; it all works so fine;
A keystroke could do it in half of the time.
High-res mice!

Little Miss Napple sat by her Apple,
Writing routines in C;
"I don't care what that flaw meant;
I'll write not one comment,
So no one can read it but me."

Hickory, dickory, dock!
Let's speed up this thing's clock:
Faster it goes and nobody knows
The CPU's getting a shock.

9T9

```
("THIRTEEN",K3$)ELSE IF K3=1
4 THEN CALL SPGET("FOURTEEN"
,K3$)ELSE IF K3=15 THEN CALL
 SPGET("FIFTEEN",K3$)
380 IF K3=16 THEN CALL SPGET
("SIX",K3$)ELSE IF K3=17 THE
N CALL SPGET("SEVEN",K3$)ELS
E IF K3=18 THEN CALL SPGET("
EIGHT",K3$)
390 IF K3=19 THEN CALL SPGET
("NINE",K3$)ELSE IF K3>15 TH
EN CALL SPGET("TEEN",T$):: K
3$=K3$&T$
400 IF K3>9 THEN 410 ELSE CA
LL SPGET(STR$(K3),K3$)
410 IF K<>K3 THEN 420 ELSE C
ALL SAY("#GOOD WORK#, THAT I
S RIGHT.....NOW"):: GOTO 210
420 CALL SAY("UHOH.THAT IS N
```

```
50
430 CALL SAY("#GOODBYE#")::
CALL CLEAR :: STOP
440 SUB DEFS1(A$)! NUMBERS
450 DATA 96,0,26
460 DATA 14,56,130,204,0
470 DATA 223,177,26,224,103
480 DATA 85,3,252,106,106
490 DATA 128,95,44,4,240
500 DATA 35,11,2,126,16,121
510 RESTORE 450
520 A$=""
530 FOR I=1 TO 29 :: READ A
:: A$=A$&CHR$(A):: NEXT I
540 SUBEND
550 END
```

==========

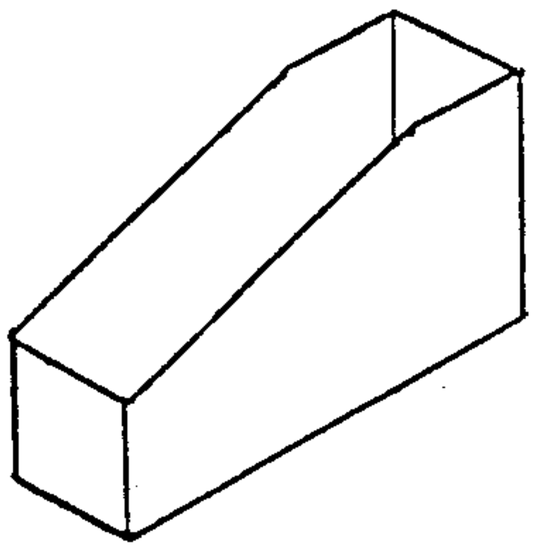
The PUNN Newsletter - Portland, OR - June 1991

HOW TO MAKE A DISKETTE CASE

by PHIL VAN NORDSTRAND8
JOHNSON SPACE CENTER USERS GROUP

Do you have stacks of disks sitting around, some grouped with rubber bands?

Possibly you have fancy plastic cases but they don't always solve the problem of disk storage and organization. I have two plastic cases that hold more than 50 disks, but I save them for master disks and others that I don't ever use, leaving a problem of how to store the rest - the ones I want to be able to find in a hurry.



The solution I came up with is to make simple storage cases from empty dry food containers. I have one box for my TIPS disks, one for my GENIAL TRAVelER disks, one for my PR-BASE disks, and one for my TI-WRITER file disks, etc. They are a light weight, scaled down version of the magazine holders advertised at over $3 each in an office supply catalog.

The boxes I use are about 5-5/8" deep and 2-1/4" wide. They hold about 20 disks and are made from Waverly cracker boxes. I also have one made from a Bisquick box that is slightly deeper.

They are made by cutting down the cardboard boxes to a height of about 4 inches. You can leave the sides straight and horizontal or you can be more elegant by curving the two wide sides or sloping them down to about 3 inches high in front.

To make them look neat and hide the advertising, cover the sides with contact paper. I use the imitation wood grain paper, but anything goes.

I have also made cases for magazines and soft cover computer manuals from 9 inch boxes and cases for small software booklets from 6-1/2" deep boxes.

SPEECH AND SUBTRACT
IN EXTENDED BASIC
by R.W. AUGUST

This program will help your children learn subtraction. It ask for the answer and gives the correct answer if entered wrong. The program will run in extended basic and is enhanced with speech synthesizer, but is not necessary for the program to run. Enjoy!!

```
100 ! SPEAK AND SUBTRACT
110 ! IN EXTENDED BASIC
120 ! BY R.W. AUGUST
130 CALL DEFS1(Z$):: CALL SP
GET("NUMBER",L$):: L=LEN(L$)
-L-3 :: S$=SEG$(L$,1,2)&CHR$
(L)&SEG$(L$,4,L):: NUM$=S$&Z
$
140 DISPLAY AT(4,3)REASE ALL
:"<< SPEAK AND SUBTRACT >>"
:: DISPLAY AT(8,1):"HELLO, I
 LIKE TO WORK WITH": :"NUMBE
RS.  DO YOU?"
150 CALL SAY("HELLO.I+LIKE+T
O+WORK+WITH",NUM$,"DO+YOU")
160 DISPLAY AT(13,1):"OK, I
WILL GIVE YOU THE": :"NUMBER
S AND YOU ENTER THE": :"ANSW
ER."
170 CALL SAY("O+K,I+WILL+GIV
E+YOU+THE",NUM$,"AND+YOU+ENT
ER+THE+ANSWER"):: DISPLAY AT
(22,1):"PRESS ENTER WHEN REA
DY"
180 CALL SAY("PRESS+ENTER,WH
EN+RED+D")
190 CALL KEY(0,K,S):: IF K<>
13 THEN 190
200 FOR I=8 TO 22 :: CALL HC
HAR(I,1,32,32):: NEXT I
210 RANDOMIZE :: K1=INT(RND*
21):: K2=INT(RND*21):: IF K1
>K2 THEN 210 :: IF K1>9 THEN
 230 ELSE CALL SPGET(STR$(K1
),K1$)
220 IF K2>9 THEN 280 ELSE CA
LL SPGET(SRT$(K2),K2$):: GOT
O 330
230 IF K1=10 THEN CALL SPGET
("TEN",K1$)ELSE IF K1=11 THE
N CALL SPGET("ELEVEN",K1$)EL
SE IF K1=12 THEN CALL SPGET(
"TWELVE",K1$)
240 IF K1=13 THEN CALL SPGET
("THIRTEEN",K1$)ELSE IF K1=1
4 THEN CALL SPGET("FOURTEEN"
,K1$)ELSE IF K1=15 THEN CALL
SPGET("FIFTEEN",K1$)
250 IF K1=16 THEN CALL SPGET
("SIX",K1$)ELSE IF K1=17 THE
N CALL SPGET("SEVEN",K1$)ELS
E IF K1=18 THEN CALL SPGET("
EIGHT",K1$)
260 IF K1=19 THEN CALL SPGET
("NINE",K1$)ELSE IF K1=20 TH
EN CALL SPGET("TWENTY",K1$)
270 IF K1<16 OR K1=20 THEN 2
20 ELSE CALL SPGET("TEEN",T$
):: K1$=K1$&T$ :: GOTO 220
280 IF K2=10 THEN CALL SPGET
("TEN",K2$)ELSE IF K2=11 THE
N CALL SPGET("ELEVEN",K2$)EL
SE IF K2=12 THEN CALL SPGET(
"TWELVE",K2$)
290 IF K2=13 THEN CALL SPGET
("THIRTEEN",K2$)ELSE IF K2=1
4 THEN CALL SPGET("FOURTEEN"
,K2$)ELSE IF K2=15 THEN CALL
 SPGET("FIFTEEN",K2$)
300 IF K2=16 THEN CALL SPGET
("SIX",K2$)ELSE IF K2=17 THE
N CALL SPGET("SEVEN",K2$)ELS
E IF K2=18 THEN CALL SPGET("
EIGHT",K2$)
310 IF K2=19 THEN CALL SPGET
("NINE",K2$)ELSE IF K2=20 TH
EN CALL SPGET("TWENTY",K2$)
320 IF K2<16 OR K2=20 THEN 3
30 ELSE CALL SPGET("TEEN",T$
):: K2$=K2$&T$
330 CALL SAY("WHAT+IS",K2$,"
TAKE A+WAY",K1$):: DISPLAY A
T(12,1):"WHAT IS ";K2;" TAKE
 AWAY";K1 :: K3=K2-K1
340 DISPLAY AT(15,9):K2;"  -
  ";K1;" =" :: DISPLAY AT(24
,3):"** ANSWER ""S"" TO STOP
 **"
350 ACCEPT AT(15,25)SIZE(2)V
ALIDATE(DIGIT,"S*"):K$ :: IF
 K$="S" OR K$="*" THEN 430 E
LSE K=VAL(K$):: IF K3<10 THE
N 400
360 IF K3=10 THEN CALL SPGET
("TEN",K3$)ELSE IF K3=11 THE
N CALL SPGET("ELEVEN",K3$)EL
SE IF K3=12 THEN CALL SPGET(
"TWELVE",K3$)
```

****
*
* **FORMATTING OF ASSEMBLY SOURCE CODE** *
* by Harold Hoyt
*
****

This month, another go at assembly. Put a lot of work into a diskfull of material, trying to move to more useful assembly routines. Barry Travers in his MICROpendium series, is working very close to where I want to be. Use XBasic as a platform for all sorts of stuff. I feel that the code written this month is way too sloppy, but the techniques used are different enough from others that the tricks are worth sharing. If I sit on it until it is just right, it will never get done.

Let's do the last part first. Lots of good source code gets mangled by the TI Writer FORMATTER. Newsletter editors often run all their files through the FORMATTER. The most common problem is the formatter eats all of the @ symbols. The source code reads. MOV @R2,R1. The @ symbol dissapears, the text shifts one character to the right, and the text changes to MOV **R2,R1**. Printed emphasized by overstriking 4 times. The ampersand also falls victim to the formatter in program listings.

The solution is to use the Transliterate command to change other characters into @ , & and ^ . I've taken the program listing of this month's major effort 'KLOK3/L ' and fixed it so that it will run through the formatter. To do this, load the file FORMFIX into TI Writer. Then, load your working file after it. In this case, type LF <ENTER> followed by 1 5 DSK1.KLOK3/L. Thus, this main file is loaded after your transliterate file. note the the Formatter commands must have no left margin, and the first character must be a period. Transliterate other characters into the & @ and ^ . For instance, I used the spanish tilde <CTRL-W> for the caret <SHIFT-6>, when you want a caret that **IS NOT BEING USED AS A REQUIRED SPACE**. Once you have your transliterate file entered, edit out the stuff to the right of the transliterate code. This stuff is "remarks", a reminder as to what character is being changed to what other character. Unfortunately, the formatter will not allow anything on the line with the transliterate code, so the "remarks" must be removed before using. The best usage of transliterate codes i've seen is in Jack Shugrue's PLUS.

Now use replace string to replace all of the @ symbols with right braces. I could add another transliterate code to convert another character to a right brace, but aren't we getting in deep? The harder I try for clarity, the muckier it gets. **JAN**, turn **OFF** the formatter! With the formatter off, view the file FORMFIX below.

```
.CO file 'FORMFIX' H. C. Hoyt Jr. 7-31-91
.CO Transliterate @,&,^ in prog listings using formatter
.TL 123:38  { to &
.TL 125:64  } to @
.TL 126:94  ~ to ^
```

ROUTINE INTERWEAVING

by:

Wayne Garrison

Most of us, at one time or another, have had experiences with game programs written in BASIC which didn't offer very good response from your joysticks. This can be very annoying to say the least. The aliens in the game blast you into never-never land before you get a decent chance to return fire. This can be very disastrous to your joysticks. You end up forcing the stick a lot farther than it was intended to be moved and sometimes you end up with the famed two-piece-joystick. If that doesn't happen, you may only get off with blisters on your hands and the frustration of an angry grizzly bear.

This slow response is due to the amount of time the computer takes to scan the keyboard and joysticks. The way most game programs I've encountered are written, there are a few program lines which contain all necessary statements for keyboard operation immediately followed by a few lines with statements required to scan the joysticks. So in essence, the computer is constantly scanning, first the keyboard, then the joysticks for something to happen, such as you pressing a key or moving the joystick. Programs are written this way so the author can offer a certain amount of flexibility. You can use it with or without a joystick. Not all of us have "sticks", you know. They offer improved agility and good scores. Well normally you will use only one of the two, joysticks or keys, thus the computer is doing a lot of unnecessary scanning. This is costing you a lot of points and frustration.

You can do one of several things. You can do a David Letterman with the disk. They fly real good out of an open window. You could just erase the lines that pertain to either the joysticks or the keyboard, but the problem with erasing lines is that you lose the flexibility of being able to use both input devices. I found a way to deal with this problem without sacrificing the flexibility.

I came up with something I call ROUTINE INTERWEAVING. Most game programs usually follow a standard practice known as structured programming. This is essentially the preferred method for good programming which maintains that a program be written in such a manner that all of the sub-routines are more or less kept separate from each other. This method assures that the program will run smooth and will be easier to debug if it doesn't run the way it is suppose to. It also prevents the program execution from needlessly jumping around which also takes time. Basic is a slow programming language by nature of the way it works. I chose to interweave the keyboard scanning routine and the joystick routine into one combined routine. I figured it would increase my response time by 50%. An example of this technique is listed below. This is a portion of a game program I found in an issue of 99'er magazine a few years ago called "Zapper". The original listing as it appeared in the magazine did not have provisions for joysticks, so I felt this would be an excellent "Guinea Pig". I have included my example in a "BEFORE & AFTER" illustration so you can see the difference. If you examine the lines and read through them you will realize the difference in time it takes for the computer to read and interpret each line and perform the tasks outlined in the statements. The program runs surprisingly fast and smooth, for a basic program. If you do not have this game in your own personal software library, I believe it is in the club library if I'm not mistaken.

This technique doesn't follow the norm when it comes to programming, but it works great and can be used to liven up games and give you a little better chance at winning some of the more difficult ones. It may even save the life of your joystick. Give it a try!

BEFORE

```
100 REM ZAPPER KEY/JOYST
1800 CALL KEY(1,K,S)
1805 IF K+1=1 THEN 1880
1810 IF K=2 THEN 2060
1815 IF K=3 THEN 2290
1820 IF K=5 THEN 2520
1825 CALL JOYST(1,X,Y)
1830 IF Y=-4 THEN 1880
1835 IF X=-4 THEN 2060
1840 IF X=4 THEN 2290
1845 IF Y=4 THEN 2520
```

AFTER

```
100 REM ZAPPER KEY/JOYST
1800 CALL KEY(1,K,S)
1805 CALL JOYST(1,X,Y)
1810 IF K+1=1 THEN 1880
1815 IF Y=-4 THEN 1880
1820 IF K=2 THEN 2060
1825 IF X=-4 THEN 2060
1830 IF K=3 THEN 2290
1835 IF X=4 THEN 2290
1840 IF K=5 THEN 2520
1845 IF Y=4 THEN 2520
```

Happy Computing!

NEXT MEETING TUESDAY SEPTEMBER 10, 1991 HAAPPY BIRTHDAY M.U.N.C.H.!! WE'RE 9

1UNCH OFFICERS AND NUMBERS (all in 508 area unless noted)

'resident	W.C. Wyman	865-9683
'ice President	Bruce Willard	852/3250
ecretary	Jim Cox	
'reasurer	Jim Cox	869-2704
ctíng Editor	Jim Cox	
dv.Prog. Chair	Dan Rogers	248-5502
ibrary	Al/Lisa Cecchini	
isk Librarian	Lou Holmes 617	965/3584
ape Librarian	Walter Nowak 413	436/7675
EW-AGE/99	Jack Sughrue	476/7630

MUNCH DUES	
NEW MEMBERSHIP	$25.00
RENEWAL MEMBERSHIP	$15.00
NEWSLETTER ONLY SUBSCRIPTION	$12.50

UGUST MEETING. Corson was able to attend this month's meeting and he showed an udio digitizer for the Mac, and he talked aabout how it might be used on a .I. He also led a spirited discussion about how the T.I., Geneve and other omputers fulfill thier users' needs. Jack spoke about how much he liked his .I. and how it did everything he wanted. Jack won the raffle. Jack also did partial demo of Mac Labels.

EPTEMBER MEETING. This month will feature the completion of Jack's demo, if he s able to attend. There will be other things of interest I'm sure. This is sually one of the best attended meetings of the year.

AFFLE. Every month we have a raffle to help defer the rental cost of our eetin hall. A typical raffle will have game and utility programs, T-Shirts, ooks, bumper stickers, blank discs and all sorts of odds and ends for the T.I.

IBRARY NOTICE. Please return any items borrowed from our library. If you can ot come to a meeting or give these items to someone who will be at the eeting.

EPRINTS. Reprints are permitted as long as credit is given to M.U.N.C.H.

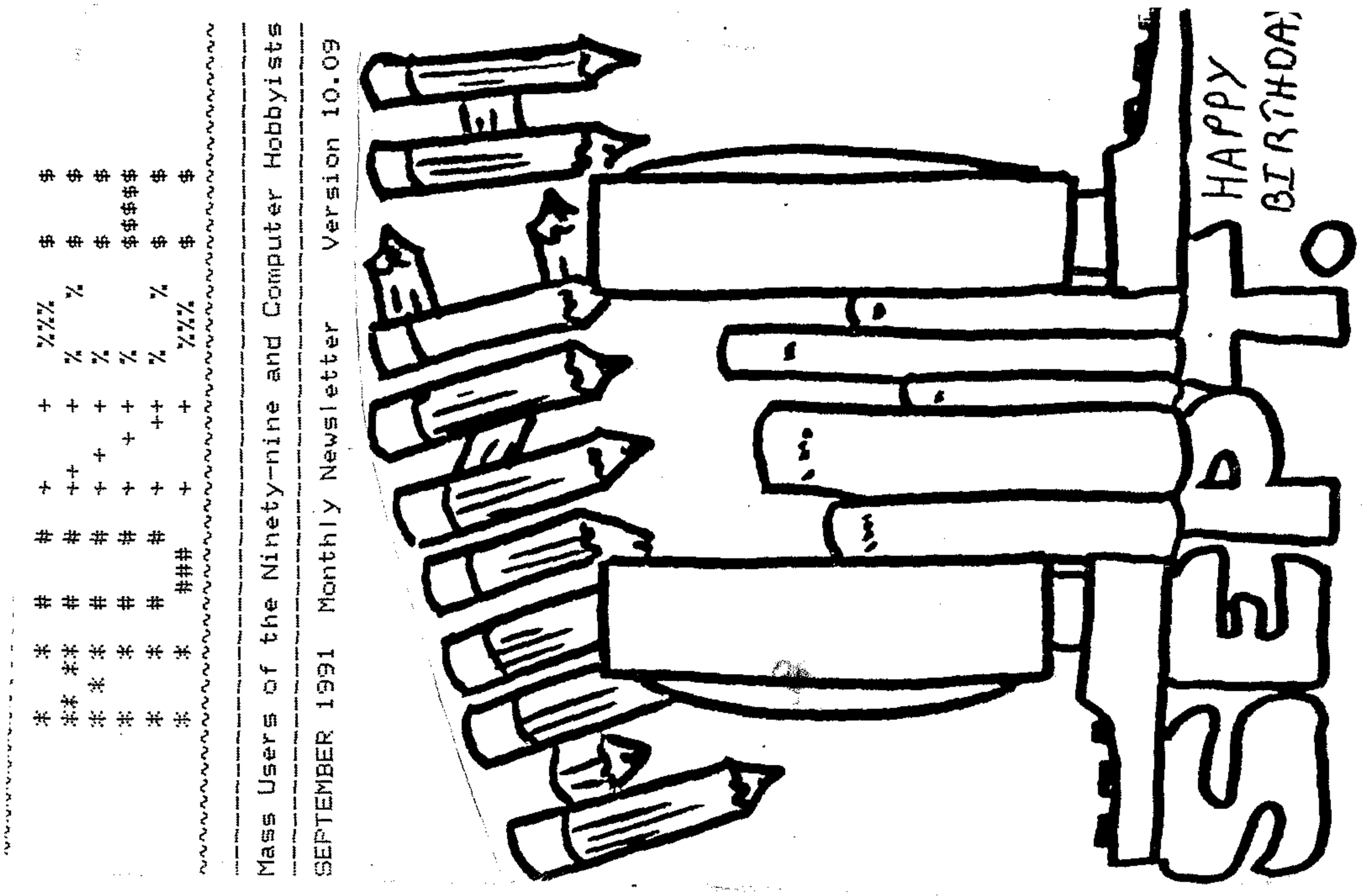
RTICLES. I am always looking for articles for this newsletter, anything which nterests you will probably interest other members of the TI community, so ease share your ideas and opinions with all of us.

ISK LIBRARY. The disk library will be at the meetings from now on. We have opies of all disks in the library and they are available to members for just ..50 each.

)R SALE. The group has a TI Count Business Software package available for le. If interested contact Jim Cox at the above numer or the club address.

ISK OF THE MONTH. This month's disk is aa flippie of Mac Labels and other eat programs. It is #97.

SK PURCHASE. For anyone interested in getting some disks for their personal e we will take pre-orders for a group purchase at this month's meeting. The st is $25.00 per hundred disks, sleves included. Bring your checks or cash to e meeting or mail them to the club address. I will send in the order on gust 31st. I will have some disks for purchase at this month's meeting.



SEPTEMBER 1991 Monthly Newsletter Version 10.09

Mass Users of the Ninety-nine and Computer Hobbyists

M.U.N.C.H.
560 LINCOLN ST.
P.O. BOX 7193
WORCESTER, MA. 01605-7193

WORCESTER, MA 016
PM
4 SEP
1991

29 USA

FIRST CLASS

Next Meeting SEPTEMBER 10th

POSTMASTER: Forwarding and Address Correction Requested.