MID ILLINOIS COMPUTER RESOURCE
ORGANIZATION
P. O. BOX 766 BLOOMINGTON, IL
61701-766

BLOOMINGTON - NORMAL MICRO NEWSLETTER

AUGUST 1985, VOL. 3, No. 8

Welcome to the age of the computer in the home! Future meetings will be held the third THURSDAY of each month at 7 P.M. at the IAA Building in Bloomington . Future dates include August 15, September 19, and October 17.

****

The program for August will consist of everyone attending to demo a cassette or disk program or a cartridge. A cassette recorder and disk drive with extra memory will be provided. Extended basic will also be available. I hope everyone brings something to share. This will give some of our members a chance to demo a program for their first time.

***** PRESIDENTS NOTES *****

As many of you know, I was not at our July meeting as I was on vacation. I hear the idea of everyone bringing a piece of software to demonstrate went very well. So well in fact that we will try it again at this month's meeting.

While I was on vacation in California, I had a chance to visit Tex-Comp's retail store in the Los Angeles area. The store is only open in the evening during the week and on Saturdays as they fill their mailorders during the day. It's was nice to walk in a store and see so much TI software and hardware available. I couldn't resist the temptation to buy. I bought my daughter an add-on 32K (CorComp's new box) so she can Logo. I plan on bringing it to the meetings so we can now run TI-WRITER, Multiplan, assembly language programs, and any other programs that require extra memory.

While I was at Tex-Comp, two other customers where there. They too were from out-of-state. It seems few stores are carrying TI software or hardware. In fact, I checked the yellow pages in everyone big town I was in. There were several ads for stores carrying TI products, but they were all out-of-business. I also saw many computer stores that had closed. Although in Phoenix, I saw a new approach for selling computers. A mall had devoted their upper floor to computers. There must have been at least ten stores with room for another ten. But as usual there wasn't any TI stores. It appears that most TI products must be mailordered. Brian McFeeters

Thanks again to Jim Peterson for his his TIPS FROM THE TIGERCUB. Also, thanks to Maurice E.T. Swinnen for his translation of "Running Text" which appeared in the MANNERS May85 issue (Mid Atlantic 99er's).

****

I received an ad from a new computer store located in Normal. It is located at 309 West Beaufort Suite #9. It appears they carry Commodore 64 software. I am not sure what else. It may be worth checking out.

****

Below is a Extended Basic program that will keep your disk drives running until you push FCTN 4 (clear). Many disk drive cleaning kits require the drive to run for 30 seconds. Use this program and stop when the clean time has been reached.

```
10 CALL CLEAR
20 CALL SCREEN(13)::FOR C=1 TO 12::CALL COLOR(C,16,13)::NEXT C
30 DISPLAY AT(12,10):"CLEANING....":DISPLAY AT(23,2):"(Hold FCTN
CLEAR to Stop)"
40 ON ERROR 60
50 GOSUB 70
60 GOTO 40
70 RUN "DSK1.B"
80 RETURN
```

Reprinted from June/July 1985 newsletter of the Wiregrass 99/4A Users Group.

****

The following programming tips are from the M.U.N.C.H. May85 newsletter.

(1) To generate random numbers with the same statistical distribution as rolling dice, use the routine:

```
ROLL=INT(RND*6)+INT(RND*6)+2
```

(2) A pretty good error trap in a program requiring a "yes" or "no" entry is:

```
1000 INPUT AN$
1010 IF SEG$(AN$,1,1)="Y" THEN ____
1020 IF SEG$(AN$,1,1)="N" THEN ____
1030 GOTO 1000
```

(3) To imitate the sound of a chime, use the routine:

```
1000 F=___
1010 CALL KEY(0,K,S)
1020 IF S=0 THEN 1010
1030 FOR V=0 TO 30 STEP 2
1040 CALL SOUND(-100,F,V,2.756*F,.5*V+15)
1050 CALL KEY(0,K,S)
1060 IF S=1 THEN 1000
1070 NEXT V
1080 GOTO 1010
```

RUNNING TEXT (X-BASIC only)

by Maurice E.T. Swinnen

Recently, I received some 360 programs from a European club. I am busily translating all the prompting into English, of course, in order to avoid a repetition of the situation at Babel at the time.

The reward for doing this usually is, that one learns a new trick here and there. One of the Dutch programmers, who by the way, dedicated an entire program to a passionate plea to stop carbon dioxide pollution of the atmosphere, came up with a couple of neat tricks I had not seen used before.

His first one consists of generating mammoth character sprites and float them on the screen as banners. The top word is created in Line 120, the lower one in Line 130. Make sure that the FOR-NEXT loop has exactly the number of repetitions as the number of letters in each word. Line 140 changes the color of both sprites in a gaudy but rather attractive way.

The second trick is even neater. It creates a text running from right to left on the screen, reminiscent of the bright lights of Broadway.

Lines 200, 210 and 220 show you how long texts may be concatenated into a very long, almost endless, chain. Line 300 is, of course, the heart of the matter, which creates the running effect. If you dislike the sound, just leave it off. It has no bearing on the creation of the running text itself.

If this program doesn`t get you to go to meetings, nothing will!

```
100 CALL CLEAR
110 CALL SCREEN(2):: CALL MAGNIFY(2)
120 T$="MICRO" :: FOR X=1 TO 5 :: CALL SPRITE(#X,ASC(SEG$(T$,X,1)),2,20+4*X,50+1
6*X):: NEXT X
130 T$="99-er CLUB" :: FOR X=1 TO 10 :: CALL SPRITE(#X+6,ASC(SEG$(T$,X,1)),2,80+
4*X,15+16*X):: NEXT X
140 FOR COUNT=1 TO 2 :: FOR X=3 TO 16 :: FOR Y=1 TO 20 :: CALL COLOR(#Y,X):: NEX
T Y :: NEXT X :: NEXT COUNT
150 REM
160 REM
170 REM
180 REM
190 REM
200 RUN$="NOW HEAR THIS:...THE MICRO CLUB MEETS EVERY THIRD THURSDAY OF THE MONT
H...."
210 RUN$=RUN$&"THE AUGUST MEETING WILL BE HELD ON THE 15TH AT THE IAA BUILDING..
.."
220 RUN$=RUN$&"SEE YOU THERE...."
230 GOSUB 290
240 END
250 REM
260 REM
270 REM
280 REM
290 FOR SET=1 TO 9 :: CALL COLOR(SET,16,6):: NEXT SET
300 RUN$=RPT$(" ",28)&RUN$ :: FOR LT=1 TO LEN(RUN$)+1 :: DISPLAY AT(20,1):SEG$(R
UN$,LT,28):: CALL SOUND(150,1000,22,-5,7):: NEXT LT
310 RETURN
```

TIPS FROM THE TIGERCUB

#22

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

This challenge was printed in Tips #21 -

```
100!The Unprintable Unkeyabl
e Program!
110!To shuffle the numbers 1
 to 255 into a random sequen
ce without duplication
120!The strings contain the
ASCII characters 1 to 127 an
d 128 to 255
130!Most of the ASCII charac
ters below 32 or above 159 c
annot be input from the keyb
oard
140!So how was this program
programmed?
150 M$="
                !""#$%&'()*+,-./0
123456789:;<=>?@ABCDEFGHIJKL
MNOPQRSTUVWXYZ[\]^_'abcdefgh
ijklmnopqrstuvwxyz{|}~"
160 M2$="




170 M$=M$&M2$
180 L=LEN(M$):: RANDOMIZE ::
 X=INT(L*RND+1):: N=ASC(SEG$
(M$,X,1)):: M$=SEG$(M$,1,X-1
)&SEG$(M$,X+1,LEN(M$))
190 PRINT N;:: IF LEN(M$)=0
THEN STOP ELSE 180
```

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

```
100 OPEN #1:"DSK1.LONGSTRING
",VARIABLE 163
110 LN=100 :: GOSUB 190 :: A
$=L$&"M$"&CHR$(190)
120 FOR J=1 TO 127 :: C$=C$&
CHR$(J):: NEXT J :: A$=A$&CH
R$(199)&CHR$(127)&C$&CHR$(0)
130 PRINT #1:A$
140 GOSUB 190 :: B$=L$&"M2$"
&CHR$(190)
150 FOR J=128 TO 255 :: D$=D
$&CHR$(J):: NEXT J :: B$=B$&
CHR$(199)&CHR$(128)&D$&CHR$(
0)
160 PRINT #1:B$
170 GOSUB 190 :: F$=L$&"M$"&
CHR$(190)&"M$"&CHR$(184)&"M2
$"&CHR$(0)
180 PRINT #1:F$ :: PRINT #1:
CHR$(255)&CHR$(255):: CLOSE
#1 :: END
190 L$=CHR$(INT(LN/256))&CHR
$(LN-256*INT(LN/256)):: LN=L
N+10 :: RETURN
```

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

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

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

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

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

```
100 FOR CH=33 TO 127 :: CALL
 CHARPAT(CH,CH$):: FOR J=1 T
O 16 STEP 2 :: X$=SEG$(CH$,J
,2)&X$ :: NEXT J :: CALL WRI
TE(CH,X$):: X$="" :: NEXT CH
1000 SUB WRITE(CH,X$):: IF F
LAG=1 THEN 1010 :: FLAG=1 ::
 OPEN #1:"DSK1.WRITE",OUTPUT
,DISPLAY ,VARIABLE 163 :: LN
=3000 :: GOSUB 3000
1010 X=X+1 :: L$=L$&CHR$(200
)&CHR$(16)&X$ :: IF X<5 AND
CH<127 THEN L$=L$&CHR$(179):
: SUBEXIT
1020 X=0 :: PRINT #1:L$&CHR$
(0):: L$="" :: IF CH=127 THE
N 1030 :: GOSUB 3000 :: SUBE
XIT
1030 PRINT #1:CHR$(255)&CHR$
(255):: CLOSE #1 :: GOTO 301
0
3000 L1=INT(LN/256):: L2=LN-
256*L1 :: L$=CHR$(L1)&CHR$(L
2)&CHR$(147):: LN=LN+10 :: R
ETURN
3010 SUBEND
```

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

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

TI-WRITER COMPANION. Loaded with ingenious ways to make your TI-Writer more effective. Well written. Send $2.50 to Dr. Bill Browning, 7541 Jersey Avenue North, Brooklyn Park, MN 55428. Money back guarantee.

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

The K-Town newsletter recently published a utility routine that is so useful that I want to pass it on to everyone. If a program is not resequenced after it is modified, this will compare it with the original and prepare a MERGE format file of all the changes, for the use of others to update their copy.

```
100 !************************
110 !* COMPARE PROGRAM      *
120 !     by Mike Dodd      *
130 !************************
131 ! In K-Town 99'er V.2 #1
      April 1985
140 !Version 85.0406.1XB
     Requires disk drive.
     Compares two programs,
     gives list of all diff-
     erences.
150 !SAVE old program in
 MERGE format (SAVE DSK1.(ol
 dfilename),MERGE). SAVE up-
 dated program in MERGE for-
 mat(SAVE DSK1.(newfilename)
,MERGE)
160 !RUN this program, answe
r prompts for OLD FILE name,
 NEW FILE name, and a differ
ent OUTPUT FILE name.
170 !When finished, type NEW
, then MERGE DSK1.(outputfil
ename) and ENTER
180 !Can be MERGED into othe
r copies of OLD program to
update them
190 DEF @(@$)=ASC(SEG$(@$,1,
1))*256+ASC(SEG$(@$,2,1))
200 A$=CHR$(255)&CHR$(255)::
 DISPLAY AT(1,1)ERASE ALL:"O
LD   FILE:": :"NEW   FILE:
": :"OUTPUT FILE:"
210 ACCEPT AT(1,13)BEEP:B$ :
: ACCEPT AT(3,13)BEEP:C$ ::
ACCEPT AT(5,13)BEEP:D$ :: OP
EN #1:B$,INPUT ,VARIABLE 163
220 OPEN #2:C$,INPUT ,VARIAB
LE 163 :: OPEN #3:D$,OUTPUT,
VARIABLE 163
230 LINPUT #1:@$ :: LINPUT #
2:E$ :: F$=SEG$(@$,1,2):: G$
=SEG$(E$,1,2):: A=@(F$):: B=
@(G$)
240 IF F$=A$ AND G$=A$ THEN
CLOSE #1 :: CLOSE #2 :: PRIN
T #3:A$ :: CLOSE #3 :: STOP
250 IF B>A THEN PRINT #3:F$&
CHR$(131)&" **DELETED LINE *
*"&CHR$(0):: LINPUT #1 :: @$
 :: F$=SEG$(@$,1,2):: A=@(F$
):: GOTO 240
260 IF A>B THEN PRINT #3:E$
:: LINPUT #2:E$ :: G$=SEG$(E
$,1,2):: B=@(G$):: GOTO 240
270 IF @$<>E$ THEN PRINT #3:
E$
280 GOTO 230
```

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

```
100 !by A. Kludge/M. Gordon/
T. Boisseau/J. Peterson/etc.
modified in Tips #22
102 OPTION BASE 1 :: DIM PG$
(127),VV(127),VX(127):: GOTO
 110
105 @,A,A$,B,C,D$,FLAG,I,J,K
,KD,KK,N$,NN,P$,PG$(),Q$,S,S
T,T$(),TT,VT,VV(),VX(),W$,X,
X$,K2,S2
106 CALL INIT :: CALL LOAD :
: CALL LINK :: CALL PEEK ::
CALL KEY :: CALL SCREEN :: C
ALL COLOR :: CALL CLEAR :: C
ALL VCHAR :: CALL SOUND :: !
@P-
150 ! **DELETED LINE **
160 T$(1)="d/f" :: T$(2)="d/
v" :: T$(3)="i/f" :: T$(4)="
i/v" :: T$(5)="pro" :: ON WA
RNING NEXT
170 IMAGE ###
180 DISPLAY AT(1,4):"TIGERCU
B MENU LOADER"
210 D$="DSK1." :: OPEN #1:D$
,INPUT ,RELATIVE,INTERNAL ::
 INPUT #1:N$,A,J,K :: DISPLA
Y AT(1,2)SIZE(27):SEG$(D$,1,
4)&" - Diskname= "&N$;
230 FOR X=1 TO 127 :: IF X/2
0<>INT(X/20)THEN 260
240 DISPLAY AT(24,1):"Type c
hoice or 0 for more 0" :: AC
CEPT AT(24,27)VALIDATE(DIGIT
)SIZE(-3):K :: IF K=0 THEN 2
50 :: IF VV(K)<>5 THEN 411 :
: IF K>0 AND K<NN+1 THEN 420
 ELSE 240
290 DISPLAY AT(X+4,2):USING
170:NN :: DISPLAY AT(X+4,6):
P$ :: PG$(NN)=P$ :: DISPLAY
AT(X+4,18):USING 170:J :: DI
SPLAY AT(X+4,22):T$(ABS(A))
291 VV(NN)=ABS(A):: VX(NN)=A
BS(B)
295 X$="  "&STR$(B):: DISPLA
Y AT(X+4,26):SEG$(X$,LEN(X$)
-2,3):: VT=VT+J
350 DISPLAY AT(X+6,1):"    C
hoice?" :: ACCEPT AT(X+6,16)
SIZE(3)VALIDATE(DIGIT):K ::
IF K<>NN AND K<>NN+1 THEN 41
0
410 IF K<1 OR K>127 OR LEN(P
G$(K))=0 THEN 320
411 IF VV(K)=5 OR(VV(K)=4 AN
D VX(K)=254)THEN 420
412 ON ERROR 417 :: CALL CLE
AR :: OPEN #2:D$&PG$(K):: CA
LL SCREEN(16)
413 LINPUT #2:W$ :: IF EOF(2
)THEN 416 :: PRINT W$
414 CALL KEY(0,K,S):: IF S=0
 THEN 413
415 CALL KEY(0,K2,S2):: IF S
2<1 THEN 415 ELSE 413
416 CLOSE #1 :: CLOSE #2 ::
END
417 DISPLAY AT(12,10):"UNLIS
TABLE" :: CALL SOUND(200,110
,0):: RETURN 400
430 ON ERROR 417 :: CALL INI
T :: CALL PEEK(-31952,A,B)::
 CALL PEEK(A*256+B-65534,A,B
):: C=A*256+B-65534 :: A$=D$
&PG$(K):: CALL LOAD(C,LEN(A$
))
```

The Menu Loader will now list up to 127 programs and files, showing the number of sectors in each and the file type, record type and record length of each file. It will stop at the end of each page, and continue on a default value of 0, or will stop for selection when any key is pressed. It gives disk name, number of sectors used and available. It adds up sectors actually used and gives a warning if all sectors are not accounted for. It will load and run any program which can be loaded from Extended Basic, displaying the program being loaded. It will delete any program or file, after first displaying the filename and requesting verification. It will list any listable file to the screen, pausing on any key input, and can be very easily modified to list to a printer. If a file is not listable, it will inform you so, and restart the menu selection. It has the pre-scan option to speed it up.

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

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

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

MEMORY FULL! - Jim Peterson

MID ILLINOIS COMPUTER RESOURCE ORGANIZATION
P.O. BOX 766
Bloomington, IL 61701-0766

EDMONTON 99'ERS USER SOCIETY
P.O. BOX 11983, EDMONTON
ALBERTA, CANADA T5J-3L1

```
*************************************************************************
*        MMM    MMM    IIIIII    CCCCCCC    RRRRRRRR      OOOOOOOO      *
*        MM M M MM       II      CC         RR      RR    OO      OO    *
*        MM M M MM       II      CC         RRRRRRRR      OO      OO    *
*        MM  M  MM       II      CC         RR      RR    OO      OO    *
*        MM     MM       II      CC         RR        RR  OO      OO    *
*        MM     MM     IIIIII    CCCCCCC    RR          RR OOOOOOOO     *
*                                                                       *
*                                                                       *
*        The MID ILLINOIS COMPUTER RESOURCE ORGANIZATION                *
*************************************************************************
```