Milwaukee
June 86

99 HOCUS

Home Computer
Users Spotlight

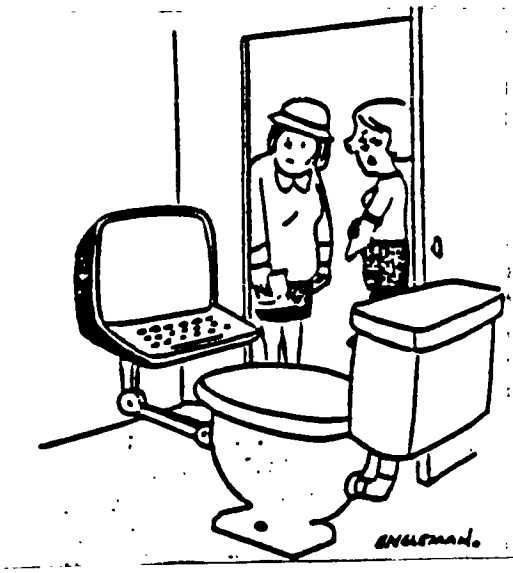
a monthly publication of the
Milwaukee Area 99/4 Users Group

J U N E - 1 9 8 6

MILWAUKEE AREA USER GROUP
4122 GLENWAY WAUWATOSA WI 53222

President...J.Schroeder	2644735
Vice-Pres...D.Walden	5292173
Treasurer...P.Norton	4628954
Secretary...G.Kasica	3217558
Librarian...E.VonDerEhe	5490593
Librarian...F.Pabian	3273618
Newsletter..G.Hitz	5350133
S.I.G.....Schroeder/Walden/Hitz	

Group Meeting
Second Saturday
Wauwatosa S&L 7500 W.State
1:00-4:00
SIG Meeting
First Monday
National S&L 3670 S.Moorland
7:00-10:00
Membership Dues
Individual - $10
Family - $15



Most people read in the bathroom.
Bob is different.

ARROW

This is another demonstration of the almost unbeleivable power and capabilities of 99/4A sprites in Extended Basic

```
1 !!!!!!!!!!!!!!!!
2 ! BY DANNY COX !
3 !!!!!!!!!!!!!!!!
4 !
5 CALL MAGNIFY(4):: CALL CLE
AR :: CALL SCREEN(2)
6 CALL CHAR(96,"FFFFFFFFFFFF
FFFFFF7F3F1F0F0703030180C0E0F0
F8FCFEFFFFFFFFFFFFFFFFFF")
7 FOR X=7 TO 4 STEP -1
8 R=190 :: C=250
9 FOR I=1 TO 25 :: CALL SPRI
TE(#I,96,RND*13+3,R,C):: R=R
-X :: C=C-7 :: NEXT I
10 R=190 :: C=250
11 FOR I=25 TO 1 STEP -1 ::
CALL SPRITE(#I,96,RND*13+3,R
,C):: R=R-X :: C=C-7 :: NEXT
 I
12 NEXT X
13 FOR X=5 TO 7
14 R=190 :: C=250
15 FOR I=1 TO 25 :: CALL SPR
ITE(#I,96,RND*13+3,R,C):: R=
R-X :: C=C-7 :: NEXT I
16 R=190 :: C=250
17 FOR I=25 TO 1 STEP -1 ::
CALL SPRITE(#I,96,RND*13+3,R
,C):: R=R-X :: C=C-7 :: NEXT
 I
18 NEXT X :: GOTO 7
```



```
100 !**********
110 !*ORGANISMS*
120 !**********
130 !By: Ed Lee
140 !1985
150 !
160 X=11 :: CALL SCREEN(2)::
 DISPLAY ERASE ALL :: CALL C
HAR(33,"1"):: FOR I=1 TO 28
:: CALL SPRITE(#I,33,X,96,12
8):: NEXT I
170 FOR I=4 TO 28 STEP 4 ::
X=(RND*5+1)*SGN(RND-.5):: Y=
(RND*5+1)*SGN(RND-.5):: FOR
J=I-3 TO I :: CALL MOTION(#J
,X,Y):: NEXT J :: NEXT I
180 GOTO 170
```

TIPS FROM THE TIGERCUB

#31

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.

Over 130 original programs in Basic and Extended Basic, available on casette or disk, only $3.00 each plus $1.50 per order for PPM. Entertainment, education, programmer's utilities.
Descriptive catalog $1.00, deductable from your first order.
Tips from The Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, just $15 postpaid.
Tips from the Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just $15 postpaid. Or, both for $27 postpaid.
Nuts & Bolts (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. All for just $19.95 postpaid.
Nuts & Bolts No. 2, another full disk of 100 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also $19.95 postpaid, or both Nuts Bolts disks for $37 postpaid.
Tigercub Full Disk Collections, just $12 postpaid! Each of these contains either 5 or 6 of my regular $3 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus!
TIGERCUB'S BEST
PROGRAMMING TUTOR
PROGRAMMER'S UTILITIES
BRAIN GAMES
BRAIN TEASERS
BRAIN BUSTERS!
MANEUVERING GAMES
ACTION GAMES
REFLEX AND CONCENTRATION
TWO-PLAYER GAMES
KID'S GAMES
MORE GAMES
WORD GAMES
ELEMENTARY MATH
MIDDLE/HIGH SCHOOL MATH
VOCABULARY AND READING
MUSICAL EDUCATION
KALEIDOSCOPES AND DISPLAYS
For descriptions of these send a dollar for my catalog!

A few people have asked for a program that they could use to encode personal messages on a BBS. considering the current legal threats to BBS's, I doubt that a SysOp will allow coded messages, but here is a coder/decoder to create code that should be quite difficult to crack. First we need another of those programs that write a program -

```
100 !CODEPRINT by Jim Peterson - creates a random code in a MERGE format program CODESTRING to be MERGEd into CODEMAKER
110 FOR J=1 TO 254 :: N$=N$&CHR$(J):: NEXT J
120 FOR J=1 TO 254 :: RANDOMIZE :: X=INT(RND*LEN(N$)+1):: C$=C$&SEG$(N$,X,1):: N$=SEG$(N$,1,X-1)&SEG$(N$,X+1,LEN(N$)):: NEXT J
130 OPEN #1:"DSK1.CODESTRING",VARIABLE 163,OUTPUT :: PRINT #1:CHR$(0)&CHR$(1)&"C$"&CHR$(190)&CHR$(199)&CHR$(127)&SEG$(C$,1,127)&CHR$(0)
140 PRINT #1:CHR$(0)&CHR$(2)&"C2$"&CHR$(190)&CHR$(199)&CHR$(127)&SEG$(C$,128,127)&CHR$(0)
150 PRINT #1:CHR$(0)&CHR$(3)&"C$"&CHR$(190)&"C$"&CHR$(184)&"C2$"&CHR$(0):: PRINT #1:CHR$(255)&CHR$(255):: CLOSE #1 :: END
```

And now the coder/decoder -

```
100 !TIGERCUB CODEMAKER written by Jim Peterson
110 !The MERGE format program CODESTRING created by the program CODEPRINT must be MERGEd into lines 1-3 of this program
120 DIM A$(254):: DISPLAY AT(3,6)ERASE ALL:"TIGERCUB CODEMAKER" :: DISPLAY AT(12,1):"Do you want to": :"(1)Encode":"(2)Decode"
130 CALL KEY(0,K,ST):: IF K=49 THEN 140 ELSE IF K=50 THEN 290 ELSE 130
140 OPEN #1:"DSK1.CODE",VARIABLE 254,OUTPUT
150 DISPLAY AT(5,6)ERASE ALL:"Type message in segments of":"not more than 254 characters":"and Enter. When done, type"
160 DISPLAY AT(9,1):"END and Enter. Type slowly":"to avoid skipped characters.":"Backspace with FCTN S to":"correct.": :"Press any key"
170 CALL KEY(0,K,ST):: IF ST=0 THEN 170
180 CALL CLEAR :: CALL LONGACCEPT(0,M$):: IF M$="END" THEN 280
190 DISPLAY AT(20,1):"WAIT, PLEASE - ENCODING"
200 FOR J=1 TO LEN(M$)
210 A$(ASC(SEG$(C$,J,1)))=SEG$(M$,J,1)
220 NEXT J
230 FOR J=1 TO 254 :: RANDOMIZE
240 IF A$(J)="" THEN A$(J)=CHR$(INT(26*RND+65))
250 CODE$=CODE$&A$(J)
260 NEXT J :: PRINT CODE$
270 PRINT #1:CODE$ :: CODE$="" :: FOR J=1 TO 254 :: A$(J)="" :: NEXT J :: GOTO 180
280 CLOSE #1 :: END
290 OPEN #1:"DSK1.CODE",VARIABLE 254,INPUT :: CALL CLEAR :: DISPLAY AT(12,1):"DECODING"
300 LINPUT #1:CODE$ :: FOR J=1 TO 254 :: M$=M$&SEG$(CODE$,ASC(SEG$(C$,J,1)),1):: NEXT J :: PRINT M$;:: M$=""
310 IF EOF(1)<>1 THEN 300 :: CLOSE #1 :: END
320 SUB LONGACCEPT(L,M$):: X=0 :: IF L<>0 THEN R=L ELSE R=R+1
330 M$="" :: C=3 :: CH=140 :: CALL CHAR(140,RPT$("0",14)&"FF")
340 CALL HCHAR(R,C,CH):: CH=CH+5+(CH=160)*25 :: CALL KEY(0,K,ST):: IF ST<1 THEN 340
350 IF K<>8 THEN 370 :: X=X-1 :: C=C-1 :: IF C=2 THEN C=3 :: R=R-1
360 M$=SEG$(M$,1,LEN(M$)-1):: GOTO 340
370 IF K=13 THEN 410
380 X=X+1 :: M$=M$&CHR$(K):: CALL HCHAR(R,C,K):: IF X=254 THEN 410
390 C=C+1 :: IF C=31 THEN C=3 :: R=R+1 :: IF R=25 THEN CALL CLEAR :: R=1
400 GOTO 340
410 R=0 :: SUBEND
```

Here is a simple little game I call Cover-Up. Use the #1 joystick, try to cover the white square with the black square. Press the fire button to speed up, release it to slow down.

```
100 CALL CLEAR :: CALL CHAR(96,RPT$("F",64)):: CALL SPRITE(#1,96,5,92,124):: CALL MAGNIFY(4):: CALL SPRITE(#2,96,16,100,100)
110 X=INT(20*RND)-INT(20*RND):: Y=INT(20*RND)-INT(20*RND):: CALL MOTION(#2,X,Y):: T=T+1 :: IF T=250 THEN 300
120 CALL JOYSPEED(1,1):: CALL COINC(#1,#2,8,A):: IF A=-1
```

```
 THEN 130 ELSE 110
130 Z=Z+1 :: DISPLAY AT(1,1)
:Z :: CALL SOUND(-50,500,5):
: GOTO 120
300 CALL DELSPRITE(ALL):: DI
SPLAY AT(12,5):"YOUR SCORE I
S "&STR$(Z):: DISPLAY AT(20,
1):"PRESS ENTER TO PLAY AGAI
N"
310 CALL KEY(0,K,S):: IF S=0
 OR K<>13 THEN 310 :: T,Z=0
:: GOTO 100
2111O SUB JOYSPEED(N,A):: CA
LL JOYST(N,X,Y):: CALL KEY(N
,K,ST):: S=S+K/9-1 :: S=S*AB
S(S)>0):: IF S>30 THEN S=30
21111 CALL MOTION(#A,-(Y*S),
X*S):: SUBEND
```

For a one-handed BREAK, if you can't reach FCTN and 4, try FCTN with J and the space bar together.

If you like to call BBS's, try the TIBBS Spirit of 99 BBS in Columbus, Ohio on (614)451-5885 and leave me a "hello!"

Probably useless info - holding down FCTN and CTRL together and typing 1, 2, 3 and 5 will give ASCII codes 145, 151, 133 and 148, which are the codes obtained from CTRL Q, W, E and T, the keys diagonally below the 1, 2, 3 and 5.

Occasionally someone sends me a program they have keyed in from my newsletter, and asks why it won't run, so I wrote this routine to help find the errors. It is also useful to check whether two copies of a program are identical, but only if they have not been resequenced.

```
100 !CHECKER by Jim Peterson
 - to compare two programs a
nd list all differing lines
 to the printer
110 DISPLAY AT(12,1)ERASE AL
L:"1st program DSK/filename?
":"DSK" :: ACCEPT AT(13,4):F
1$
120 DISPLAY AT(12,1)ERASE AL
L:"2nd program DSK/filename?
":"DSK" :: ACCEPT AT(13,4):F
2$
130 OPEN #1:"DSK"&F1$,INPUT
:: DIM M$(500),CH(500):: OPE
N #2:"PIO",VARIABLE 255 :: P
RINT #2:CHR$(15)
140 X=X+1 :: LINPUT #1:M$(X)
:: M$(X)=M$(X)&" " :: IF EOF
(1)<>1 THEN 140 :: CLOSE #1
:: OPEN #1:"DSK"&F2$,INPUT
150 IF EOF(1)=1 THEN 230 ::
LINPUT #1:X$ :: X$=X$&" "
160 FOR Y=1 TO X
170 IF X$=M$(Y)THEN CH(Y)=1
:: GOTO 150
180 NEXT Y
190 P2=POS(X$," ",1):: P2$=S
EG$(X$,1,P2-1)
200 FOR Y=2 TO X :: P1=POS(M
$(Y)," ",1):: P1$=SEG$(M$(Y)
,1,P1-1)
210 IF P2$=P1$ THEN PRINT #2
:"1st program = ";M$(Y):"2nd
 program = ";X$ :: CH(Y)=1 :
: GOTO 150
220 NEXT Y :: PRINT #2:"2nd
program = ";X$ :: GOTO 150
230 FOR J=1 TO X :: IF CH(J)
=0 THEN PRINT #2:"1st progra
m = ";M$(J)
240 NEXT J
250 CLOSE #1 :: CLOSE #2
```

Here's a great idea that was printed and reprinted in several newsletters -

At the beginning of a program that will run only in Basic, add the lines -

```
1 IF PI=3 THEN (first line
of program)
2 PRINT "YOU ARE IN EXTENDED
BASIC":"THIS PROGRAM RUNS
ONLY IN BASIC"
3 STOP
```

The idea is that PI is a function in XBasic with the value of pi, but is just a variable name in Basic with an undefined value of 3.

The trouble is, it doesn't work! If PI is keyed in from Basic and saved, it is saved in token format as a variable name, and when loaded back into XBasic is still just a variable name. And if PI is saved from XBasic, it is tokenized as a function, loads back into Basic as an unrecognized function and crashes! Can anyone come up with a way around that?

The above is the answer to the Challenge in Tips #3. Lines 100 and 110 were keyed in and saved from Basic, and loaded back into XBasic, then lines 120 and 130 were keyed in.

Here is a handy PEEK that hasn't been published as widely as most of them -

```
100 CALL INIT
110 CALL PEEK(8192,X)!Thanks
to Dale Loftis in the Orange
County UG newsletter!
120 PRINT X !If X=32 you are
in Extended Basic; if X=165
you are in Basic with the
Editor Assembler or
Min:Memory module inserted.
```

And another 3-D sprite demo, just to make all the Apple polishers jealous. See if you can figure out how it works.

```
100 CALL CLEAR :: CALL SCREE
N(5):: CALL CHAR(100,RPT$("F
",64)):: CALL MAGNIFY(4):: F
OR S=5 TO 9 :: CALL COLOR(S,
16,1):: NEXT S
110 DISPLAY AT(3,3):"TIGERCU
B SPRITE SHUFFLE" !by Jim Pe
terson
120 DATA 70,116,2,75,121,7,6
9,124,11,78,115,16
130 FOR J=5 TO 8 :: READ P(J
,1),P(J,2),L(J):: CALL SPRIT
E(#J,100,L(J),P(J,1),P(J,2))
:: NEXT J :: W=45
140 DATA 5,6,7,8,8,5,6,7,7,8
,5,6,6,7,8,5
150 RESTORE 140 :: FOR Y=5 T
O 8 :: READ A,B,C,D
160 FOR J=1 TO W :: CALL LOC
ATE(#A,P(A,1)-J,P(A,2),#B,P(
B,1),P(B,2)-J,#C,P(C,1)+J,P(
C,2),#D,P(D,1),P(D,2)+J):: W
=99 :: NEXT J :: GOSUB 180
170 NEXT Y :: GOTO 150
180 FOR J=5 TO 7 :: CALL POS
ITION(#J,P(J+1,1),P(J+1,2)):
: NEXT J :: CALL POSITION(#8
,P(5,1),P(5,2))
190 T=L(8):: L(8)=L(7):: L(7
)=L(6):: L(6)=L(5):: L(5)=T
200 FOR J=5 TO 8 :: CALL SPR
ITE(#J-4,100,L(J),P(J,1),P(J
,2)):: NEXT J
210 FOR J=5 TO 8 :: CALL SPR
ITE(#J,100,L(J),P(J,1),P(J,2
)):: NEXT J :: CALL DELSPRIT
E(#1,#2,#3,#4):: RETURN
```

Do you need some really REAL BIG letters on the screen? Just type your letter at the beep.

```
100 DIM X$(96):: CALL CLEAR
:: FOR CH=33 TO 89 STEP 8 ::
 FOR A=0 TO 7 !REAL BIG LETT
ERS by Jim Peterson
110 CALL CHARPAT(CH+A,X$(CH+
A-32)):: CALL CHAR(CH+A,"0")
:: L$=L$&RPT$(CHR$(CH+A),3):
: NEXT A
120 FOR T=1 TO 3 :: R=R+1 ::
 DISPLAY AT(R,4):L$ :: NEXT
T :: L$="" :: NEXT CH
130 CH$(1)=RPT$("0",16):: CH
$(2)=RPT$("F",16)
140 CALL SOUND(100,500,0)
150 CALL KEY(0,CH,S):: IF S=
0 OR CH>96 THEN 150
160 CALL HEX_BIN(X$(CH-32),B
$):: FOR J=9 TO 64 :: CALL C
HAR(J+32,CH$(VAL(SEG$(B$,J,1
))+1))
170 NEXT J :: GOTO 140
180 SUB HEX_BIN(H$,B$):: HX$
="0123456789ABCDEF" :: BN$="
0000X000100X001000X00110X100
1X0110X0111X1000X1001X1010X1
011X1100X1101X1110X1111"
190 FOR J=LEN(H$)TO 1 STEP -
1 :: X$=SEG$(H$,J,1)
200 X=POS(HX$,X$,1)-1 :: T$=
SEG$(BN$,X*5+1,4)&T$ :: NEXT
 J :: B$=T$ :: T$="" :: SUBE
ND
```

Thought for the day. The excuses for piracy are exactly the same as the excuses for shoplifting, but you probably won't have to tell them to the judge - in this world, at least.

And that is almost

MEMORY FULL

Jim Peterson

TIPS FROM THE TIGERCUB

#32

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.

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, only $3.00 each plus $1.50 per order for PPM. Entertainment, education, programmer's utilities.
Descriptive catalog $1.00, deductable from your first order.
Tips from The Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, just $15 postpaid.
Tips from the Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just $15 postpaid. Or, both for $27 postpaid.
Nuts & Bolts (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. All for just $19.95 postpaid.
Nuts & Bolts No. 2, another full disk of 100 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also $19.95 postpaid, or both Nuts Bolts disks for $37 postpaid.
Tigercub Full Disk Collections, just $12 postpaid! Each of these contains either 5 or 6 of my regular $3 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus!
TIGERCUB'S BEST
PROGRAMMING TUTOR
PROGRAMMER'S UTILITIES
BRAIN GAMES
BRAIN TEASERS
BRAIN BUSTERS!
MANEUVERING GAMES
ACTION GAMES
REFLEX AND CONCENTRATION
TWO-PLAYER GAMES
KID'S GAMES
MORE GAMES
WORD GAMES
ELEMENTARY MATH
MIDDLE/HIGH SCHOOL MATH
VOCABULARY AND READING
MUSICAL EDUCATION
KALEIDOSCOPES AND DISPLAYS
For descriptions of these send a dollar for my catalog!

I've found a bug in the Tigercub Menuloader V.05 which won't let you print a disk catalog if the disk contains the maximum 127 files. This should fix it.

```
340 I=I+1 :: IF I>127 THEN K=X :: GOTO 430
520 DISPLAY AT(X+5,12)SIZE(12)," #?" :: ACCEPT AT(X+5,15)SIZE(3)VALIDATE(DIGIT);KD :: IF KD<1 OR KD>NN THEN 520
```

I think that all program listings should be printed in 28-column format, exactly as they appear on the screen - it makes it so much easier to key them in without errors. I combined parts of two of my programs to make the following. It is written for the Gemini 10X but the lines of printer control codes are annotated to help others make adjustments.

```
100 DIM K$(240):: LN=100 :: DISPLAY AT(3,4)ERASE ALL:"TIGERCUB PROGLISTER": :" Will convert a program":"listing to 28-column format,"
110 DISPLAY AT(7,1):"exactly as it appears on the":"screen, and print it in 4":"columns."
120 DISPLAY AT(11,1):" Program must be RESequenced":"and LISTed to disk by":"RES (enter)":"LIST DSK1.(filename) (Enter)"
130 DISPLAY AT(18,1):"Filename? DSK" :: ACCEPT AT(18,14)BEEP:F$
140 OPEN #1:"DSK"&F$,DISPLAY,VARIABLE 80,INPUT
150 IF EOF(1)=1 THEN 260 :: LINPUT #1:A$
160 IF LEN(A$)<80 THEN LN=LN+10 :: GOTO 210
170 LINPUT #1:B$ :: IF POS(B$,STR$(LN),1)=1 THEN FLAG=1 :: LN=LN+10 :: GOTO 210
180 A$=A$&B$ :: IF LEN(A$)<160 THEN LN=LN+10 :: GOTO 210
190 LINPUT #1:B$ :: IF POS(B$,STR$(LN),1)=1 THEN FLAG=1 :: LN=LN+10 :: GOTO 210
200 A$=A$&B$ :: LN=LN+10
210 S=1
220 L$=SEG$(A$,S,28)
230 IF L$<>"" THEN 240 :: IF FLAG=1 THEN FLAG=0 :: A$=B$ :: GOTO 160 :: ELSE GOTO 150
240 X=X+1 :: K$(X)=L$ :: S=S+28 :: IF X=240 THEN 250 :: GOTO 220
250 X=0 :: CALL PRINTER(K$()):: GOTO 220
260 CLOSE #1 :: FOR J=X+1 TO 240 :: K$(J)="" :: NEXT J :: CALL PRINTER(K$()):: PRINT #2:CHR$(12):: END
270 SUB PRINTER(B$()):: IF F=1 THEN 340 :: F=1
280 OPEN #2:"PIO.LF",VARIABLE 132 :: PRINT #2:CHR$(15);CHR$(27);"N";CHR$(6);!condensed print and perforation skip
290 PRINT #2:CHR$(27);"6";!- double-struck printing, optional
300 PRINT #2:CHR$(27);CHR$(42);CHR$(0);!download normal characters - required if lines 310-330 are used
310 PRINT #2:CHR$(27);CHR$(42);CHR$(1);CHR$(48);CHR$(0);CHR$(64);CHR$(30);CHR$(96);CHR$(17);CHR$(72);CHR$(5);CHR$(66);CHR$(61);CHR$(0);!slash the zero - optional
320 PRINT #2:CHR$(27);CHR$(42);CHR$(1);CHR$(42);CHR$(0);CHR$(8);CHR$(34);CHR$(8);CHR$(0);CHR$(62);CHR$(0);CHR$(8);CHR$(34);CHR$(8);!broaden the asterisk - optional
330 PRINT #2:CHR$(27);CHR$(36);CHR$(1);!activate redefined characters - required if lines 310-320 are used
340 FOR C=1 TO 60 :: IF B$(C)="" THEN 360 :: PRINT #2;TAB(10);B$(C);TAB(41);B$(C+60);TAB(72);B$(C+120);TAB(103);B$(C+180);CHR$(10)
350 NEXT C
360 SUBEND
```

I had trouble in debugging that program because printing the control codes gave me unwanted line feeds, and using semicolons to prevent line feeds will interfere with tabs in the first line of text. An article by Art Byers in the Central Westchester UG newsletter gave me the solution - suppress all the line feeds by opening the printer with PIO.LF, and put them back in where you need them with CHR$(10)!

We haven't had a random music player in a long time. This one is called ECHO but I don't know where it came from.

```
100 RANDOMIZE :: DEF X=INT(RND*7):: FOR B=0 TO 6 :: A(B)=VAL(SEG$("247262294333439392448",(B+1)*3-2,3)):: NEXT B :: B,C,D=X
110 CALL SOUND(-900,A(B),0,A(C),9,A(D),19):: D=C :: C=B :: B=X :: GOTO 110
```

Sound effects - thanks to Greg Healy in the Edmonton User Group newsletter -

```
100 CALL INIT
110 FOR J=2000 TO 2300 STEP
10 :: CALL LOAD(-31568,J)::
 NEXT J
```

To go directly from XBasic to console Basic - thanks to Greg Healy in the Edmonton User Group newsletter -

```
CALL INIT :: CALL LOAD(-3196
2,8787)
```

Enter. Ignore the error message. Type NEW and Enter.

```
> TI BASIC READY
```

This routine will read a file of 28-character records and scroll them up the lower half of the screen without disturbing the upper half.

```
100 DISPLAY AT(12,1)ERASE AL
L:"FILENAME? DSK" :: ACCEPT
AT(12,14)BEEP:F$ :: CALL CLE
AR
111 OPEN #1:"DSK"&F$,INPUT
112 DIM M$(400)
113 X=X+1 :: LINPUT #1:M$(X)
120 DISPLAY AT(24,1):M$(X)
125 R=24
130 FOR T=X-1 TO 1 STEP -1 :
: IF R>13 THEN R=R-1 :: DISP
LAY AT(R,1):M$(T)
140 NEXT T :: IF EOF(1)<>1 T
HEN 113 ELSE CLOSE #1
```

```
10 !ONE-LINE MORTGAGE PAYMEN
T CALCULATOR BY SAM MORABITO
100 CALL CLEAR :: INPUT "ENT
ER P,R,N WHERE P=AMOUNT, R=R
ATE, N=YEARS":P,R,N :: PRINT
 "$";INT((P*R/1200)/(1-1/(1+
R/1200)^(N*12))*100+.5)/100;
"PER MONTH"
```

A number always prints out with a blank space before and after it (except that a negative number is preceded by -). This is not always desirable when foreatting a screen or printout. The solution is to change the number to a string by using STR$ -

```
100 CALL CLEAR
110 PRINT "  MULTIPLICATION
TABLES": :
120 FOR J=1 TO 9
130 FOR K=1 TO 9
140 PRINT TAB(K*3-2);STR$(J*
K);
150 NEXT K
160 PRINT : :
170 NEXT J
```

Regarding the CHECKER program in Tips #31, I should have mentioned that the two programs to be compared must first be LISTed to one disk by -

```
LIST "DSK1.(filename)
```

- using a different filename for each.

We are still finding new ways to skin the kitty. In Tips #26 I listed three algorithms to alternate between the two joysticks. Rick Humburg sent me another which is the simplest and fastest of all -

```
100 Z=2
110 Z=3-Z :: CALL JOYST(Z,X,
Y).....and back to 110!
```

Here are some more dark secrets Texas Instruments didn't tell us. The User's Reference Guide claims that the computer can produce frequencies up to 44733 Hz, "well above human hearing limits", but then admits "the actual frequency produced may vary from 0 to 10 percent depending on the frequency." According to Jim Hindley, the highest frequency actually produced is 37287 (which is certainly not above the hearing range of some humans, but neither is 44733!), and the maximum error rate far exceeds 10 % because any frequency you call for from 31953 to 43733 ends up as exactly 37287! Not to worry, the frequencies in the normal range of music are accurate enough and your TV speaker probably can't reproduce frequencies above 20000 anyway.

And did you know that TI really gave us only 15 volumes, not 30? Listen and count them -

```
100 FOR V=0 TO 29 STEP 2
110 CALL SOUND(1000,500,V)
120  CALL SOUND(1000,500,V+1
1)
130 FOR D=1 TO 500
140 NEXT D
150 NEXT V
```

And the duration values are just as inaccurate. Experimenting with a series of 8 CALL SOUNDs in a loop repeated 100 times, I found that execution time was 40 seconds for any duration between 1 and 49, or a negative duration; 54 seconds for any duration between 50 and 66; 67 seconds between 67 and 83; 80 seconds between 84 and 99; 94 between 100-116; 106 between 117-133....!

I guess I've been neglecting those who don't have the Extended Basic module, so -

```
100 CALL SCREEN(16)
110 CALL CLEAR
120 PRINT TAB(8);"GREENSLEEV
ES": : : : : : : : : : : : :
 :"programmed by Jim Peterso
n"
130 DIM S(15)
140 FOR N=1 TO 12
150 READ S(N)
160 NEXT N
170 M$="42100000995ABDC324E7DB
A5100669918240000425A000DBC35A66
A5243C7EB19942000A57E66BD3CA5
423C187E423CBD5A810000FFC3"
180 RANDOMIZE
190 FOR R=1 TO 12
200 CALL COLOR(R+1,1,1)
210 CALL CHAR(32+R*8,CH$&CH$
)
220 FOR T=R TO 25-R
230 CALL HCHAR(T,R,32+R*8,34
-2*R)
240 NEXT T
250 NEXT R
260 CALL SCREEN(2)
270 FOR R=1 TO 12
280 CALL COLOR(R+1,R+2,1)
290 CH$=SEG$(M$,INT(47*RND+1
)*2-1,8)
300 CALL CHAR(32+R*8,CH$&CH$
)
310 NEXT R
320 DATA 247,277,294,311,330
,370,392,440,494,523,554,587
330 DATA 2,5,5,4,7,5,2,8,5,3
,9,5,1,10,1,2,9,3,4,8,3,2,6,
3,3,3,1,1,5,3
340 DATA 2,6,1,4,7,5,3,5,2,1
,4,2,2,5,2,4,6,1,2,4,4,4,1,1
350 DATA 2,5,1,4,7,5,2,8,5,3
,9,5,1,10,5,2,9,5
360 DATA 4,8,3,2,6,3,3,3,3,1
,5,3,2,6,3,3,7,5,1,6,2,2,5,1
370 DATA 3,4,1,1,2,2,2,4,1,4
,5,1,2,1,5,6,5,1
380 DATA 2,12,9,2,12,7,2,12,
3,3,12,12,1,11,9,2,9,7
390 DATA 4,8,6,2,6,3,3,3,3,1
,5,5,2,6,3,4,7,5,2,5,3
400 DATA 3,5,5,1,4,4,2,5,5,4
,6,1,2,4,1,6,1,1
410 DATA 6,12,9,3,9,12,1,11,
8,2,9,7,4,8,6,2,6,3,3,3,3
420 DATA 1,5,3,2,6,2,3,7,5,1
,6,6,2,5,5,3,4,1,1,2,2,2,4,4
,6,5,1,1,1,5,7,5,1
430 FOR J=1 TO 223 STEP 3
440 READ T,A,B
450 GOSUB 530
460 FOR TT=1 TO T
470 CALL SOUND(-999,S(A),5,S
(B),7)
480 NEXT TT
490 NEXT J
491 FOR V=0 TO 28
492 CALL SOUND(-999,S(A),V,S
(B),V+7)
493 NEXT V
500 CALL SCREEN(INT(14*RND+2
))
510 RESTORE 330
520 GOTO 270
530 CALL COLOR(A+1,INT(14*RN
D+2),1)
540 CALL COLOR(B+1,INT(14*RN
D+2),1)
550 RETURN
```

```
1 !from 9 T 9 UG newsl. Aug
85
100 PRINT """Hello"" said TI
"
110 PRINT "Press ""ENTER"" t
o continue"
```

If you bite the hand that feeds you, you'll go hungry tomorrow. Don't be a pirate!

MEMORY FULL TO BUSTIN'

Jim Peterson

>> FASTERM << by Paul Charlton

Along with this month's Terminal Emulator Demonstration we'll review V1.11 and V1.16 of Paul Charlton's FASTERM program, a FREEWARE program currently in our library. A requested donation of $10-$15 should be sent to Paul Charlton 1110 Pinehurst Ct, Charlottesville, Virginia 22901.

Version 1.16 consists of two program image files, FASTERM and FASTERN (or UTIL1 & UTIL2) loaded from EDITOR/ASSEMBLER option 5 or TI-WRITER option 3. Version 1.11 loads directly from the ED/ASS option 3 or from BASIC with ED/ASS via the program BASICLOAD. Included are separate options for TEII or XMODEM protocols. Other programs often included are CHARA1 (true lower case), FTERMDOC, COMMSUM (a file of summarized commands), DEFAULT or TERMSET (a basic program used to set up program parameters).

To create a default parameter file, use console basic (ED/ASS) and just follow the screen directions. Using FASTERM you will be given a prompt for the filename you used when you created it. Thus you may set up several files for different boards you contact.

File transfers are done with one of the following protocols:
1. ASCII text files 2. TERMINAL EMULATOR II 3. XMODEM

To receive ASCII files, first create a logging file by pressing FCTN 'B'. This will close any previous logging file and ask for a new logging filename. Press FCTN '.' to stop data from going to the logging file. Press FCTN 'B' to write data to the logging file and close it. FCTN 'Y' clears data from memory buffer.

To send ASCII files press FCTN 'N', give the filename and FCTN ',' to initiate sending. You will be asked if you want to send line-by-line. Sending data this way you must press the SPACEBAR for each line. Press FCTN '4' to abort sending. If not sent line-by-line, the file is sent as fast the other system can handle (XOFF/XON handshake). Press FCTN '4' when finished. ASCII files sent by either method may be DV80 or DF80 format. Press FCTN 'J' to add linefeeds with each carriage return and again to disable it.

Using TEII protocol, press FCTN SHIFT 'T' simultaneously and then FCTN 'N' to to input the filename. Inform the other system you are to upload or download and and FCTN ',' to send, and again, FCTN '4' aborts at anytime.

To download with XMODEM press FCTN 'N', give the filename and input upload or download. When informed that transfer is started, press FCTN SHIFT 'X' to select XMODEM protocol and enter 'R' to receive or 'S' to send. Answer 'YES' if asked about CRC. Again FCTN '4' will abort transfer.

Other special functions included in FASTERM:

FCTN '0' will suspend incoming data,
then SPACEBAR scrolls back thru the memory buffer,
FCTN SHIFT 'P' does a screen dump,
and FCTN '0' returns to incoming data.

FCTN 'K' enables an elapsed timer for baud rates not over 1200.
Timer stops with any disk operation.
FCTN '7' changes text color.
FCTN '8' changes screen color.
FCTN '9' gives a catalog of filenames.
CTRL '0' toggles screen widths between 40 and 80 columns.
CTRL '1' toggles BAUD rates.
CTRL '2' toggles print spooler ON/OFF
CTRL '3' toggles parity ODD/EVEN/NONE
CTRL '4' changes Modem port.
CTRL '5,6,7' for printer parity, port and baud rates.

MILWAUKEE AREA 99-4 USER GROUP
4122 N. GLENWAY
WAUWATOSA, WI 53222

MILWAUKEE, WI
PM
10 JUN
1986

Edmonton 99UG
box 11983
Alberta
Canada T5J 3L1