PUG PERIPHERAL

VOLUME 9 September 1986 10.9

Next Meeting Of The PUG Will Be

Sunday September 21, 1986
at the south campus of the community college of
Allegheny county.

Time	Planned Activity
4:00	Repairing your keyboard with J. Wilforth
4:00	Basic with J. Zittrain
5:00	Potporri with DFL
5:00	Forth with S. Coleman
6:30	General Meeting

There will be a $5 raffle for either a gram Kracker, Ram Disk Card or a printer depending on how many tickets we sell. As usual tickets will be $5 for one or $15 for 4. See DFL for yours.

Sorry about the breifity of this newsletter, but I just got back from my mountain retreat and am in a state of oraganized chaos.

Included in this issue of the peripheral is a copy of my graphics article that was printed in the June Micropendium. In case you are wondering, I am currently writing another part of the series which should be completed shortly.

Quick, a five letter word to describe your summer. SHORT????

Please attend this meeting if you are concerned about the future of the PUG. I will be attempting to develope a game plan for the next 7 months and I want your input. If you do not offer you ideas now, don't complain tommorrow. I have a lot of ideas and would like to know what the consensus is. BE THERE!

Don't miss John Wilforth's class(per se) on repairing your console. John is our aspiring electrical handyman and if I know him I think you will find the class well worth of your time.

DARREN'S 6th LAW:
Time is a one-way commodity, you can sell it,
but you can never buy it back.

See you at the meeting.....>DFL

COMPUTER GRAPHICS

by Darren Leonard PUG

Have you ever seen the television commercial for high performance cars, in which there is a man in a lab jacket standing next to a computer terminal. On the monitor is a 3-D representation of the car and by pushing a few keys the car rotates on the screen so that it may be viewed from several different perspectives. Have you ever wondered how this was done? If so read on.

The concept of using computers as design tools is often called CAD for Computer Aided Design, and in many places the draftsman is replacing the drawing board and T-square with a keyboard and monitor. As new monitors are giving better resolution than ever before, CAD is only going to become more important in the future.

What are the benefits of CAD that make it so advantageous? First of all, once the critical data is entered for a design, it can be viewed and edited on the screen very easily. The drawing can be enlarged,rotated or printed on paper at the touch of a key. Finally, if the computer is properly programmed, the design can be analysed for stresses and other engineering related design problems.

It should be obvious that these benefits outway the high cost of CAD systems and should be taken seriously.

Now to the TI. The TI has limited resolution and cannot compare to high priced sophisticated equipment. However, we can do some basic things on the 99/4A that will give you a feel of what CAD is about.

In these series of articles, I will go over,step by step, how to plot circles,lines,squares and cubes on your screen. You will then be able to adjust the size(scale), move to a different screen location or rotate the image in either 2 or 3 dimensions.

Subroutines will be provided to make things as understandable as possible.

Firstly, let's make sure we are familiar with cartesian coordinates. In figure A, a cartesian axis is drawn, and the vertical axis is labled Y and it is positive above the horizontal axis and negative below the horizontal axis. The horizontal axis is labled X and is positive to the right of the vertical axis and negative to the left of the vertical axis.

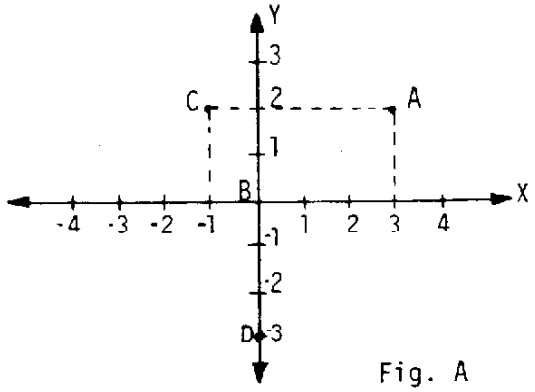


Fig. A

Points within the axis can be identified by their location with respect to the place where X and Y are zero. It is convential to give the location of a point by specifying its X location first followed by its Y-value, this can by abbreviated at (X value,Y-Value). Therefore, (3,4) means to go three places to the right of zero(center) and four places up. To check that you understand this the position of the points in figure A are A=(3,2) B=(0,0) C=(-1,2) D=(0,-3)

Unfortunatley, computer manufacturers decided to invert the vertical axis so that is has the largest positive value at the bottom of the screen. Figure B represents your TV screen. The zero point is in the top left corner.

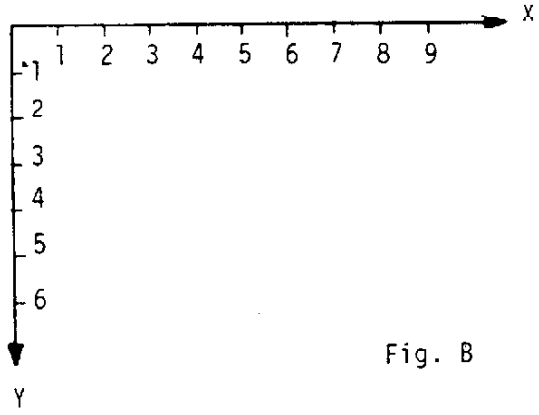


Fig. B

Since it is easiest to work with cartesian coordinates, we must develope a relationship that will convert cartesian coordinates to screen coordinates. This will be inserted into the PUTDOT subroutine.

The expressions

X+125=screen position in X direction

95-Y=screen position in Y direction

To verifiy that this is so, use the cartesian location of(0,0) and change it to its screen position. 0+125=125 and 95-0=95, thus the screen coordinates are (125,95), which is the center of the TV screen. Try a few other conversions to prove to yourself that this is correct.

To keep things simple, the PUTDOT subroutine will do the conversions for you and all you have to provide to it is the cartesion coordinates of the point you want to plot.

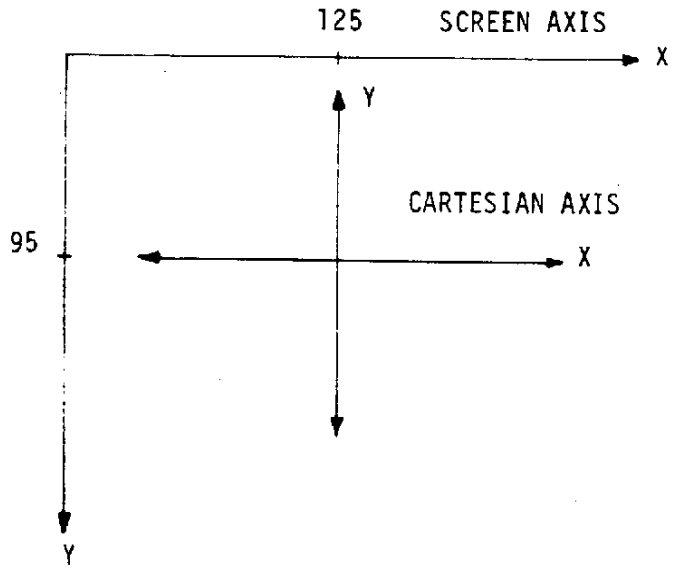


FIGURE OF CARTESIAN COORDINATE AXIS CENTERED WITH RESPECT TO THE SCREEN COORDINATE AXIS.

It is now important that we define a point. In this article, a point will refer to a single pixel on the screen. The TI has about 50,000 points on its screen. Single pixels can be acessed easily only in the bit map mode of the 9918A VDP.

Since Extended Basic includes no intrinsic provisions for plotting in tl bit map mode, we must improvise.**

The subroutine, PUTDOT does just that. It redefines the character definitions so that it appears to be plotting in BIT-MAP. However,there are limitations, first of all it is somewhat slower that we would like, and secondly, since ther are only about 130 different redefinable characters, it can only plot around 120 to 400 points depending on the specific geometry. It will be more than adequate for the rest of this article. Subroutine PUTDOT is listed in PROGLST A.

The PUTDOT subprogram is called from your program with the command

CALL PUTDOT(X,Y)

where X=the x-value in cartesian coordinates and Y=the y value in cartesian coordinates. Since the subprogram will convert them to the corresponding screen coordinates, the range of plottable values are

-125<=X<=125

-95<=Y<=95

if you call the subprogram and send it values outside this range, the subprogram will ingnore the call and return to the main program without crashing!

By using the PUTDOT command you can now plot coordinate axis on your screen with this small calling program. (remember to merge putdot at the end!)

```
100 FOR X=-125 TO 125
110 CALL PUTDOT(X,0)
120 NEXT X                PROGLST2
130 FOR Y=-95 to 95
140 CALL PUTDOT(0,Y)
150 NEXT Y
```

If you change the zeros in lines 110 and 130 to other integer values, you can move the point of intersection anywhere on the screen. Try different values and observe the change in the location of the zero.

Now, suppose you would like to plot a diagonal line on the screen. Vertical and horizontal lines can be drawn rather easily with the above PROGLST2. Diagonals, are slightly more involved. Lets take a look at a diagonal line and then I will explain the soltion.

Figure C is a diagonal line plotted on cartesian coordinates.

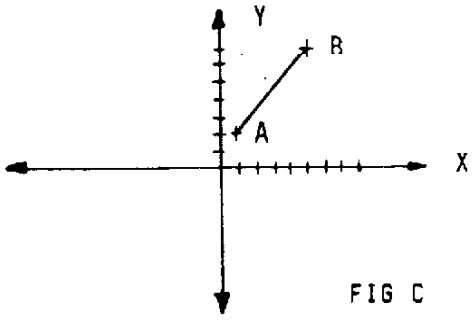


FIG C

Point A is at (1,2) and point B is at (5,7). In order to connect the two points we need to determine the slope of the line. All that the slope means is the

$$\frac{\text{(change in Y)}}{\text{(change in X)}} = \frac{Y2-Y1}{X2-X1} = \frac{7-2}{5-1} = \frac{5}{4} = 1.25$$

thus the slope in FIG C is 1.25, all this means is for every time you increase X by one, you must increase Y by 1.25.
If you are having trouble understanding this, just use the line subroutine, you needn't fully understand it to use it.

If you have any two points on the screen and want to connect a line between them use PROGLST3. The form of the call is

```
CALL LINE(X1,Y1,X2,Y2)
```

where the first point has the position (X1,Y1) and the second point has the position (X2,Y2). The subroutine then determines the slope and fits the straigtest possible line,that can be made on the TI,between the two points.

If you think about it for a minute, all you need to define a square or rectangle is two of the diagonal points. This is fairly simple and you should be able to pick out what to do from the subroutine SQUARE. To call this subprogram, use the form

```
CALL SQUARE(X1,Y1,X2,Y2)
```

where (X1,Y1) and (X2,Y2) are diagonal corners of the square or rectangle to be drawn. Remember to merge PUTDOT at the end of SQUARE!!!

Now it is time to enter the realm of the circle. First a brief review of simple geometry and basic trig definitions. Figure D is a drawing of a circle plotted on the cartesian coordinate axis and centered about the point (0,0).

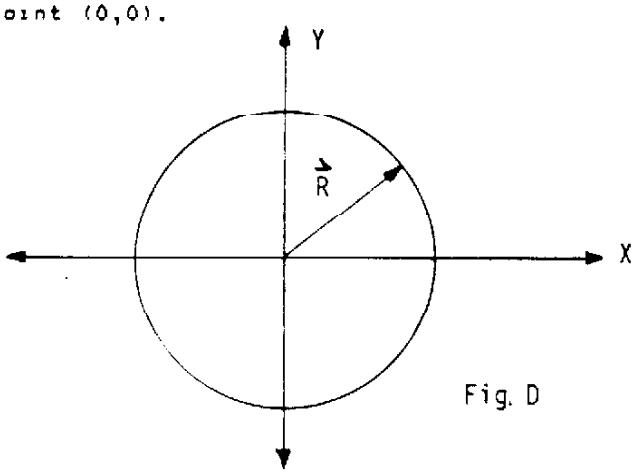


Fig. D

From the definition of a circle, we know that all points on the circle are the same distance from the center point of the circle. This distance is known as the radius and is drawn in figure D as a line connecting the center to a point on the circle.

This being established, we know the size and the location of a circle that we want to plot. Now we need an equation to do this. Many of you may remember that the equation of a circle can be given as:

$$X^2 + Y^2 = R^2$$

Where;

X= x-coordinate
Y= y-coordinate
R= radius of circle

You would normally solve the equation to make x the independent variable and y the dependent variable. But this leads to some rather cumbersome values and requires a great deal of effort.

If you suspect that there is a much simpler method, you are right. This is where the Trig come in. By using the basic definitions of the Sine and Cosine functions, we plot the point for each incremental angle.

Remembering that this computer like most computers works in radians instead of degrees,we will plot the corresponding point given the radius and the angle.

A full circle has 360 degrees which is the same as 2PI or 6.28 radians. Extended basic intrinsically defines PI=3.141592653. So we should use increments in our loop that correspond to logical values such as PI/36. A moments thought immediately reaveals that the smaller the increments, the more smooth the circle will be. However, since the TI has limited resolution, it is worthless to use very small incremental values since they would exceed the resolution of the computer.

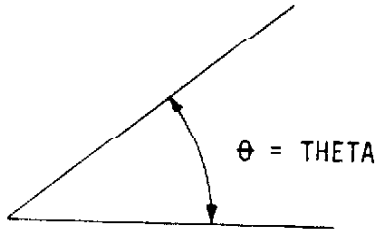
As a rule of thumb, the larger the circle, the smaller the incremental value should be. Thus a very large circle might best be drawn with a PI/72 incremental angle ,whereas a small circle might be best "fitted" with a PI/12 incremental angle. For reasons beyond the scope of this article, you should try to select the denominator of the incremental angle so that it is evenly divisible by 6.

If you need to review the Sine and Cosine functions, I refer you to any high school trigonometry book.

The algorythm for plotting a circle of raduis R and angle THETA is:

```
100 FOR THETA=0 TO 2*PI STEP PI/INC
110 X=R*COS(THETA)
120 Y=R*SIN(THETA)
130 CALL PUTDOT(X,Y)
140 NEXT THETA
```

THETA is just an commonnly used Greek letter to represent an angle.



DEFINITION OF THETA (θ)

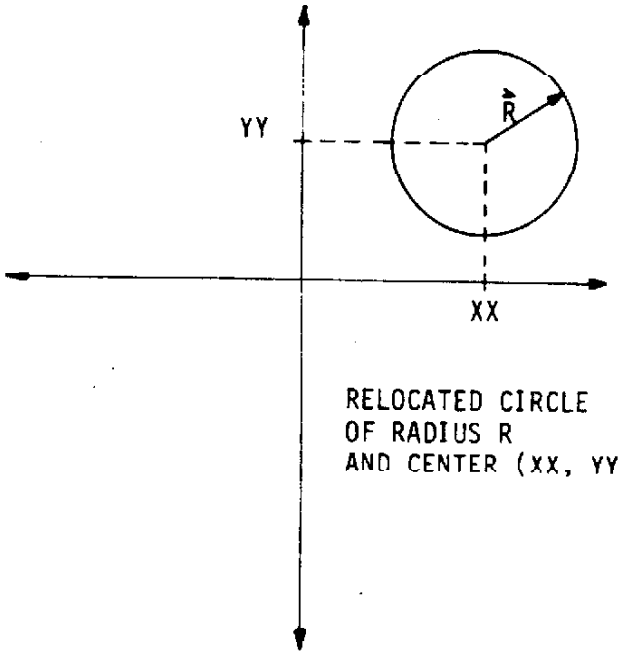
The above routine plots a circle at the center of the screen ,which is at location (0,0) in cartesian coordinates. If you would like to plot a circle that is centered around any given point (XX,YY) just include the following to the above routine.

```
121 X=X+XX :: Y=Y+YY
```

Now, lets take a quick look at what we have to define a circle.

R= Raduis
INC= Incremental value of THETA
XX= X-offset
YY= Y-offset

By changing R we cange the size of the circle. We must then adjust INC to produce the smoothest fit for the given raduis. The best way to do this is simply trial and error. If we want the circle plotted in the center of the screen, we set XX and YY both to zero. When it is desired to plot the circle at with a center other that (0,0) we set XX and YY to the values of the new center so that the circle will be plotted around point (XX,YY).



RELOCATED CIRCLE
OF RADIUS R
AND CENTER (XX, YY)

I would like to close on a reccomendation.

Before you try to do anything in this article READ the ENTIRE article TWICE! I think that re-reading will greatly clarify any ambiguous concepts that my befuddle you the first time around.

Till Part 2

Darren Leonard PUG

```
100 SUB LINE(X1,Y1,X2,Y2)
110 SLOPE=(Y2-Y1)/(X2-X1)
120 FOR PARA=X1 TO X2
130 Y=SLOPE*PARA
140 X=PARA
150 CALL PUTDOT(X,Y)
160 NEXT PARA
170 SUBEND
```

Now, we have covered Points,Lines and Circles. I threw the squares in to illustrate how many common shape can be made from these basic elements. For example if any three points are given, a triangle can be drawn merely by 3 succesive calls to the LINE routine to connect all three points.

I would also like to point out that there are some new Extended Basic modules out that can produce lines by themselves. In general, they just have the same algorythm included on the cartridge. I have also read about single pixel graphics as well, but I have not had a chance to check into this, so I cannot give any opinion of it.

It is worth noting that the simple extended basic PUTDOT routine is limited and can be replaced by a good program such as DRAW-n-PLOT by Quality99 software. It allows you to use the bit map mode by clever manipulation of memory.

```
100 SUB SQUARE(X1,Y1,X2,Y2)
110 FOR BN=X1 TO X2::X=BN::Y=Y1
120 CALL PUTDOT(X,Y)
130 NEXT BN
140 FOR BN=X1 TO X2::X=BN::Y=Y2
150 CALL PUTDOT(X,Y)
160 NEXT BN
170 FOR BN=Y1 TO Y2::Y=BN::X=X1
180 CALL PUTDOT(X,Y)
190 NEXT BN
200 FOR BN=Y1 TO Y2::Y=BN::X=X2
210 CALL PUTDOT(X,Y)
220 NEXT BN
230 SUBEND
```

In part 2 of this article, I will go into rotating lines and squares along with some other shapes and interesting graphics. In part 3 I hope to dive into 3D to a limited degree so I hope that every one will be looking foward to

```
100 SUB CIRCLE(R,X1,Y1)
110 FOR THETA=0 TO 2*PI STEP PI/36
120 X=R*COS(THETA)
130 Y=R*SIN(THETA)
140 X=X+X1
150 Y=Y+Y1
160 NEXT THETA
170 SUBEND
```

155 Call Putdot(X,Y)

```
500 SUB PUTDOT(V,Q)
501 X=Q :: Y=V :: IF X<-125 OR X>125 THEN 532
502 IF Y<-95 OR Y>95 THEN 532
503 X=(95-X):: Y=INT(Y+125)
504 IF C=0 THEN C=143
505 W=INT((X-1)/8)+1 :: Z=INT((Y-1)/8)+1 :: X1=X-((W-1)*8):: Y1=Y-((Z-1)*8):: CA
LL GCHAR(W,Z,A):: A1=A :: IF A<32 THEN A=32 :: A1=32
506 CALL CHARPAT(A,A$):: P=X1*8-8+Y1 :: Q=INT(P/4.06)+1 :: IF Y1>4 THEN Y1=Y1-4
507 IF A<>32 THEN C=C+1 ELSE A=C :: IF C<34 THEN 32767
508 B$=SEG$(A$,Q,1):: B=ASC(B$):: IF B<65 THEN B=B-47 ELSE B=B-54
509 ON B GOSUB 515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530
510 D$=SEG$(C$,Y1,1):: IF D$="1" THEN IF A1=32 THEN 532 ELSE 531
511 B=B+2^(ABS(Y1-4))
512 ON B GOSUB 515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530
513 A$=SEG$(A$,1,Q-1)&B$&SEG$(A$,Q+1,16-Q)
514 CALL CHAR(A,A$):: CALL HCHAR(W,Z,A):: GOTO 531
515 C$="0000" :: B$="0" :: RETURN
516 C$="0001" :: B$="1" :: RETURN
517 C$="0010" :: B$="2" :: RETURN
518 C$="0011" :: B$="3" :: RETURN
519 C$="0100" :: B$="4" :: RETURN
520 C$="0101" :: B$="5" :: RETURN
521 C$="0110" :: B$="6" :: RETURN
522 C$="0111" :: B$="7" :: RETURN
523 C$="1000" :: B$="8" :: RETURN
524 C$="1001" :: B$="9" :: RETURN
525 C$="1010" :: B$="A" :: RETURN
526 C$="1011" :: B$="B" :: RETURN
527 C$="1100" :: B$="C" :: RETURN
528 C$="1101" :: B$="D" :: RETURN
529 C$="1110" :: B$="E" :: RETURN
530 C$="1111" :: B$="F" :: RETURN
531 C=C-1 :: IF C=0 THEN C=1
532 SUBEND
```

TIPS FROM THE TIGERCUB

#38

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 PPH. 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.

```
##############################
#                            #
#  Tips from the Tigercub    #
#  Vol. 3 is now ready.      #
#  Another 62 programs,      #
#  routines, tips, tricks.   #
#  from Nos. 25 thru 32.     #
#  Also $15 postpaid. Any    #
#  two Tips disks $27 or     #
#  all 3 for $35 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, PROGRAM-TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION 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 have discovered a rare bug in the 28-Column Converter, published in Tips #18, which will cause an I/O 25 ERROR if the very last line of the program being converted happens to have exactly 80 characters. You can fix it by adding a line -

```
215 IF EOF(1)=1 THEN 260
```

There is also a rare bug in the SIDEWAYS subroutine on my Nuts & Bolts #2 disk, which prevents turning some redefined character sets sideways. If you are one of those who BOUGHT that disk from me, you can fix it by changing the L=LEN(B$) in line 21639 to L=64.

I was in too much of a hurry to go fishing when I put the last couple of Tips together. In the Gordian Knot in Tips #35, I left out some essential instructions. Please add -

```
131 DISPLAY AT(11,1):" When
you cross your track,"!"pres
s D to go over, U to go"!"un
der, C to go across."
```

To make that fit, you will have to change the DISPLAY AT in line 130 to (8,1), in line 140 to (15,1) and in line 150 to (20,1), also the ACCEPT At in 160 to (20,11). And this change will prevent a lockup when you reach a border -

```
200 D=D-1 :: IF ABS(D-D2)=2
OR R+(D=1)=0 OR R-(D=3)=25 O
R C+(D=4)=2 OR C-(D=2)=31 TH
EN 180 :: GOSUB 510 :: IF D<
>D2 THEN GOSUB 450
```

I wrote the dulcimer music in Tips #36 in Basic, but I forgot to test it in Basic. It actually runs much better in Extended Basic, but will run fairly well in Basic if you delete the delays in lines 280 and 300.

If you liked the ESCHER ART in Tips #37, these modifications will improve it considerably -

```
110 DISPLAY AT(12,1):"Press
-": :" Q for new pattern":"
B to change background":" F
to change foreground":" R to
 reverse colors": : :"Any ke
y to start"
280 A=INT(6*RND+3):: H=INT(2
4/A):: RX=24-H*A :: HC=INT(2
8/A):: CX=28-HC*A :: W=ABS(H
C/2=INT(HC/2))-(RX>8):: DIM
M(8,8):: FOR P=1 TO A
330 IF K<>66 THEN 346
340 BC=BC+1+(BC=16)*15 :: IF
 BC=F THEN 340 ELSE 347
346 IF K<>78 THEN 360 :: F=F
+1+(F=16)*15 :: IF F=BC THEN
 346
347 FOR S=7 TO 14 :: CALL CO
LOR(S,F,BC):: NEXT S :: GOTO
 310
350 ! **DELETED LINE **
360 IF K<>ASC("R")THEN 310 :
: T=F :: F=BC :: BC=T :: GOT
O 347
600 GOSUB 900 :: FOR T=1 TO
A :: DISPLAY AT(R-1+T,C):M$(
V,T):: NEXT T :: NEXT C
601 IF CX>0 THEN AA=A :: GOS
UB 800
605 GOSUB 1000 :: NEXT R
606 IF RX=0 THEN 610
607 GOSUB 1000 :: FOR C=1 TO
 A*HC STEP A :: GOSUB 900 ::
 FOR T=1 TO RX :: DISPLAY AT
(R-1+T,C):M$(V,T):: NEXT T :
: NEXT C
608 IF CX>0 THEN AA=RX :: GO
SUB 800
800 GOSUB 900 :: FOR T=1 TO
AA :: DISPLAY AT(R-1+T,C):SE
G$(M$(V,T),1,CX):: NEXT T :
: RETURN
900 V=V+1+(V=4)*4 :: RETURN
1000 V=V+W :: V=V+(V>4)*4 ::
 RETURN
```

I had a letter from a teacher who was using the PRK module to keep student grades, and wanted to know how to average them. It can be done, but is so impractical that I wrote this program. While I was at it, I speeded up the loading and saving to cassette greatly by converting the grades to an ASCII string and combin-the student's name and all grades into one record.

```
100 DIM N$(50),T(50,20)
110 CALL CLEAR
120 PRINT "       TEACHER'S
HELPER"! : : :
130 REM  - by Jim Peterson
140 PRINT "(1)CREATE A FILE?
"!"(2)ADD TO FILE?"!"(3)LOAD
 A FILE?"!"(4)SAVE A FILE?"!
"(5)PRINT A FILE?"
150 PRINT "(6)CORRECT A FILE
?"!"(7)COMPUTE AVERAGES?"!"(
8)QUIT?"
160 CALL KEY(0,K,S)
```

```
17$ IF (S=$)+(K<49)+(K)56)TH
EN 16$
18$ ON K-48 GOTO 19$,25$,61$
,88$,38$,99$,112$,151$
19$ X=$
20$ INPUT "SUBJECT? ";S$
21$ GOSUB 137$
22$ INPUT "TEST #? ";N
23$ GOSUB 144$
24$ GOTO 14$
25$ PRINT :;:"(1)ADD NAMES?"
:"(2)ADD GRADES?"
26$ CALL KEY($,K,S)
27$ IF (S=$)+(K<49)+(K)5$)TH
EN 26$
28$ ON K-48 GOTO 29$,31$
29$ GOSUB 137$
30$ GOTO 14$
31$ INPUT "TEST #? ";Q
32$ IF T(1,Q)=$ THEN 35$
33$ PRINT :;:"TEST #";STR$(Q
);" ALREADY RECORDED"
34$ GOTO 14$
35$ N=Q
36$ GOSUB 144$
37$ GOTO 14$
38$ CALL CLEAR
39$ PRINT "OUTPUT TO":"(1)SC
REEN?":"(2)PRINTER?"
40$ CALL KEY($,K,S)
41$ IF (S=$)+(K<49)+(K)5$)TH
EN 40$
42$ IF K=49 THEN 46$
43$ INPUT "PRINTER DESIGNATI
ON? ":P$
44$ OPEN #2:P$
45$ F@=2
46$ PRINT "PRESS ANY KEY TO
PAUSE"; :
47$ PRINT #F@:S$; :
48$ FOR J=1 TO X
49$ PRINT #F@:"";N$(J)&" ";T
AB(18);
50$ FOR K=1 TO HN
51$ PRINT #F@:T(J,K);
52$ NEXT K
53$ CALL KEY($,K,S)
54$ IF S<>$ THEN 53$
55$ NEXT J
56$ PRINT #F@
57$ IF F@=$ THEN 14$
58$ F@=$
59$ CLOSE #2
60$ GOTO 14$
61$ PRINT :;:"(1)CASETTE?":"
(2)DISK?"
62$ CALL KEY($,K,S)
63$ IF (S=$)+(K<49)+(K)5$)TH
EN 62$
64$ ON K-48 GOTO 65$,67$
65$ OPEN #2:"CS1",INPUT ,FIX
ED
66$ GOTO 69$
67$ INPUT "FILENAME? DSK":F$
68$ OPEN #2:"DSK"&F$,INPUT
69$ INPUT #2:X,HN,S$
70$ FOR J=1 TO X
71$ INPUT #2:K$
72$ N$(J)=SEG$(K$,1,POS(K$,C
HR$(255),1)-1)
73$ K$=SEG$(K$,POS(K$,CHR$(2
55),1)+1,255)
74$ FOR K=1 TO HN
75$ T(J,K)=ASC(SEG$(K$,K,1))
-5$
76$ NEXT K
77$ NEXT J
78$ CLOSE #2
79$ GOTO 14$
80$ PRINT :;:"(1)CASETTE?":"
(2)DISK?"
81$ CALL KEY($,K,S)
82$ IF (S=$)+(K<49)+(K)5$)TH
EN 81$
83$ ON K-48 GOTO 84$,86$
84$ OPEN #2:"CS1",OUTPUT,FIX
ED
85$ GOTO 88$
86$ INPUT "FILENAME? DSK":F$
87$ OPEN #2:"DSK"&F$,OUTPUT
88$ PRINT #2:X:HN:S$
89$ FOR J=1 TO X
90$ K$=""
91$ FOR K=1 TO HN
92$ K$=K$&CHR$(T(J,K)+5$)
93$ NEXT K
94$ PRINT #2:N$(J)&CHR$(255)
&K$
95$ K$=""
96$ NEXT J
97$ CLOSE #2
98$ GOTO 14$
99$ CALL CLEAR
100$ INPUT "STUDENT'S NAME?
":Q$
101$ FOR J=1 TO X
102$ IF N$(J)=Q$ THEN 106$
103$ NEXT J
104$ PRINT :;:"NAME NOT FOUN
D"; :
105$ GOTO 14$
106$ INPUT "CORRECT WHICH TE
ST? ($ TO QUIT) ":C
107$ IF C=$ THEN 111$
108$ PRINT :;:N$(J);"'S TEST
 #";STR$(T(J,C)); :
109$ INPUT "CORRECT TO? ":T(
J,C)
110$ GOTO 106$
111$ GOTO 14$
112$ CALL CLEAR
113$ PRINT "OUTPUT TO":"(1)S
CREEN?":"(2)PRINTER?"
114$ CALL KEY($,K,S)
115$ IF (S=$)+(K<49)+(K)5$)T
HEN 114$
116$ IF K=49 THEN 120$
117$ INPUT "PRINTER DESIGNAT
ION? ":P$
118$ OPEN #2:P$
119$ F@=2
120$ PRINT #F@:S$
121$ FOR J=1 TO X
122$ PRINT #F@:N$(J);" AVERA
GE ";
123$ FOR K=1 TO HN
124$ TT=TT+T(J,K)
125$ NEXT K
126$ AV=TT/HN
127$ TAV=TAV+AV
128$ PRINT #F@:AV
129$ TT=$
130$ NEXT J
131$ PRINT #F@:"CLASS AVERAG
E ";TAV/X
132$ TAV=$
133$ IF F@=$ THEN 136$
134$ F@=$
135$ CLOSE #2
136$ GOTO 14$
137$ PRINT :;:"STUDENT'S NAM
ES - ":"type END when finish
ed": :
138$ X=X+1
139$ M$="NAME #"&STR$(X)&" "
140$ INPUT M$:N$(X)
141$ IF N$(X)<>"END" THEN 13
8$
142$ X=X-1
143$ RETURN
144$ FOR J=1 TO X
145$ M$=N$(J)&"'S GRADE? "
146$ INPUT M$:T(J,N)
147$ NEXT J
148$ IF N<HN THEN 24$
149$ HN=N
150$ RETURN
151$ END
```

The reason that 5$ is added to the value in line 92$, before saving, and subtracted again in line 75$ after loading, is because of a quirk of the computer that I don't recall seeing in print anywhere. Did you know that INPUT will read a string beginning with ASCII $, 2, 4, 7, 18, 12, 14, 18, 28, 26, 27, 31, 32, or 44 as a null string (a blank), and will drop these characters at the end of a string? And ASCII 32 will be dropped at the beginning or end of a string. And ASCII $ within a string, or ASCII 34 anywhere, will crash, while ASCII 44 within a string will lose the rest of the string. I should have known what ASCII $, 32 (the space), 34 (quotes) and 44 (comma) would do, but why the others?

LINPUT will accept anything, of course, but I wanted to keep this in BASIC for the teachers who are struggling along without the XBasic module or disk drive.

Chick De Marti published in LA 99ers TOPICS the surprising discovery that PRINT USING and DISPLAY USING can read the IMAGE format from a variable, array or string!

Which led me to some fooling around -

```
10$ !PRINT USING DEMO by Jim
 Peterson, based on a discov
ery by Chick De Marti
11$ CALL CLEAR :: RANDOMIZE
:: CALL SCREEN(5):: FOR S=2
TO 14 :: CALL COLOR(5,S,S)::
 NEXT S
12$ N=INT(13*RND+1):: C$=CHR
$(8*N+32-(N=4)*11)
13$ FOR J=N TO 12 :: A$=RPT$
(" ",J)&"*"&RPT$(" ",26-J*2)
&"*" :: PRINT USING A$:C$,C$
 :: NEXT J
14$ FOR J=12 TO N STEP -1 ::
 A$=RPT$(" ",J)&"*"&RPT$(" "
,26-J*2)&"*" :: PRINT USING
A$:C$,C$ :: NEXT J :: GOTO 1
2$
```

Here is one last Tigercub challenge. What is the longest possible one-liner? And what is the longest possible one-liner that actually does something?

MEMORY FULL

Jim Peterson

TIPS FROM THE TIGERCUB

#39

Copyright 1986

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.

```
############################
#                          #
# Tips from the Tigercub   #
# Vol. 3 is now ready.     #
# Another 62 programs,     #
# routines, tips, tricks.  #
# from Nos. 25 thru 32.    #
# Also $15 postpaid. Any   #
# two Tips disks $27 or    #
# all 3 for $35 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 108 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, PROGRAM-TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION 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!

Answer to last month's challenge - for the longest possible one-liner, run the following "program to write a program" -

```
100 OPEN #1:"DSK1.LONG",VARIABLE 163,OUTPUT
110 FOR J=1 TO 79 :: M$=M$&CHR$(149)&CHR$(130):: NEXT J :: M$=CHR$(254)&CHR$(254)&M$&CHR$(149)&CHR$(0):: PRINT #1:M$ :: PRINT #1:CHR$(255)&CHR$(255):: CLOSE #1
```

Then enter NEW, then MERGE DSK1.LONG, then LIST - over 34 lines long! But that one doesn't do anything, so try this -

```
100 OPEN #1:"DSK1.LONG",VARIABLE 163,OUTPUT
110 FOR J=1 TO 52 :: M$=M$&CHR$(162)&"X"&CHR$(130):: NEXT J :: M$=CHR$(254)&CHR$(254)&M$&CHR$(162)&"X"&CHR$(0):: PRINT #1:M$ :: PRINT #1:CHR$(255)&CHR$(255):: CLOSE #1
```

Again enter NEW, and MERGE DSK1.LONG, then RUN. You'll get a message BREAKPOINT IN 32510 (don't ask me why! Can anyone tell me?) but just enter RUN again. Then LIST it - over 24 lines long!

Explanation? Programs are saved in token code similar to MERGE format code. The maximum length of a record is 163 bytes - which is why MERGE files are D/V 163. The token for RANDOMIZE is ASCII 149, for the double colon is 130. Repeating that 79 times takes only 158 bytes, plus one more RANDOMIZE, the two-byte tokenized line number and the mandatory ASCII 0 to end the record, totals 162.

Here's a spooky one for Hallowe'en -

```
100 CALL CLEAR :: CALL MAGNIFY(4):: CALL SCREEN(2) ! The Blob by Jim Peterson
110 CALL CHAR(96,RPT$("3C7EFFFFFFFF7E3C",4)):: J=-1
120 FOR L=1 TO 28 :: CALL SPRITE(#L,96,16,L*4+20,10,0,L+8):: NEXT L
130 FOR L=1 TO 28 :: CALL MOTION(#L,0,L*J):: NEXT L
140 J=J*-1 :: GOTO 130
```

Wes Johnston published an unusual sprite 2-liner in the Charleston Area 99ers newsletter. It is based on a CALL LOAD which freezes all sprite motion until they are turned loose by another CALL LOAD -

```
100 R=PI*2/28 :: CALL CLEAR :: CALL SCREEN(2):: CALL INIT :: CALL LOAD(-31806,96):: FOR I=1.TO 28 :: CALL SPRITE(#I,46,16,96,128,COS(I*R)*10,SIN(I*R)*10):: NEXT I
110 CALL LOAD(-31806,0):: GOTO 110
```

You might like to try adding my "jewels" to that -

```
100 FOR CH=33 TO 60 :: FOR A=1 TO 4 :: X=INT(8*RND+1):: T$=SEG$("18243C425A667E81",X*2-1,2):: A$=A$&T$ :: B$=T$&B$ :: NEXT A :: CALL CHAR(CH,A$&B$):: A$,B$="" :: NEXT CH
110 R=PI*2/28 :: CALL CLEAR :: CALL SCREEN(2):: CALL INIT :: CALL LOAD(-31806,96):: FOR I=1 TO 28 :: CALL SPRITE(#I,32+I,INT(14*RND+3),96,128,COS(I*R)*10,SIN(I*R)*10):: NEXT I
120 CALL LOAD(-31806,0):: GOTO 120
```

Also try CALL MAGNIFY(2)

And, here is a companion program to the TAKE AWAY in Tips #35 -

```
100 CALL CLEAR :: CALL TITLE(5,"ADD & CARRY")!by Jim Peterson
110 DISPLAY AT(3,10):"COPYRIGHT":TAB(11);"TIGERCUB SOFTWARE":TAB(10);"FOR FREE":TAB(10);"DISTRIBUTION":TAB(11);"SALE PROHIBITED"
120 CALL PEEK(-28672,A@):: IF A@=0 THEN 160
130 DATA FINE,NO,GOOD,UHOH,RIGHT,TRY AGAIN,YES,THAT IS NOT RIGHT
140 FOR J=1 TO 4 :: READ RIGHT$(J),WRONG$(J):: NEXT J
150 FOR D=1 TO 1000 :: NEXT D :: CALL DELSPRITE(ALL)
160 CALL CLEAR :: CALL CHAR(95,"FFFF"):: CALL MAGNIFY(2) :: RANDOMIZE :: CALL SCREEN(14):: FOR SET=5 TO 8 :: CALL COLOR(SET,16,1):: NEXT SET
170 CALL CHAR(120,"E7004200180007E0000E700420099423CE700420099423C00E7004218003C4200")
180 CALL CHAR(124,"0E00401000708007000208000E01000")
190 DISPLAY AT(3,8):"ADD AND CARRY" :: CALL CHAMELEON
200 CALL COLOR(14,2,2):: CALL HCHAR(4,4,143,2):: CALL HCHAR(5,4,143,2):: CALL SPRITE(#25,120,11,25,25)
```

```
21$ T=T+1 :: IF T=6 THEN T=$
 :: GOTO 25$
22$ Z=INT(8*RND+2):: IF Z=Z2
 THEN 22$ ELSE Z2=Z
23$ Y=INT(Z*RND):: IF Y=Y2 T
HEN 23$ ELSE Y2=Y :: X=Z-Y
24$ N=1 :: GOSUB 47$ :: GOTO
 21$
25$ T=T+1 :: IF T=11 THEN T=
$ :: GOTO 29$
26$ X=INT(1$*RND):: IF X=X2
THEN 26$ ELSE X2=X
27$ Y=INT(1$*RND):: IF Y=Y2
OR X+Y<1$ THEN 26$ ELSE Y2=Y
 :: Z=X+Y
28$ N=1 :: GOSUB 47$ :: GOTO
 25$
29$ T=T+1 :: IF T=11 THEN T=
$ :: GOTO 33$
30$ X=INT(9$*RND+1$):: IF X=
X2 THEN 30$ ELSE X2=X
31$ Y=INT(9$*RND+1$):: IF Y=
Y2 THEN 31$ ELSE Y2=Y :: Z=X
+Y
32$ N=2 :: GOSUB 47$ :: GOTO
 29$
33$ X=INT(9$$*RND+1$$):: IF
X=X2 THEN 33$ ELSE X2=X
34$ Y=INT(9$$*RND+1$$):: IF
Y=Y2 THEN 34$ ELSE Y2=Y :: Z
=X+Y
35$ N=3 :: GOSUB 47$ :: GOTO
 33$
36$ R=96 :: CC=96 :: FOR J=1
 TO N :: CALL SPRITE(#J,48+A
(J),11,R,CC):: CC=CC+16 :: N
EXT J
37$ R=116 :: CC=96 :: FOR J=
1 TO N :: CALL SPRITE(#4+J,4
8+B(J),11,R,CC):: CC=CC+16 :
: NEXT J
38$ CALL HCHAR(18,12,95,N*3)
:: CC=CC-16 :: CALL SPRITE(#
22,43,16,R,8$):: RETURN
39$ R=14$ :: FOR J=LEN(STR$(
Z))TO 1 STEP -1 :: CALL SPRI
TE(#2$,63,11,R,CC)
40$ CALL KEY(3,K,ST):: IF ST
<1 OR K<48 OR K>57 THEN CALL
 PATTERN(#2$,32):: CALL PATT
ERN(#2$,63):: GOTO 40$
41$ CALL DELSPRITE(#2$):: CA
LL SPRITE(#12+J,K,11,R,CC)
42$ IF K-48<>C(J)THEN GOSUB
48$ :: CALL DELSPRITE(#12+J)
:: CALL SPRITE(#2$,63,11,R,C
C):: GOTO 40$
43$ IF A(J-N)+B(J-N)>9 THEN
CALL SPRITE(#28,49,16,8$,CC-
16)
44$ CC=CC-16 :: NEXT J :: GO
SUB 51$ :: RETURN
45$ FOR J=1 TO LEN(STR$(X))
: :: A(J)=VAL(SEG$(STR$(X),J
,1)):: NEXT J :: FOR J=1 TO
LEN(STR$(Y)):: B(J)=VAL(SEG$
(STR$(Y),J,1)):: NEXT J
46$ FOR J=1 TO LEN(STR$(Z)):
: C(J)=VAL(SEG$(STR$(Z),J,1)
):: NEXT J :: N=LEN(STR$(Z))
-LEN(STR$(X)):: RETURN
47$ GOSUB 45$ :: GOSUB 36$ :
: GOSUB 39$ :: FOR D=1 TO 2$
$ :: NEXT D :: CALL DELSPRIT
E(ALL):: DISPLAY AT(18,1)::
CALL CHAMELEON :: CALL SPRIT
E(#25,12$,11,25,25):: RETURN
48$ DATA 123,124,125,123,124
,125,123,12$
49$ IF A$=$ THEN 5$$ :: CALL
 SAY(WRONG$(INT(4*RND+1)))
5$$ RESTORE 48$ :: FOR JJ=1
TO 8 :: READ P :: CALL PATTE
RN(#25,P):: XX=2^25$ :: NEXT
 JJ :: RETURN
51$ DATA 121,122,121,122,121
,122
52$ IF A$=$ THEN 53$ :: CALL
 SAY(RIGHT$(INT(4*RND+1)))
53$ RESTORE 51$ :: FOR JJ=1
TO 6 :: READ P :: CALL PATTE
RN(#25,P):: XX=2^25$ :: NEXT
 JJ :: RETURN
54$ SUB CHAMELEON
55$ M$="18$$665AC342DB667E18
81$$995AC3A5E78142BD24DB66$$
8142992499?E5AC3A53C2418$$FF
DB5AFF7EFF$$991881$$66$$18"
56$ RANDOMIZE :: CALL CHAR(1
28,SEG$(M$,INT(43*RND+1)*2-1
,16)):: X=INT(14*RND+3)
57$ Y=INT(14*RND+3):: IF Y=X
 THEN 57$ :: CALL COLOR(13,X
,Y)
58$ CALL HCHAR(1,2,128,3$)::
 CALL HCHAR(24,2,128,3$):: C
ALL VCHAR(1,31,128,96):: SUB
END
59$ SUB CHAMWIPE
6$$ T=T+1+(T=2)*2 :: ON T GO
TO 61$,62$
61$ CALL VCHAR(1,3,128,768):
: GOTO 63$
62$ CALL HCHAR(1,1,128,768)
63$ CALL CLEAR :: SUBEND
64$ SUB TITLE(S,T$)
65$ CALL SCREEN(S):: L=LEN(T
$):: CALL MAGNIFY(2)
66$ FOR J=1 TO L :: CALL SPR
ITE(#J,ASC(SEG$(T$,J,1)),J+1
*-(J+1=5)+(J+1=5+13)+(J>14)*1
3,J*(17$/L),1$+J*(2$$/L))::
NEXT J
67$ SUBEND

  A mathematical curiosity -
1$$ !MAGIC NINES by Jim Pete
rson
11$ CALL CLEAR
12$ INPUT "TYPE ANY 3-DIGIT
NUMBER OF 3 DIFFERENT DIGITS
 ":N :: IF N<>INT(N)OR N>999
 OR N<$ THEN 12$
13$ N$=STR$(N):: IF N<1$$ TH
EN N$="$"&N$
14$ IF SEG$(N$,1,1)=SEG$(N$,
2,1)OR SEG$(N$,1,1)=SEG$(N$,
3,1)OR SEG$(N$,2,1)=SEG$(N$,
3,1)THEN PRINT ")))THREE DIF
FERENT DIGITS<<," :: GOTO 12
$
15$ PRINT :: N2$="" :: FOR J
=1 TO 3 :: N2$=SEG$(N$,J,1)&
N2$ :: NEXT J :: N2=VAL(N2$)
:: D=ABS(N-N2)
16$ PRINT N$;" BACKWARDS IS
";N2$; :
17$ N3=ABS(N-N2):: N3$=STR$(
N3):: IF N3<1$$ THEN N3$="$"
&N3$
18$ IF N>N2 THEN PRINT N$;"
MINUS ";N2$;" EQUALS ";N3$;
:ELSE PRINT N2$;" MINUS ";N$
;" EQUALS ";N3$; :
19$ FOR J=1 TO 3 :: N4$=SEG$
(N3$,J,1)&N4$ :: NEXT J
2$$ PRINT N3$;" BACKWARDS IS
 ";N4$; :N3$;" PLUS ";N4$;"
IS 1$89": :"I KNEW THAT WOUL
D BE THE"; :"ANSWER!": :"LIS
T THE PROGRAM AND SEE!"
21$ !!!!!!!!!!!!!!!!!!!!!!!!
22$ !  THE ANSWER WILL BE  !
23$ !        1$89         !
24$ !!!!!!!!!!!!!!!!!!!!!!!!

1$$ DISPLAY AT(8,1$)ERASE AL
L:"SHENANDOAH": : :"  Across
 the wide Missouri": : : : :
 : : : : : : :"programmed by
 Jim Peterson"
11$ FOR D=1 TO 1$$$ :: NEXT
D :: CALL CLEAR :: DIM S(24)
:: RANDOMIZE :: M$="4218$$5A
$$7E9981$$5A24DBC31824243C5A
7EA5661$$$3CDD66BD3CA5421$7E
5AC324425A18A5186681$$811$7E
423CB$DDBC3" :: R=1
12$ FOR CH=4$ TO 136 STEP 8
13$ CALL CHAR(CH,SEG$(M$,INT
(43*RND+1)*2-1,16)):: CALL H
CHAR(R,1,CH,64):: R=R+2*ABS(
R<23)
14$ NEXT CH :: R=$ :: FOR SE
T=2 TO 14 :: X=INT(14*RND+2)
15$ Y=INT(14*RND+2):: IF Y=X
 THEN 15$
16$ CALL COLOR(SET,X,Y)
17$ NEXT SET :: CALL CLEAR :
: CALL COLOR(1,5,5):: CALL V
CHAR(1,29,1,192):: CALL SCRE
EN(16):: F=262 :: FOR N=$ TO
 23 :: S(N)=INT(F*1.$5946349
4^N):: CALL SOUND(-999,S(N),
$)
18$ NEXT N
19$ DATA 2,1,1,1,6,1,1,1,6,2
,6,1,1,1,6,1,8,8,1,1$,1$,1,1
1,11,1,15,6,3,13,6,2,13,11
2$$ DATA 1,1$,1$,1,17,17,4,1
5,11,1,11,15,1,13,13,1,15,11
,1,13,13,1,1$,1$,3,13,1$
21$ DATA 2,13,13,2,13,1$,1,1
5,1$,1,1$,15,2,15,15,1,15,1$
,1,1$,1$,1,13,13,1,1$,1$
22$ DATA 1,8,3,3,6,3,2,6,6,2
,8,8,4,1$,1,1,1$,6,1,6,6,1,1
8,1$,1,15,15
23$ DATA 2,13,1,2,13,5,2,13,
1$
24$ DATA 1,6,6,1,8,8,6,1$,6,
2,3,3,2,8,5,1,8,1,3,6,1,7,6,
1
25$ A=1 :: B=1 :: E=5
26$ FOR J=1 TO 144 STEP 3 ::
 CALL HCHAR(A,E,32,T*4):: CA
LL HCHAR(A+1,E,32,T*4):: CAL
L HCHAR(B,E,32,T*4):: CALL H
CHAR(B+1,E,32,T*4):: READ T,
A,B :: E=17-T*2
27$ CALL HCHAR(A,E,32+INT((A
+1)/2)*8,T*4):: CALL HCHAR(A
+1,E,32+INT((A+1)/2)*8,T*4):
: CALL HCHAR(B,E,32+INT((B+1
)/2)*8,T*4)
28$ CALL HCHAR(B+1,E,32+INT(
(B+1)/2)*8,T*4):: FOR D=1 TO
 T :: CALL SOUND(-999,S(A),$
,S(B),7)
29$ NEXT D
3$$ NEXT J :: LL=$ :: FOR SE
T=2 TO 14 :: X=INT(15*RND+2)
31$ Y=INT(15*RND+2):: IF Y=X
 THEN 31$
32$ CALL COLOR(SET,X,Y):: CA
LL SOUND(-999,S(6),LL,S(1),L
L):: LL=LL+2
33$ NEXT SET :: RESTORE :: G
OTO 26$
)))))))MEMORY FULL(((((((((
```