WORDPLAY

The PUNN Newsletter Portland, Oregon

June 1989-Volumn 8-No. 6

Ø36

8906

What's Inside

Club Officers

President	Al Kinney 640-5860
Vice-President	Don Barker 223-1749
Secretary	Mary Durham 652-2646
Treasurer	Ashley Read 775-1210

PUNN Staff

Librarians	Keith Fast 777-1531
	Ted Peterson 244-1587
	Jim Thomas 284-2425
Hardcopy	Mike Calkins 636-1839
Program Chairman	Ted Peterson 244-1587
Workshop Chairman	Chuck Neal 642-7292
Membership Chairman	Terry Priest 649-3934
Newsletter Editor	Charles Ball 639-0466

16576 SW Matador Lane-King City, OR 97224

BBS Committee

Chairman:	Al Kinney 640-5860
	Ron Mayer 232-7363
	Mike King 357-4413
BBS Phone Number	503/233-6804

MURPHY'S LAW:

If you try to please everybody, nobody will like it.

News and Views

Twelve officers and members attended the Board meeting on May 16th—It was held at Walt Morey's apartment and we all got to look at some of his electronic paraphernalia—Also the goodies consisted of Strawberry Shortcake—Thanks Walt!—Plans are in the works for the 3rd Annual PUNN Picnic—It will be Tuesday August 1st at the Milwaukie Elks picnic grounds, so put it on your calendar now—Jim Luque, who has appeared several times before our meeting, will be on hand at the June meeting with a new program to demonstrate—Jim's presentations are always interesting, so you'll want to see this—Keith Fast will be conducting the program for June and his subject will be the new TI Base—The workshop will be a further look at how to use Funnelweb and how to configure it to your own needs—The much awaited program "Press" has still not made its appearance, but a recent notice said it will not be much longer--When it is available we plan to demonstrate it at an early meeting—Don Barker, President of the Columbia-Willamette Chapter of the Merchant Marines, was the Master of Ceremonies at the dedication of the Memorial Plaque at Riverfront Park—Several PUNN members attended this long time coming memorial to those who gave their lives at sea during World War II—It seems the 'Motor-Bike' season is upon us and Keith Fast, Al Kinney and Mike King are hitting the road—Are there any others out there who ride?--From what we hear these machines are not cheap—Rich Gilbertson announces that WINDYXB, Version 1.1 will be available in September with many updated features including a screen dump to disk and a disk dump to screen--Look for a review of this program soon in Micropendium—Your Editor will be attending his 50th high school graduation reunion next month—Lincoln High School in Seattle (now closed) is where it happened--Interesting event and it co-incides with the 100th anniversary of the State of Washington--Al Kinney reports we are getting near to installing a hard disk on the BBS—Remember your support of the Library helps to make purchases like this—

Obstacles and Goals

"Obstacles are what you see when you take your eyes off the goal."

How many of us have experienced this: we begin working toward a goal, but then we get sidetracked by the obstacles we all encounter in our everyday routines? Later we find that through our long and hard concentration on the obstacles, our work is no longer directed toward our goal, and in fact these efforts have prevented us from accomplishing our real goal!

Let's take a look at an experiment that demonstrates this point, done with of all things, bees and flies. An equal number of each were placed in an open glass bottle. Their goal was to get free; their obstacle the bottle's glass. The experimenters knew that bees are smarter than flies, and so should get free quicker. The clear bottle was mounted horizontally with its base to a sunlit window, and its open neck facing toward the darkness of an interior room. In just 2 minutes, all the flies had flown out to freedom through the open neck of the bottle. But the "smarter" bees persisted in trying to get out through the bottle's sunlit base facing outdoors, until they collapsed from exhaustion. The bees had made the mistake of taking their eyes off their real goal. Reaching the source of light wasn't the goal -- getting free was!

Let's keep our goals clearly in mind. If we get bogged down in dealing with unimportant problems that have little or no direct connection with our goals, then we waste our talents, energies and resources. We can, by clearly reviewing our goals from time to time, see beyond obstacles that could sidetrack us to exhaustion!

Style A Line

Although written in the form of a TINYGRAM, this program is really a workhorse. If you ever need to print just a line or two such as a page header, an article or picture title, etc., then this program is for you.

Many of you are familiar with the programs PRINTALINE and PRINTSTYLE written by Ed Machonis. Ed has taken the best features of each and combined them into one short program.

STYLE A LINE is the result of this combination. One major revision was to change an INPUT statement to a LINPUT. No more need to enclose in quotes any text lines containing commas or leading spaces.

Using LINPUT requires that the program to runs in XBASIC. After some streamlining by deletion of unneeded features of PRINTALINE and the consolidation of statements into multi-statement lines, the resulting program was written in just 9 lines!

Don't let its brevity fool you. You can select many type styles and options that should work on an EPSON compatible printer. With a little work you could change the selections and DATA to suit your own purpose. Although there are better ways of doing it, you can even produce a right margin justified letter. Using Emphasized Pica, set the left margin at 13, and enter text. Two screen lines total 54 characters (since LINPUT uses 2 character spaces). Justify text by inserting spaces between words so that the second line ends at the screen edge. Of course, this program is certainly not a replacement for TI Writer, but it can serve a purpose at certain times.

Using the program is very easy. When RUN a menu is displayed for programming the printer. It is always best to select "1" first to clear the printer. If your printer doesn't support a master reset code, just turn the printer off and back on before selecting the various styles. You can combine various selections and then choose Option 10 to input text.

If you wish to change the type style, or do repeated printings of the same text, type "ZZZ" or "zzz" and you will return to the menu. Option 9 will do repeat printing of the same text and styles can be changed as required. When in text mode, pressing ENTER will print a blank line.

The program is real easy to type in. Watch the commas in line 10 and note the next to last DATA item is a lower case "L", not a figure 1.

```
1 ! ### STYLE A LINE ###
  a TINYGRAM by Ed Machonis
  BB-99ers, Bayside, NY
2 DIM P$(15):: FOR I=1 TO 15
  :: READ P$(I):: NEXT I
3 OPEN #1:"PIO",VARIABLE 132
4 CALL CLEAR :: PRINT "1 PIC
A/RESET","9 PRINT TEXT","2
ELITE","10 INPUT TEXT","3 EX
PANDED","11 SUPERSCRIPT","4
COMPRESSED","12 SUBSCRIPT"
5 INPUT "5 EMPHASIZED 13 I/
2 LINE SP6 ITALIC 14 L
MARGIN 137 D'BLE STRIK 15 R
MARGIN 678 UNDERLINE ?":I
6 P$(9)=" "&TEX$ :: PRINT #1
:CHR$(27)&P$(I):: IF I=4 THE
N PRINT #1:CHR$(27)&CHR$(15)
7 IF I()10 THEN 4
8 PRINT :"ENTER TEXT OR 'ZZZ
' FOR MENU" :: LINPUT TRY$
9 IF TRY$="ZZZ" OR TRY$="zzz
" THEN 4 ELSE TEX$=TRY$ :: P
RINT #1:TEX$ :: GOTO 8
10 DATA @,M,W1,,E,4,G,-1,,,S
0,SI,1,1,QC
```

July Meeting Changed

The July meeting date has been postponed, since the first Tuesday of July falls on the 4th this year. The proposed new date is on Tuesday July 11th pending confirmation with PGE.

We should know by the June meeting if this July 11th date has been OK'd. You might want to mark you calendar about this change.

Quiet Fans

If you ordered a quiet fan for your PE Box and have not yet picked it up, see Terry Priest at the June meeting. The fans are available along with an instruction sheet on how to install it for $5.00.

These are industrial type fans and should last a long time. Your editor has installed one and can attest to the reduced noise these fans put out.

Program-June

The program for June will be a demonstration and explanation of the new data base, TI BASE. This program is receiving a lot of attention in the TI world. It is easy to use and has features found in other computers.

Keith Fast, who has previously demonstrated PR BASE, will present the program.

Picnic in August

The Third Annual Punn Picnic will be held this year on August 1st.

As in the previous two years, the picnic will be held at the Milwaukie Elks picnic grounds.

Good food and drink will be furnished for a modest fee, with the club underwriting some of the cost. Swimming will be available in the Elks pool.

This is the time for all members to enjoy themselves with their families and club members. If you weren't able to attend in the past, ask those that did. Everyone had a great time.

We'll have more information about the picnic in next month's Word Play. In the meantime mark your calendar for Tuesday, August 1st.

The fun starts at 6:30pm and the food will be ready about 7:30.

Workshop-June

As a follow up to last month's program on configuring Funnel Web, the June Workshop will be devoted to a more detailed explanation of how to use the various programs that are added to the Funnel Web menu.

Chuck Ball will demonstrate how he uses the various programs to edit and publish Word Play. The techniques explained will include editing, hyphenating, formatting, columnizing and other uses to enhance your word processing.

"Moonlight Sonata"

Our musical offering for the month is "Moonlight Sonata". It was written by Kevin Noesner and typed in by our own Walt Morey. Besides the beautiful music, it is accompanied by some moving sprites to make it more interesting. Try typing it in, it's good practice, but if you don't have the time, pick it up from the library.

```
100 REM **KNICK KNACK**
          SOFTWARE
      Kevin Noesner
      2672 EASTCLEFT DR.
      COL, OH.  43221
110 CALL CLEAR
120 CALL SCREEN(2)
130 PRINT "
    **
     **
     **
    **
  .
140 PRINT "  **  **  ***  ***  *
   *          *  *  *  *  *  *  **
   *          *   *  *  *  *  *  *
**            *   *  ***  ***  *
 *
 *
150 PRINT "  *   ***  ***  *   *
 ***         *    *  *    *   *
  *          *    *  *  *  ****
   *         ***  ***  ***  *   *
   *.
160 PRINT "
             ***  ***  *   *   *
 ***  *      *    *  *  **  *  *
  *  * *     ***  *  *  *  **  ***
  *  ***       *  *  *  *  *  *  *
  *  *  *
170 PRINT "  ***  ***  *   *  *  *
   *  * *
       .
180 CALL COLOR(2,12,12)
190 FOR I=1 TO 10 :: CALL SP
RITE(#I,42,15,240,8,-AA/42+(
4#I),0):: NEXT I
200 FOR I=1 TO 10 :: CALL MO
TION(#I,0,AA/42+(2#I)):: NEX
T I
210 FOR A=1 TO 4
220 CALL SOUND(300,294,5,440
,3)
230 CALL SOUND(300,294,5,587
,3)
240 CALL SOUND(300,294,5,698
,3)
250 NEXT A
260 FOR B=1 TO 4
270 CALL SOUND(300,262,5,440
,3)
280 CALL SOUND(300,262,5,587
,3)
290 CALL SOUND(300,262,5,698
,3)
300 NEXT B
310 FOR C=1 TO 2
320 CALL SOUND(300,233,5,466
,3)
330 CALL SOUND(300,233,5,587
,3)
340 CALL SOUND(300,233,5,698
,3)
350 NEXT C
360 FOR D=1 TO 2
370 CALL SOUND(300,196,5,466
,5)
380 CALL SOUND(300,196,5,622
,5)
390 CALL SOUND(300,196,5,784
,5)
400 NEXT D
410 CALL SOUND(300,220,7,440
,7)
420 CALL SOUND(300,220,7,554
,7)
430 CALL SOUND(300,220,7,784
,7)
440 CALL SOUND(300,220,7,440
,7)
450 CALL SOUND(300,220,7,587
,7)
460 CALL SOUND(300,220,7,698
,7)
470 CALL SOUND(300,220,7,440
,7)
480 CALL SOUND(300,220,7,587
,7)
490 CALL SOUND(300,220,7,659
,7)
500 CALL SOUND(300,220,7,440
,7)
510 CALL SOUND(300,220,7,554
,7)
520 CALL SOUND(300,220,7,659
,7)
530 FOR G=1 TO 3
540 CALL SOUND(300,294,7)
550 CALL SOUND(300,349,7)
560 CALL SOUND(300,440,7)
570 NEXT G
580 CALL SOUND(300,440,1,880
,1)
590 CALL SOUND(200,440,1,880
,1)
600 FOR H=1 TO 3
610 CALL SOUND(300,277,5,880
,1)
620 CALL SOUND(300,392,5,880
,1)
630 CALL SOUND(300,440,5,880
,1)
640 NEXT H
650 CALL SOUND(300,440,1,880
,1)
660 CALL SOUND(200,440,1,800
,1)
670 CALL SOUND(300,880,1,294
,5)
680 CALL SOUND(300,800,1,349
,5)
690 CALL SOUND(300,880,1,440
,5)
700 CALL SOUND(300,880,1,294
,5)
710 CALL SOUND(300,880,1,349
,5)
720 CALL SOUND(300,880,1,294
,5)
730 FOR H=1 TO 2
740 CALL SOUND(300,932,1,196
,1)
750 CALL SOUND(300,932,1,233
,1)
760 CALL SOUND(300,932,1,294
,1)
770 NEXT H
780 FOR I=1 TO 2
790 CALL SOUND(300,880,1,262
,5)
800 CALL SOUND(300,800,1,349
,5)
810 CALL SOUND(300,880,1,440
,5)
820 NEXT I
830 CALL SOUND(300,784,1,262
,5)
840 CALL SOUND(300,784,1,330
,5)
850 CALL SOUND(300,784,1,466
,5)
860 CALL SOUND(300,1047,1,26
2,5)
870 CALL SOUND(300,1047,1,33
0,5)
880 CALL SOUND(300,1047,1,39
2,5)
890 FOR J=1 TO 3
900 CALL SOUND(300,698,1,349
,5)
910 CALL SOUND(300,698,1,440
,5)
920 CALL SOUND(300,698,1,523
,5)
930 NEXT J
940 CALL SOUND(300,1047,1,52
3,1)
950 CALL SOUND(200,1047,1,52
3,1)
960 FOR F=1 TO 3
970 CALL SOUND(300,1047,1,33
0,5)
980 CALL SOUND(300,1047,1,39
2,5)
990 CALL SOUND(300,1047,1,46
6,5)
1000 NEXT F
1010 CALL SOUND(300,1047,1,5
23,1)
1020 CALL SOUND(200,1047,1,5
23,1)
1030 CALL SOUND(300,1047,1,3
49,5)
1040 CALL SOUND(300,1047,1,4
40,5)
1050 CALL SOUND(300,1047,1,5
23,5)
1060 CALL SOUND(300,1047,1,3
49,5)
1070 CALL SOUND(300,1047,1,4
40,5)
1080 CALL SOUND(300,1047,1,3
49,5)
1090 CALL SOUND(300,1109,1,3
30,5)
1100 CALL SOUND(300,1109,1,3
92,5)
1110 CALL SOUND(300,1109,1,4
40,5)
1120 CALL SOUND(300,1175,1,2
94,3)
1130 CALL SOUND(300,1175,1,3
49,3)
1140 CALL SOUND(300,1175,1,4
40,3)
1150 FOR L=1 TO 2
1160 CALL SOUND(300,1319,1,1
335,3,227,1)
1170 CALL SOUND(300,1319,1,1
335,3,392,1)
1180 CALL SOUND(300,1319,1,1
335,3,440,1)
1190 NEXT L
1200 FOR M=1 TO 2
1210 CALL SOUND(300,1397,1,1
430,1,294,1)
1220 CALL SOUND(300,1397,1,1
430,1,349,1)
1230 CALL SOUND(300,1397,1,1
430,1,440,1)
1240 NEXT M
1250 CALL SOUND(300,1245,3,3
92,5)
1260 CALL SOUND(300,1245,3,4
66,5)
1270 CALL SOUND(300,1245,3,3
92,5)
1280 CALL SOUND(300,1245,3,4
66,5)
1290 CALL SOUND(300,1245,3,3
92,5)
1300 CALL SOUND(300,1245,3,4
66,5)
1310 FOR N=1 TO 2
1320 CALL SOUND(300,1109,3,4
40,5)
1330 CALL SOUND(300,1109,3,3
92,5)
1340 CALL SOUND(300,1109,3,3
30,5)
1350 NEXT N
1360 FOR O=1 TO 4
1370 CALL SOUND(300,1175,1,1
200,1,294,1)
1380 CALL SOUND(300,1175,1,1
200,1,370,1)
1390 CALL SOUND(300,1175,1,1
200,1,440,1)
1400 NEXT O
1410 CALL SOUND(300,1245,1,3
92,3)
1420 CALL SOUND(300,1245,1,3
92,3)
1430 CALL SOUND(300,1245,1,4
66,3)
1440 CALL SOUND(300,1109,1,3
92,3)
1450 FOR P=1 TO 4
1460 CALL SOUND(300,1175,1,2
94,3)
1470 CALL SOUND(300,1175,1,3
70,3)
1480 CALL SOUND(300,1175,1,4
40,3)
1490 NEXT P
1500 CALL SOUND(300,1245,1,3
92,3)
1510 CALL SOUND(300,1245,1,3
92,3)
1520 CALL SOUND(300,1245,1,4
66,3)
1530 CALL SOUND(300,1109,1,3
92,3)
1540 FOR Q=1 TO 3
1550 CALL SOUND(300,294,1,44
0,1,698,1)
1560 CALL SOUND(300,294,1,44
0,1,880,1)
1570 CALL SOUND(300,294,1,44
0,1,1175,1)
1580 NEXT Q
1590 CALL SOUND(300,440,1,88
0,1)
1600 CALL SOUND(200,440,1,88
0,1)
1610 FOR R=1 TO 3
1620 CALL SOUND(300,277,1,44
0,1,659,1)
1630 CALL SOUND(300,277,1,44
0,1,784,1)
1640 CALL SOUND(300,277,1,44
0,1,880,1)
1650 NEXT R
1660 CALL SOUND(300,440,1,88
0,1)
1670 CALL SOUND(200,440,1,88
0,1)
1680 FOR S=1 TO 3
1690 CALL SOUND(300,294,1,44
0,1,587,1)
1700 CALL SOUND(300,294,1,44
0,1,698,1)
1710 CALL SOUND(300,294,1,44
0,1,880,1)
1720 NEXT S
1730 CALL SOUND(300,440,1)
1740 CALL SOUND(200,440,1)
1750 FOR T=1 TO 3
1760 CALL SOUND(300,277,1,44
0,1,659,1)
1770 CALL SOUND(300,277,1,44
0,1,784,1)
1780 CALL SOUND(300,277,1,44
0,1,880,1)
1790 NEXT T
1800 CALL SOUND(300,440,1)
1810 CALL SOUND(200,440,1)
1820 FOR U=1 TO 2
1830 CALL SOUND(300,587,1)
1840 CALL SOUND(300,494,1)
1850 CALL SOUND(300,800,1)
1860 NEXT U
1870 FOR V=1 TO 2
1880 CALL SOUND(300,294,1)
1890 CALL SOUND(300,349,1)
1900 CALL SOUND(300,440,1)
1910 NEXT V
1920 FOR W=1 TO 2
1930 CALL SOUND(600,440,1,58
7,1,698,1)
1940 FOR I=1 TO 300
1950 NEXT I
1960 NEXT W
1970 CALL SOUND(2000,(50,1,1
75,1,590,1)
1980 END
```

Quickie

We suspose this could be called a "quickie", a "space filler" or some other such name, but call it any thing you like.

Type it in just for the fun of it and give your sound chip a little exercise.

```
100 CALL INIT
110 FOR C=1 TO 4
120 FOR Z=1 TO 400 STEP 8
130 CALL LOAD(-31744,Z*(1-C)
)
140 NEXT Z
150 NEXT C
160 CALL SOUND(1,1000,0)
170 GOTO 100
```

What is a Nibble?

This article originally appeared in the User Group of Orange County, California ROM

WHAT IS A NIBBLE, ANYWAY?

This month I am going to try and explain all of the various number words we run across. With luck, after you finish reading this, you will have some understanding of bit, byte, nibble, word, hex, binary and where -31952 really is in memory. With luck.

Computers really think in binary. In this numbering system there are two numbers, 0 and 1 (or, if you are a computer, off and on). While this works for your 4A, binary is cumbersome for humans. For example, in binary 41,576 is 1010001100011100.

Hex, or hexidecimal, has sixteen numbers from zero to F. Here are the first sixteen numbers in binary, decimal and hex:

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

The next number would be 16 or >10 or b1000 (> means hex and b means binary).

One binary number is a bit. Four bits is a nibble. So, 10 or A or 1010 takes four bits or a nibble to express.

A byte is eight bits or two nibbles. With a bit you can count from zero to one. A nibble gets you from zero to fifteen. The range of byte is:

Base	Low	High
Binary	0	11111111
Hex	0	FF
Decimal	0	255

You have probably noticed the numbers 16 and 255 when using your TI. ASCII character run from 0 to 255. There are sixteen colors (1 to 16, really 0 to 15). A string can be up to 255 characters long. And on and on.

Before tackling the next thing, a word, lets see if we can decode something. Lets take b10100 or >14. To convert either number to decimal, we need a method:

```
>14 is >10 plus >4
>10 is 16 and >4 is 4
 16 plus 4 is 20
 Hence, >14 is 20

b10100 is b10000 plus b100
b10000 is 16 and b100 is 4
16 plus 4 is 20
b10100 is 20
```

Further than that I cannot go in this space.

A word is sixteen bits or four nibbles or two bytes. The range of a word is:

Base	Low	High
Binary	0	1111111111111111
Hex	0	FFFF
Decimal	0	65,535

But there are no negative numbers. Since we need them, we use something called twos compliment (which is way beyond the scope of this column and this writer). I can tell you, however, the impact;

Hex range	Decimal Range
0-7FFF	0 to 32,767
8000-FFFF	-32,768 to -1

Remember that >8000 is the next number after >7FFF.
Some examples:

```
7FFF is  32,767
8000 is -32,768
FFFF is -1
0 is  0
```

Confused? So was I until I worked with it for a while. These conversion rules may help:

>>Any number less than or equal to 32,767 requires no conversion.

>>Subtract 65536 from any number over 32,767.

>>Add 65536 to any number less than zero.

This conversion process can be expressed in basic:

```
AD=AD+65536*(AD>32767)
```

If AD is the address, this returns the same number if AD is less than or equal to 32767. If AD is greater than 32767, the test returns true (-1) and a negative 65536 is added to AD. Try it on your computer.

Bottom line time. Suppose you see CALL PEEK(-31952,A,B). Where is -31952? Well, since it is less than zero, we add 65536 and get 33584 -or- >8330. Now you know!

Disk of the Month

Librarian Jim Thomas has assembled two "Disks of the Month", that will be available at the June meeting. Single disks are $3.00 or you can have both for $5.00.

DM-99 (Disk Manager 99), a fairware program written by Mike Dodd, is a resident disk manager for use with Extended Basic or Console Basic. It can unprotect files, protect files, rename files, initialize a disk and perform a host of other disk related utilities. It's easy to use and if you need to organize that 'pile of disks', this program is for you.

The second disk consists of five games, five selections of music and 2 utilities. The utilities are 'Superfont' and 'Sprite Definition'.

You can of course inspect the catalog of various programs available from the library and order anything you like. Your purchases from the club library help keep our dues low and also support other activities of the club.

Setup

SETUP is another of those fine programs found on the PLUS disks, written by Jack Sughrue. It is much like some of the other programs used to initiate certain features on your printer. It does however include some controls not found on previous programs such as NX (NX-1000 printer).

If you like this program you will like the other programs you'll find on the PLUS disks. They are in our library and if you use them you are encouraged to contribute a fee to Jack Sughrue, Box 459, E. Douglas, MA 01516.

After you run the program you will be presented with a menu to select from. Pick any selection or a combination of selections to suit your style. Your editor has this configured into his ram disk when he produces Word Play and has found it to be very helpful and time saving.

```
100 @=0 :: \=1 :: [=2 :: ]=3
 :: =4
110 OPEN #\:"PID" :: DISPLAY
 ERASE ALL :: CALL SCREEN(1
]
120 DISPLAY AT(6,6)BEEP ERAS
E ALL:"$$$$$$$$$$$$$$$$$$" ::
DISPLAY AT(8,6)[illegible]:"Printer
[illegible] " :: DISPLAY AT(10
,6[illegible]:" STAR PRINTERS " ::
 DISPLAY AT(12,6)BEEP:" & co
mpatibles"
130 DISPLAY AT(14,6)BEEP:" f
rom Data Mill" :: DISPLAY AT
(16,6)BEEP:" for [illegible] "
:: DISPLAY AT(18,6[illegible]:"$$$
(modified)$$$" :: FOR A=\ TO
 999 :: NEXT A :: B=\ :: GOT
O 210
140 DISPLAY AT(5, )BEEP ERAS
E ALL:A$ :: DISPLAY AT(7, ):
"1 ";B$ :: DISPLAY AT(9, ):
"2 ";C$ :: DISPLAY AT(11, )
:"3 ";D$ :: DISPLAY AT(13,
):"4 ";E$ :: DISPLAY AT(15,
 ):"5 ";F$ :: DISPLAY AT(17
, ):"6 ";G$ :: DISPLAY AT(1
9, ):"7 ";H$
150 DISPLAY AT(22, )BEEP:I$
:: DISPLAY AT(24,9):J$
160 ACCEPT AT(22,24)BEEP SIZ
E(8):C :: IF C>D THEN 160 EL
SE IF C=0 THEN 165 ELSE RETU
RN
165 RUN "DSK1.LOAD"
170 PRINT #\:CHR$(D):: RETUR
N
180 PRINT #\:CHR$(D)&CHR$(F)
:: RETURN
190 PRINT #\:CHR$(D)&CHR$(F)
&CHR$(G):: RETURN
200 PRINT #\:CHR$(D)&CHR$(F)
&CHR$(G)&CHR$(H):: RETURN
210 A$="Select Preference" :
: C$="PITCH of Font" :: B$="
STYLE of Font" :: D$="SPECIA
L CODES" :: F$="HORIZONTAL C
ontrols" :: E$="VERTICAL Con
trols"
220 E=\ :: G$="FORM Controls
" :: H$="TESTS & CONTROLS" :
: I$="CONFIGURATION" :: D=8
:: J$="PRESS 0 TO END" :: GO
[illegible] 140 :: IF C=0 THEN GOTO
[illegible] ELSE ON C GOSUB 240,300,
340,360,430,530,600
230 GOTO 210
240 A$="Select ASCII Charact
er STYLE Preference" :: B$="
Standard" :: C$="Italic" ::
D$="International" :: E$=""
:: F$="" :: G$="" :: H$=""
250 J$="" :: I$="CONFIGURATI
ON" :: D=3 :: GOSUB 140 :: O
N C GOTO 260,270,280
260 D=27 :: F=53 :: GOSUB 18
0 :: RETURN
270 D=27 :: F=52 :: GOSUB 18
0 :: RETURN
280 E=@ :: A$="0 AMERICAN"
:: B$="ENGLISH" :: C$="G[illegible]
N" :: D$="[illegible]" :: E$="[illegible]
[illegible]" :: F$="[illegible]H" :: G$="
[illegible]IAN" :: H$="SPA[illegible]" ::
D=7 :: J$="" :: I$="[illegible]P?"
:: GOSUB 140
290 D=27 :: F=55 :: G=C :: G
OSUB 190 :: RETURN
300 A$="Select Preference:"
:: B$="Pica (10 cpi)" :: C$
="Elite (12 cpi)" :: D$="Co
ndensed (17 cpi)" :: E$="" :
: F$="" :: G$="" :: H$="" ::
 J$="" :: I$="Font Pitch ?"
 :: D=3 :: [illegible] 140
310 IF C=@ [illegible] 210 :: D=27
:: F=66 :: G=C :: GOSUB 190
320 DISPLAY AT(7, )ERASE ALL
:"ENLARGED Mode ?? (YN)" ::
DISPLAY AT(9, ):"(will reduc
e cpi)" :: ACCEPT AT(11,11)V
ALIDATE("YN"):B$ :: IF B$="N
" THEN PRINT #\:CHR$(27)&CHR
$(87)&CHR$(@)ELSE PRINT #\:C
HR$(27)&CHR$(87)&CHR$(\)
330 RETURN
340 A$="Special PRINT Modes
:" :: B$="Emphasized (just p
ica)" :: C$="Quit Emphasized
" :: D$="Doublestrike" :: E$
="Quit Doublestrike"
350 J$="" :: F$="" :: G$=""
:: H$="" :: I$="Choose Mode
 " :: D=4 :: GOSUB 140 :: D
=27 :: F=C+68 :: GOSUB 180 :
: RETURN
360 A$="LINE FEED Controls"
:: B$="1/8 in./line" :: C$
="7/72 in./line" :: D$="1/6
 in./line" :: E$="x/72 in
./line" :: F$="x/144 in./lin
e"
370 J$="" :: G$="VERTICAL Ta
bulations" :: H$="" :: I$="S
et Line Feed :" :: D=6 :: GO
SUB 140 :: ON C GOTO 380,380
,380,390,400,410
380 D=27 :: F=C+47 :: GOSUB
180 :: RETURN
390 INPUT "Value of x=":I ::
 D=27 :: F=65 :: G=I :: GOSU
B 190 :: RETURN
400 INPUT "Value of x=":I ::
 D=27 :: F=51 :: G=I :: GOSU
B 190 :: RETURN
410 GOSUB 490
420 PRINT #\:CHR$(27)&CHR$(8
0);:: GOTO 520
430 A$="HORIZONTAL Controls"
 :: B$="Left Margin" :: C$="
Right Margin" :: D$="Set Tab
s" :: E$="" :: F$="" :: G$="
" :: H$="" :: J$="" :: I$="S
elect Set-up :" :: D=3 :: GO
SUB 140 :: ON C GOTO 450,460
,470 :: RETURN
440 INPUT "Column For "&K$&"
 Margin :":J :: IF J>255 THE
N 440 [illegible] IF J<\ THEN 440 E
LSE RE[illegible]
450 K$="Left" :: GOSUB 440 :
: D=27 :: F=77 :: G=J :: GOS
UB 190 :: RETURN
460 K$="Right" :: GOSUB 440
:: D=27 :: F=81 :: G=J :: GO
SUB 190 :: RETURN
470 GOSUB 490
480 PRINT #\:CHR$(27)&CHR$(6
8);:: GOTO 520
490 A$="TAB Directions:" ::
B$="1 =< # of Tabs <= 255" :
: C$="Please Set In Order."
:: D$="All Present Tabs"
500 B=] :: E$="Will Be Wiped
 Out." :: F$="" :: G$="" ::
H$="" :: B=3 :: I$="How Many
 Tabs ?" :: J$="" :: D=255 :
: GOSUB 140 :: K=INT(C):: DI
M L(255):: M=C :: FOR N=\ TO
 M :: I$="Tab Location # "&S
TR$(N):: J$="" :: GOSUB 150
510 L(N)=C :: NEXT N :: RETU
RN
520 FOR N=\ TO M :: PRINT #\
:CHR$(L(N));:: NEXT N :: PRI
NT #\:CHR$(@):: RETURN
530 A$="FORM F[illegible] Controls:"
 :: B$="PAGE LE[illegible]H (lines)"
 :: C$="[illegible] LE[illegible]H (inches)
" :: D$="[illegible] LINE [illegible]o
n" :: E$="[illegible]"
540 F$="!NO! SKIP F[illegible]" ::
G$="" :: H$="" :: I$="Select
CONFIGURATION" :: J$="" :: B
=] :: D=5 :: GOSUB 140 :: ON
 C GOTO 550,560,570,580,590
:: RETURN
550 I$="LENGTH (max 127 line
s) ?" :: D=127 :: B=] :: GOS
UB 150 :: D=27 :: F=67 :: G=
C :: GOSUB 190 :: RETURN
560 I$="LENGTH (max 32 in.)?
" :: D=32 :: B=[ :: GOSUB 15
0 :: D=27 :: F=67 :: G=@ ::
H=C :: GOSUB 200 :: RETURN
570 I$="HEADER LINE (max 16)
 ?" :: D=16 :: B=[ :: GOSUB
150 :: D=27 :: F=82 :: G=C :
: GOSUB 190 :: RETURN
580 I$="FOOT SPACE (max 127)
 ?" :: D=127 :: B=] :: [illegible]
 150 :: D=27 :: F=78 :: G=C
:: GOSUB 190 :: RETURN
590 D=27 :: F=79 :: GOSUB 18
0 :: RETURN
600 A$="CONTROLS & TESTS" ::
 B$="Re-Set to Top of Form"
:: D$="'Paper Out' ON" :: E$
="'Paper Out' OFF" :: C$="Re
-initialize Printer" :: F$="
Printing Tests"
610 B=\ :: D=5 :: G$="" :: H
$="" :: J$="" :: I$="Pick A
Number :" :: GOSUB 140 :: O
N C GOTO 670,630,640,650,660
620 PRINT #\:CHR$(12):: RETU
RN
630 PRINT #\:CHR$(27)&CHR$(6
4):: RETURN
640 PRINT #\:CHR$(27)&CHR$(5
7):: RETURN
650 PRINT #\:CHR$(27)&CHR$(5
6):: RETURN
660 A$="CHOOSE" :: B$="LIST
CODE" :: C$="WRITE TEXT" ::
D$="TEST CHARSET" :: E$="GO
NUTS!" :: F$="" :: G$="" ::
H$="" :: J$="" :: I$="Prefer
ence?" :: D=4 :: B=] :: GOSU
B 140 :: ON C GOTO 670,750,7
60,770
670 PRINT #\:"These characte
rs can be imbedded within a
document,":"when using most
word processors":"for desire
d results:"
680 PRINT #\: : :"(CTRL)N
= Enlarged Mode until CR":"(
CTRL)T = Quit Enlarged Mod
e":"(CTRL). 4 = Italics" ::
PRINT #\:"(CTRL). 5 = Quit I
talics"
690 PRINT #\:"(CTRL)R = Pi
ca":"(CTRL)O = Condensed":
"(CTRL). G = Start Doublestr
ike":"(CTRL). H = Stop Doubl
estrike"
700 PRINT #\:"(CTRL). E = St
art Emphasized (Pica ONLY)":
"(CTRL). F = Stop Emphasized
":"(CTRL). - (FCTN)7 = Start
 Underlining"
710 PRINT #\:"(CTRL). - (FCT
N), = Stop Underlining":"(CT
RL). S (FCTN), = Superscript
":"(CTRL). S (FCTN)7 = Subsc
ript"
720 PRINT #\:"(CTRL). T = St
op Super/Subscript":"(CTRL)J
 = Single-Line Feed":"(CTR
L)L = Form Feed (Next Top
of Form)"
730 PRINT #\:"(CTRL)K = Ne
xt Vertical Tab":"(CTRL)M
= Home Print-Head":"(CTRL)I
 = Next Horizontal Tab":"(C
TRL)H = Back [illegible]"
740 PRINT #\:"[illegible]G = BE
LL (beep)":"(CTRL)S = Prin
ter Off Line":"(CTRL)Q = P
rinter On Line" :: RETURN
750 DISPLAY AT(7, )ERASE ALL
:"START TYPING" :: DISPLAY A
T(9, ):"(a buffer full)" ::
INPUT L$ :: PRINT #\:L$ :: R
ETURN
760 FOR O=33 TO 126 :: PRINT
 #\:CHR$(O);:: N[illegible] O :: PRI
NT #\: : : :: R[illegible]N
770 RANDOMIZE :: FOR N=\ TO
1000 :: PRINT #\:CHR$(INT(32
+RND*94));:: NEXT N :: PRINT
 #\:CHR$(12):: RETURN
780 INPUT "Reset TOP OF PAGE
 ? (YN)":M$ :: IF M$="Y" THE
N D=12 :: GOSUB 170
790 END
```

Type - Like

Here is another short program that really doesn't do a whole lot, but type it in and listen to your computer sound like a typewritter when you type in any message to the screen.

We've included a sample message in lines 120, 140 and 160, but you change that to anything you like.

If anyone else has some of these little quick programs they would like to see published in Word Play, why not turn them over to the editor for printing in a future issue.

```
100 CALL CLEAR
110 CALL SCREEN(15)
120 S$="This is a test"
130 GOSUB 190
140 S$="of a typewriter simu
lation."
150 GOSUB 190
160 S$="123456789 123456789
123456789 12"
170 GOSUB 190
180 STOP
190 N=LEN(S$)
200 X=(32-N)/2
210 FOR L=1 TO N
220 C$=SEG$(S$,L,1)
230 C=ASC(C$)
240 IF C=32 THEN 260
250 CALL SOUND(1,-6,0)
260 CALL HCHAR(18,L+X,C)
270 NEXT L
280 PRINT : : :
290 RETURN
```

Loan Calculator

(NOTE FROM EDITOR: This program was submitted by our own Walt Morey. It is simple to use and easy to type in. Thanks Walt. WordPlay encourages all members to submit their programs also. Why not search through those disks for that favorite and send it in to the editor.)

Loan Calculator is a simple program written in X-BASIC to show how long a loan will take to be paid back. Three values are put in. The PRIMARY LOAN amount, the amount to be paid each month, and the Annual Interest or Percentage rate. With these amounts the program then calculates the BALANCE, MONTHLY INTEREST AMOUNT and the CUMULATIVE INTEREST amount.

Using IMAGE statements and opening a file to PIO ends up with a columnar printout with YEAR and MONTH indicated. An on-screen readout is BALANCE and INTEREST only. (This can be changed if you wish).

This program was written for a friend but was found useful for any amounts that were used for the variables. I am not an accountant but I think this program has possibilities. I'm sure it could be further modified by someone to do many more things than I could think of.

Written by Walt Morey in April of 1989. If you like it and can use it you are welcome to do what you want. My address is 2437 S.E. Taylor St., Portland, OR, 97214.

```
100 REM WRITTEN BY WALT MORE
Y
110 REM ON APRIL 12, 1989
120 CALL CLEAR
130 INPUT "PRIMARY LOAN ":PR
140 INPUT "MONTHLY PAY  ":PA
Y
150 INPUT "ANNUAL IN RT ":IA
160 PRINT
170 OPEN #1:"PIO"
180 IMAGE APR: ##.##%
190 IMAGE $#####.##    $###.
##
200 IMAGE $#####.##    #.####
   $###.##
210 IMAGE ## ###    $#####.#
#    $###.##    $#####.##
220 IR=(IA/100)/12
230 YR=1
240 PRINT #1,USING 180:IA
250 PRINT USING 200:PR,IR,PA
Y
260 PRINT
270 PRINT #1:"PRIMARY      I
NT RT  PAYBACK"
280 PRINT #1,USING 200:PR,IR
,PAY
290 PRINT #1
300 BAL=PR-PAY
310 IN=BAL*IR
320 INA=IN
330 CUM=CUM+INA
340 BAL=BAL+IN
350 PRINT USING 190:BAL,IN
360 MTH=MTH+1
370 IF MTH>12 THEN MTH=1 ELS
E GOTO 390
380 YR=YR+1
390 PRINT #1:"YR  MTH    BAL
ANCE     INTEREST     CUM-IN
T"
400 PRINT #1,USING 210:YR,MT
H,BAL,IN,CUM
410 BAL=BAL-PAY
420 IF BAL<=0 THEN END
430 GOTO 310
```

"God Save the Queen!"

A merchant in London hung a big sign outside his shop:

WE MAKE SAUSAGE FOR QUEEN ELIZABETH

Business boomed until a clever competitor across the street hung up his own sign:

GOD SAVE THE QUEEN

What a wonderful, seldom appreciated thing competition is. It forces us to stay alive and elert. In return it brings us better value for every dollar we spend.

What a pleasure it is, when you're looking for a new car, to have so many makes and models to choose from. When the coffee or beer you drink starts tasting flat, there are lots of other brands to try. If you think one dry cleaner charges too much, doesn't give prompt service, or has a surly, unpleasant attitude, you can always try another one down the street.

Unfortunately, for every winner there is also a loser. If too many people decide they don't like a certain kind of car, beer, or coffee, somebody is going to lose his job and have to look for another one. But what kind of country would this be if there were no penalties for poor work, faulty products and bad service? And no rewards for good work, excellent products and fine service?

In return for our freedom of choice--the right to judge the quality and value of other people's products and service--we have to give them the right to judge ours. That's fair enough, isn't it? As a result , everybody does better work; everybody benefits.

Does it occasionally iritate you that your employer keeps on insisting on better work, greater efficiency and better service to the customer? Actually, it isn't your employer who demands these things. It's your customer, a fellow just like you who wants to get the most for his money. And if you don't give it to him, he'll get it from somebody else.

Businesses continue to exist only because they do a good job for their customers. They deliver products and services good enough to attract customers and hold them. It's a battle every business--your company included--is fighting every day.

One of the most helpful things you can do as an employee is to understand this problem and try to make your company a winner. Your job depends on it.

Reprinted from the Economics Press.

Scrambler

The following is a short program that can be used to unscramble or decipher the Jumble Puzzles that appear in the daily newspapers. Some of these words, though short can be hard to figure out and with this little program the chore will be a lot easier.

It only takes a couple of minutes to type in, but can save hours of struggling to solve these kinds of puzzles.

```
10 CALL CLEAR
20 RANDOMIZE
30 INPUT "WORD TO SCRAMBLE "
:W$
40 FOR Y=1 TO 10
50 FOR I=1 TO LEN(W$)
60 R=INT(RND*LEN(W$)+1)
70 IF A(R)THEN 60
80 A(R)=1
90 X$=X$&SEG$(W$,R,1)
100 NEXT I
110 N$(Y)=X$ :: X$=""
120 PRINT N$(Y);" ";
130 FOR I=1 TO LEN(W$)
140 A(I)=0 :: NEXT I
150 NEXT Y
160 INPUT "REPEAT THAT WORD
Y/N ":ANS$
170 IF ANS$="Y" THEN 40
180 STOP
```

User Defined Functions

User defined functions in TI Basic and XBasic can be helpful when you sit down to write a program. You would generally use functions when you have a numeric or string expression that would be repeated many times throughout a program.

Re-read pages 72 and 73 of your XBasic manual to get a good understanding of defined functions.

The following examples will also help you to understand this important feature of your TI-99/4A computer

```
DEF ROUND(X)=INT(X+.5)
```

Then whenever you wish to round a value, you can use this function. For example: A=ROUND(B) will assign to A the value of B rounded to the nearest dollar or the nearest integer (it doesn't have to be just dollar amounts). If you wish to round to the nearest cent (two places after the decimal point(, change the function to:

```
DEF ROUND(X)=INT(100*X+.5)/100
```

One problem with functions is that they can only take one argument. It would be nice to write a function such as ROUND(X,D) which would round the value of X to D places, but this is not allowed in TI Basic. You can use other variables in your function definition, but they have to be assigned a value before calling the function — for example: rewrite the rounding function as:

```
DEF ROUND(X)=INT(50*D*X+.5)/100
```

To use this function, first set D to the number of decimal places that you want (for cents, you would use D=2). Then use the function as above.

Functions can be used for strings as well as numbers. If you will be looking at the first character of a string in a number of places in your program, you can define a function:

```
DEF FIRST$(X$)=SEG$(X$,1,1)
```

You may want to abbreviate the name as F$ if it will be used often. Now you may use statements such as IF FIRST$(S$)="Y" THEN. . . ., which will check to see if the first character of X$ is a Y.

You can pass a string variable to a numeric function, and visa-versa. You can also use one function within another. For example, if you already have defined the function FIRST(X$), you can define a logical function YES that will return true (-1) if the first character of a string is a Y as follows

```
DEF    YES(X$)=FIRST(X$)="Y"
```

Functions can save a lot of typing and memory, since long expressions can be reduced to a few characters. However, using functions is S-L-O-W especially in large programs or when functions call other functions. If you call a function in a time-critical part of your program (for example, inside a FOR loop), it may be better to write out the expression. One way to get some speed-up is to type in the function definitions last. It doesn't matter if they are at the beginning, middle or end of the program. When searching program memory for function definitions, Basic will look at the most recently entered line first, so it will find the definitions faster this way.

If you want to get an idea of how slow function usage is, type TRACE before running your program. This will show the line number of each line as it is executed. You may notice a significant pause on lines that call a function. Be sure to type UNTRACE to turn off the tracing.

—Steve Karasek

Kaleidoscope

Kaleidoscope is our fun program for the month and it is easy to type in. It has been written by Jim Peterson and we thank him for it.

When I was young, I can remember those round tubes that you pointed into a source of light and then looking through the tube you would turn the barrel and each turn would change the pattern of colors.

Jim Peterson has accomplished this for our computer. When you run the program you will see a continuing series of patterns similar to the old kaleidoscope of years gone by. If you see an interesting pattern, you can keep it on the screen by holding down any key. When you release the key the patterns will continue to develope.

This little gem would be a good one to type in for the kids and let them have a little fun. We continue to provide all programs to the library for all who do not have the time to type in programs listed in WordPlay.

```
100 CALL CLEAR :: DISPLAY AT
(2,5):"POCKET KALEIDOSCOPE"
:: DISPLAY AT(20,3):"Progra
m made by Jim Peterson"
110 DISPLAY AT(15,3):"Hold d
own any key to freeze." :: F
OR D=1 TO 800 :: NEXT D
120 DIM L$(12):: RANDOMIZE :
: M$="0018243C[illegible]67E8199A5
8DC3DBE7FF" :: [illegible]
130 FOR CH=40 TO [illegible] B
:: FOR L=1 TO 4 :: X$=SEG$(M
$,INT(16*RND+1)*2-1,2):: B$=
B$&X$ :: C$=X$&C$ :: NEXT L
:: CALL CHAR(CH,B$&C$):: B$,
C$=[illegible]$ :: NEXT CH
135 CALL CLEAR
140 FOR L=1 TO 12 :: FOR L2=
1 TO 12 :: X$=CHR$(INT(13*TN
D+5)*8):: L1$=L1$&X$ :: L2$=
X$&L2$ :: NEXT L2 :: L$(L)=L
1$&L2$ :: PRINT TAB(3);L$(L)
:: L1$,L2$=NUL$ :: NEXT L
150 FOR P=12 TO 2 STEP -1 ::
 PRINT TAB(3);L$(P):: NEXT P
:: PRINT TAB(3);L$(1);
160 GOSUB 250
170 Z=INT(7*RND+1):: ON Z GO
SUB 180,220,190,220,200,220,
210 :: GOTO 160
180 FOR C=2 TO 14 :: CALL CO
LOR(C,1,1):: GOSUB 250 :: NE
XT C :: RETURN
190 CALL SCREEN(INT(15*RND+2
)):: RETURN
200 X=INT(15*RND+2):: FOR C=
2 TO 14 :: CALL COLOR(C,X,X)
:: NEXT C :: GOSUB 250 :: RE
TURN
210 FOR C=2 TO 14 :: X=INT(1
5*RND+2):: CALL COLOR(C,X,X)
:: GOSUB 250 :: NEXT C :: RE
TURN
220 FOR C=2 TO 14 :: X=INT(1
3*RND+2)
230 Y=INT(13*RND+2):: IF Y=X
 THEN 230
240 CALL COLOR(C,X,Y):: GOSU
B 250 :: NEXT C :: RETURN
250 CALL KEY(0,K,ST):: IF ST
<>0 THEN 250 ELSE RETURN
```

WORDPLAY
The PUNN Newsletter
P. O. Box 15037
Portland, Oregon 97215

DISCLAIMERS: The PUNN User's Group is not affiliated with or sponsored by TI and has no relationship with them, implied or otherwise.
Mention of a company or product is not an endorsement of that company or product.
We are not a subsidiary or branch of any other User's Group and any relationship we may have with other groups is on the basis of equals.

All general meetings are held on the first Tuesday of each Month, at the PGE Building 1700 S.E. 17th. Avenue Portland, Oregon

-Next Meeting Date-
June 6th. - 7:00p.m.

The PUNN Newsletter
WORDPLAY

P. O. Box 15037 Portland, Oregon 97215

June 1989-Volumn 8-No. 6