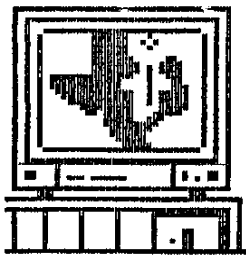

QB-MONITOR

QB-99'ERS U.G. NEWSLETTER

T.I.C.O.F.F. March 1989 EDITION

The QB MONITOR is the Newsletter of the QB-99'ers User Group, is printed Sept. thru June and sent in exchange for other User Group Newsletters. Send Exchange Newsletter to Frank Cotty, Queensborough Community College, Bayside, NY 11364. Credit original sources.

The QB 99'ers meets the second Saturday of each month September through May, at Queensborough Community College, Bayside New York, room S225, at 2 P.M. Calendar at right shows dates

February 1989						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	(11)
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

March 1989						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	(11)
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

April 1989						
S	M	T	W	T	F	S
						1
2	3	4	5	6	7	(8)
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

T.I.C.O.F.F. MARCH 18, 1898

This month is T.I.C.O.F.F. at Roselle Park High School. The QB-99'ers are fully supporting this event. Come see our representatives at the User Group's Table. Talk to our members, pick up a free issue of the MONITOR newsletter, buy a diskette full of programs written by our members. But, most of all enjoy the company of others with the same interest as you!

To formally acknowledge T.I.C.O.F.F. 1989 this month's MONITOR contains only TInygrams. Written by Ed Machonis and by Jack Youngs these programs have appeared in earlier editions of the MONITOR. They will make your T.I.-99/4A hum. The instructions/programs found in this issue complement the programs found on the TInygram disk being sold by our group at T.I.C.O.F.F. 1989.

Contents **Page**

FORTUNE OF WHEELS

A Tinygram

by Ed Machonis

Last month I promised that I would have something this month for the people without printers. In lieu of Epsons and Axioms I give you a Fortune of Wheels. Just to prove that I CAN write a program that doesn't use a printer.

I must admit I had a little help. The Tinygram presented here is an enhanced version of son Michael's WORDGUESS which is on the TIMARC disk. Originally a TI Basic 10 Liner, it has been recast in Extended Basic and the hidden phrase display resembles that used in a popular TV show of similar name. Sorry, no Vanna White to turn over the letters. What do you expect from a TINYgram?

Unlike the TV show, where the amount of the prize depends on the random spin of a wheel, the prize in this game is proportionate to the relative difficulty of the puzzle and how quickly you solve it. The longer the phrase, the greater the prize; the fewer tries, the greater the prize.

Fortune of Wheels is a two player or two team game. The first player or team leaves the room or turn their backs to the screen while the second player or team enters the mystery phrase. As soon as ENTER is pressed the screen will clear. Alternatively, if you are sure of your typing, the TV brightness or contrast can be turned down, or the program revised to black out the screen during entry.

If you wish to black out the screen during entry of the mystery phrase, change Line 2 to read as follows:

```
2 CALL SCREEN(2):: INPUT M$
:: CALL CLEAR :: CALL SCREEN
(8):: L=LEN(M$)
```

The first player or team can now try to guess the individual letters or the entire phrase. You must enter the entire phrase to be recognized as a winner. If you do not enter the entire phrase, do not enter more than one character. Entering a wrong letter, or more than 1 letter, or an incorrect phrase will cost you a try and reduce your prize.

Cumulative totals can be kept on paper. (Horrors! Let it be a challenge to you. Either sharpen your pencils or your programming skills!) Negative amounts won (Possible!), should be subtracted from the cumulative totals.

This Tinygram is easy to type in, quick to load (cassette users take note), and FUN to play. It can be as simple or challenging as you and your opponents care to make it.

Minimum requirements are Console, Cassette Player, TV and Extended Basic.

```
1 ! *** FORTUNE OF WHEELS **
    *      A TINYGRAM      *
    * by Mike & Ed Machonis*
    ************************

2 CALL CLEAR :: INPUT "ENTER
 THE MYSTERY PHRASE    ":M$
:: CALL CLEAR :: L=LEN(M$)

3 D$=RPT$(CHR$(30),L):: FOR
J=1 TO L :: IF SEG$(M$,J,1)<
>" " THEN 4 ELSE D$=SEG$(D$,
1,J-1)&" "&SEG$(D$,J+1,L)

4 NEXT J :: PRINT D$

5 T=T+1 :: PRINT :"TRY No.";
T;; :: INPUT "TYPE LETTER O
R ENTIRE PHRASE":A$ :: IF LE
N(A$)>1 AND LEN(A$)<L THEN 5

6 W=L+1-T :: IF A$=M$ THEN 9

7 FOR J=1 TO L :: IF SEG$(M$
,J,1)=A$ THEN D$=SEG$(D$,1,J
-1)&A$&SEG$(D$,J+1,L)ELSE 8

8 NEXT J :: PRINT :D$ :: GOT
O 5

9 FOR J=1 TO W :: CALL SOUND
(200+J*10,330+40*J,0):: NEXT
 J :: PRINT :"YOU WIN ";STR$
(W);",000 WHEELS!":;: :: INP
UT "PRESS ENTER TO PLAY AGAI
N":G$ :: T=0 :: GOTO 2
```

The 2nd word in the 2nd line originally read "listed". Using STYLE A LINE, the printed article was corrected to read "typed".

STYLE A LINE
A TINYGRAM
by Ed Machonis

TINYGRAM: A short program which can be typed in its entirety on one screen without any program lines scrolling off the screen. (REM statements can scroll off.) Popularized, I believe, by Mike Stanfill of the Dallas TI Home Computer Group.

First of all let me make clear that this is not a novelty program. It is a work horse, provided you have the work for it. What kind of work? Do you ever have to print just a line or two, such as a page header, an article or picture title, a title for a data base printout, a credit line for a reprinted newsletter article, etc., etc. Further, would you like to print this in an Expanded Compressed Italicized Double Strike Underlined type style? Yes all the same time! If so, this program is for you.

What no printer? I will try to have something for you next month. (A TINY GRAM - NOT a printer!)

Many of you are familiar with my 10 Line basic programs, PRINTSTYLE and and PRINTALINE. (Both TINYGRAMS, written before I knew the name existed.) I often use both of them in titling data base printouts or copy for the Newsletter but it got to be a pain to change between the two every time I wanted to change a type style. Finally the light dawned! Why not marry the two?

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

Using LINPUT required that the program run in extended basic. After some streamlining by deletion of unneeded features from PRINTALINE and the consolidation of statements into multi-statement lines, we wound up with 9 Lines of code. (After merging TWO TEN Line programs. The power of extended basic!)

Don't let its brevity fool you. You can select any of the 128 type styles available on the Epson RX-80 and many compatibles. With line spacing and margin variations, over 2000 different selections can be had. (Half line spacing and condensed superscript will let you tack on several lines of comment onto a photocopied article.)

Although there are better ways of doing it, you can even produce a right margin justified letter. (THIS is novelty!) Using Emphasized Pica, set Left Margin at 13, and enter text. Two screen lines will print text 54 characters wide (LINPUT uses two character spaces.) Justify text by inserting spaces between words so that second line ends at screen edge. But it will NEVER replace TI-Writer!

Using the program is very easy. When RUN, a menu is displayed for programming the printer. It is always best to select "1" to clear the printer. If your printer doesn't support a master reset code, turn it off then on to clear it. Combine styles by successive selections. Select Option 10 to input text.

If you wish to change the type style, or do repeated printings of the same text, typing "ZZZ" or "zzz" will return you to the menu. Option 9 will do repeat printing of the same text and styles can be changed as required. To input new text, select Option 10 again. When in text mode, pressing ENTER with no text input will print a blank line.

Watch those commas in Line 10. The next to last data item is a lower case "L", not the figure 1.

BRAIN TEASER: Where is the data to set the left margin at column 13?

```
1 ! *** STYLE A LINE   ***
  a TINYGRAM by Ed Machonis
    QB-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 6P6 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 :"INPUT 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,4,6,-1,,,6
0,S1,1,1,QC
```

TIny LOTTO
A TIny GRAM
by Ed Machonis

The first program I ever wrote on my newly purchased TI-99/4A was a random number generator for NY State Lotto games. When I finally got the program to generate 6 numbers between 1 and 40, I was elated. Not having any way to save the program, I copied the code onto paper with a pencil.

Knowing I was on my way to my first million, I decided to splurge on a cassette recorder. Leaving the computer turned on, I embarked on a two hour trip into the city to purchase a recorder. Upon my return, about five hours later, I saved the program. That first number generator eventually grew to 46 sectors, with nearly every conceivable bell and whistle, and took over 3 minutes to load from cassette.

Here I am, 5 years later, still writing Lotto programs and still chasing that first million. Hope also springs eternal in the compulsive programmer's [Gambler's?] breast! If nothing else, I did learn to write them smaller.

But don't let the small size fool you. This screenfull of code does a lot of work and once again proves the power of the TI-99/4A. It will generate random numbers for any of the popular lottery games, WIN 3, WIN 4, Pick 6 LOTTO, and WIN 10 or Keno. The low number can be a zero or a one. The high number can be whatever is being used, 40, 48, 54, 80, 999 or 9999. It should work in any state.

The same RND statement, in Line 4, borrowed from son Michael's Basic 10 liner LUCKY LOTTO, is used to generate the random numbers for all games. A clever piece of code well worth your study.

Where multiple numbers are generated for a game, as in Lotto or Keno, duplicate numbers are discarded and the numbers are sorted in ascending order to make it easier to fill out your bet slip. Output can be directed to screen or printer. When several games are played, the hard copy is easier to check for winners than the individual tickets.

Leading zeroes are inserted where required to keep the columns neatly aligned and to reduce the possibility of transcription errors. A total of 10 Lotto games (the bet slip capacity) can be displayed on the screen without any scrolling off.

The N.Y. State Lottery states "...if you are playing Lotto for the big prize, pick your numbers randomly." Early on, I distrusted the randomness of TI's RND function, and in my naivety visualized having to split that first million with half the TI owners in NY State. Despite a RANDOMIZE statement, the computer often generated identical series of numbers.

TI's User's Reference Guide states on page II-96, "The random number function gives you the NEXT PSEUDO-RANDOM number in the current SEQUENCE of pseudo-random numbers." Page II-95 states: "When the RANDOMIZE statement is used ..... a different and unpredictable SEQUENCE of random numbers is generated ..... each time the program is run." RND generates numbers in accordance with a built-in sequence. The RANDOMIZE statement merely insures that a program does not ALWAYS start with the same sequence. But it can, HAS and will.

The RANDOMIZE statement in Line 3 can be placed in three different positions. Placing it before the start of the G loop will cause an unpredictable sequence to be selected each time the program is RUN. Placed before the start of the K loop, a new sequence is used for each game. Placing it after the start of the K loop, as it is, causes an unpredictable sequence to be selected for each number that is generated. As only one number is used from each sequence, we are no longer governed by the built-in sequence and the program generates truly random numbers.

WIN 3 numbers can be selected with TIny LOTTO in one of two ways. We can use a Low Number of 0, a High Number of 999, and 1 number per game. Or one can use a Low Number of 0, a High Number of 9, and three numbers per game. The same two methods are available for four digit numbers, using 9999 and 1, or 9 and 4, as required. In the first case, a three digit number is selected, in the second case each digit of the three digit number is separately selected. Just a little user friendliness to conform to the way the user thinks of the numbers.

If you find you only play one type of game, and are always entering the same information in response to the prompts, Line 2 can be changed to permanently assign values to the variables. Suppose your regular selection is for 10 games of Pick 6 Lotto, with a low number of 1 and a high number of 54, with output to a printer. Line 2 would read:

```
2 L=1 :: H=54 :: T=6 :: Q=10
 :: P=1 :: OPEN #P:"PIO"
```

Due to sale of public domain programs by some software distributors, a copyright notice has been placed on this program. It may be freely distributed provided no fee of any kind is charged. This article and/or the program listing may be published in Newsletters of non profit user groups.

```
1 !$$$$    TIny LOTTO    $$$$
  $      Copyright 1988     $
  $      by Ed Machonis     $
  $ QB-99'ers, Bayside NY $
  $$$$$$$$$$$$$$$$$$$$$$$$$$

2 CALL CLEAR :: INPUT "LOW N
UMBER? ":L :: INPUT "HIGH NU
MBER? ":H :: INPUT "NUMBERS
PER GAME? ":T :: INPUT "HOW
MANY GAMES? ":Q :: INPUT "SC
REEN=0 - PRINTER=1 (0/1)?":P
 :: IF P THEN OPEN #P:"PIO"

3 H$=STR$(H):: FOR G=1 TO Q
:: FOR K=1 TO T :: RANDOMIZE

4 N(K)=INT(RND*(H+ABS(L=0)))
+L :: FOR D=1 TO K-1 :: IF N
(K)=N(D)AND H>9 THEN 4

5 NEXT D :: NEXT K :: U=T-1
:: IF H<10 THEN 9

6 F=0 :: FOR K=1 TO U :: IF
N(K)<N(K+1)THEN 8

7 M=N(K):: N(K)=N(K+1):: N(K
+1)=M :: F=1 :: U=K

8 NEXT K :: IF F=1 THEN 6

9 FOR K=1 TO T :: N$(K)=STR$
(N(K)):: PRINT #P:RPT$("0",L
EN(H$)-LEN(N$(K)))&N$(K)&" "
;:: NEXT K :: PRINT #P:;;;;;
:: NEXT G ! GOOD LUCK!
```

NAME THAT PHONE

A TIny Gram.....by Ed Machonis

(Sorry Regina, I just couldn't resist that title!)

Do you remember calling TI CARES? Never had to look up that number, it was always at the tip of your fingertips! And what an asset it was to TI. I don't believe that Helpline cost TI a penny. It paid its weight in free advertising. Knowing that help was at the other end of a toll free line sold many a computer.

Today, many businesses strive for a telephone number that can be easily remembered. Perhaps there is a word or phrase hidden in your own phone number? But how to discover it? You could look over the dial and see what letters are involved and try arranging them into words. Good Luck!

Each of the seven digits in your phone number, excluding ones and zeros, can represent any one of three letters of the alphabet. The number of possible combinations is 3 to the 7th power, or 2187. Try arranging them into words some rainy weekend.

Sounds like one of the tasks we bought our computers for and it is. The TIny Gram described herein can do the job in just over 11 minutes. It will present you with every combination of letters existing in your number. It will display them on screen or send them to your printer. The printout, in 6 columns takes up about six pages. Not bad for a screenfull of code.

When RUN, the program will prompt you to enter your phone number, one digit at a time. It will then ask you to choose Screen or Printer. (You may enter any number from 1 to 255 for the Printer.)

There are no letters assigned to One or Zero. If you enter these numbers into the program, it will display asterisks for zeroes and number signs for ones.

The screen display is in two columns and scrolls by just about fast enough for you to follow. Should you spot some interesting combinations you would like a closer look at just break the program with FUNCTION 4. When you are ready to continue, just tell the computer so by typing CON and it will resume where it left off.

If you don't find a word or phrase, you should certainly be able to find a phrase, the initial letters of which would represent your number. Try to work out a phrase which is easily associated with you.

In my own case, the best I could find was EGDRAMA which I turned into the mnemonic "Ed's Great DRAMA." Using initial letters, I can also be reached with "Ed Is Forever Programming A Marvelous Computer." Avoid using numbers in your mnemonics, like "Ed Has Damaged Seven Brand New Cars." Your friends start to wonder, "Let's see, is it SIX or SEVEN new cars that maniac has destroyed?" And then there are the characters who will persist in dialing the number 7 instead of the initial letter "S".

There may be a fantastic mnemonic hiding in your phone number, but you won't know it unless you run this program. Again, GOOD LUCK!

```
1 !****NAME THAT PHONE*****
   * No 1's or 0's Please *
   *     A TIny Gram      *
   *    by Ed Machonis    *
   **QB-99'ers Bayside NY**

2 DEF S$=SEG$(G$,1,LEN(G$)-1
):: A$="******ABCDEFGHIJKLMN
OPRSTUVWXY" :: FOR C=1 TO 7

3 INPUT "ENTER DIGIT "&STR$(
C)&" OF PHONE # ":D :: E$=E$
&SEG$(A$,D*3+1,3):: NEXT C

4 PRINT :"0=SCREEN":"1=PRINT
ER":"CHOICE (0/1)",:: INPUT
P :: IF P THEN OPEN #P:"PIO"

5 FOR F=1 TO 3 :: G$=""&SEG$
(E$,F,1):: FOR H=1 TO 3 :: G
$=G$&SEG$(E$,3+H,1):: FOR J=
1 TO 3 :: G$=G$&SEG$(E$,6+J,
1):: FOR K=1 TO 3 :: G$=G$&S
EG$(E$,9+K,1):: FOR L=1 TO 3

6 G$=G$&SEG$(E$,12+L,1):: FO
R M=1 TO 3 :: G$=G$&SEG$(E$,
15+M,1):: FOR N=1 TO 3

7 G$=G$&SEG$(E$,18+N,1):: PR
INT #P:G$,:: G$=S$ :: NEXT N
 :: G$=S$ :: NEXT M :: G$=S$
 :: NEXT L :: G$=S$ :: NEXT
K :: G$=S$ :: NEXT J :: G$=S
$ :: NEXT H :: NEXT F
```

LA STYLER was writtn by Tom Freeman of LA 99 er's. I don't believe I saw the original but I did see and use the revision by Ed Machonis of QB 99er's.

I have a JX-80 color printer, and I thought this would be a very good program for me if I could add the color codes to the program. I did it before for Ed's label and tag programs and others. I also have a merge program to use when needed. Anyway the program is as follows for anyone out there that has a JX-80 color printer.

This is the screen you see when the program loads

```
MODE            1=ON,0=OFF
1  ELITE           0
2  COMPRESSED      0
3  EMPHASIZED      0
4  DOUBLE STRIKE   0
5  EXPANDED        0
6  ITALICS         0
7  UNDERLINE       0
8  SUPERSCRIPT     0
9  SUBSCRIPT       0
10 X/72 IN. LF   X=12
11 L MARGIN      X=0
12 R MARGIN      X=80
13 SKIP X LINES  X=0
14 COLOR FOR EPSON JX80
15 TEST
16 RESET/PICA
17 INPUT TEXT
18 EXIT
```

Some of the control codes of the JX80 are different so they were changed also. for instance, there is no Elite with Emphasized and no Compressed with Emphasized and a few others.

Most important in typing in this program, the 3rd and 4th DATA items in line 360 are not blank spaces but CHR$(18) (Type CONTROL plus R) and CHR$(15) (CONTROL O),respectively. Similarly ,the apparent two blank spaces at the end of EACH quoted string in line 370 are actually CHR$(10) (CONTROL J) and CHR$(13) (CONTROL M). The blank space at the beginning of EACH quoted string is a true blank space and required in this program (as it saves sending additional Escape code to the printer).

```
100 ! *** JX80 STYLER ***
by Jack Youngs QB 99ers-a
revision of program by Ed
Machonis QB 99ers-based on a
 program by Tom Freeman

110 DIM P$(16,2),N(18):: M(1
)=12 :: M(2),M(4)=0 :: M(3)=
80

120 FOR X=1 TO 16 :: FOR J=0
 TO 1 :: READ P$(X,J):: NEXT
 J :: NEXT X :: FOR X=1 TO 4
 :: READ T$(X):: NEXT X :: O
PEN #1:"PIO.CR" :: PRINT #1:
CHR$(27)&"@"

130 DISPLAY AT(3,1)ERASE ALL
:"MODE","1=ON,0=OFF","1  ELI
TE":"2  COMPRESSED":"3  EMPH
ASIZED":"4  DOUBLE STRIKE":"
5  EXPANDED":"6  ITALICS":"7
  UNDERLINE"

140 DISPLAY AT(11,1):"8  SUP
ERSCRIPT":"9  SUBSCRIPT"

150 IF I=8 THEN N(9)=0 ELSE
IF I=9 THEN N(8)=0

160 IF N(1)=1 THEN N(3)=0 EL
SE IF N(3)=1 THEN N(1)=0

170 IF N(2)=1 THEN N(3)=0 EL
SE IF N(3)=1 THEN N(2)=0

180 FOR X=1 TO 9 :: DISPLAY
AT(X+3,18):N(X):: NEXT X

190 FOR X=1 TO 4 :: DISPLAY
AT(X+12,1):T$(X);STR$(M(X)):
: NEXT X

200 DISPLAY AT(17,1)SIZE(23)
:"14 COLOR FOR EPSON JX80"

210 DISPLAY AT(18,1):"15 TES
T":"16 RESET/PICA":"17 INPUT
 TEXT":"18 EXIT"

220 ACCEPT AT(22,1)VALIDATE(
DIGIT," ")SIZE(-2)BEEP:I

230 IF I>18 THEN 220 ELSE N(
I)=N(I)XOR 1 :: ON I GOTO 33
0,330,330,330,330,330,330,33
0,330,310,310,310,310,240,33
0,300,340,350

240 CALL CLEAR :: DISPLAY AT
(3,2):"0 BLACK  4 YELLOW
      1 RED    5 ORANGE
      2 BLUE   6 GREEN
      3 VIOLET 7 STYLER"

250 INPUT "COLOR OR STYLER?"
:I

260 IF (I<0)+(I>7)THEN 240

270 PRINT #1:CHR$(27);"r";CH
R$(I);

280 IF (I<>7)THEN 240

290 IF I=7 THEN 130

300 FOR X=1 TO 14 :: N(X)=0
:: NEXT X :: GOTO 320

310 ACCEPT AT(I+3,19)VALIDAT
E(DIGIT," ")SIZE(-2)BEEP:M(I
-9)

320 PRINT #1:CHR$(27)&P$(I,N
(I))&CHR$(M(I-9)):: GOTO 150

330 PRINT #1:CHR$(27)&P$(I,N
(I)):: IF I=16 THEN M(1)=12
:: M(2),M(4)=0 :: M(3)=8 0 :
: GOTO 150 :: ELSE 150

340 PRINT "INPUT A LINE OF T
EXT":"(ZZZ RETURNS TO MENU)"
 :: LINPUT A$ :: IF A$="ZZZ"
 OR A$="zzz" THEN 130 ELSE P
RINT #1:A$&CHR$(10)&CHR$(13)
:: GOTO 340

350 CLOSE #1

360 DATA P,M, , ,F,E,H,8,WO,
W1,5,4,-0,-1,T,S0,T,S1,A,A,1
,1,0,Q,N,N,x0,x1

370 DATA " QUICK BROWN FOX J
UMPS OVER THE LAZY RED DOG 1
234567890  "," QUICK BROWN F
OX JUMPS OVER THE LAZY RED D
OG 1234567990  ",0,0

380 DATA 10 X/72 IN. LF  X=,
11 L MARGIN     X=,12 R MARG
IN     X=,13 SKIP X LINES X=

390 DATA BLACK,RED BLUE,VIOL
ET,YELLOW,ORANGE,GREEN
```

FLEXI LABEL

A TIny Gram

By Ed Machonis

They say the only way to finish a program is to shoot the programmer. I guess the same could be said for the steady stream of label printing programs which seem to come out of this TI-99. I thought I had written all the label printing programs I would ever need, but I seem to keep discovering new needs.

In the past, most of my video tape labeling has consisted of pencil entries on the slip case. A recent visit by my grandchildren resulted in a stack of unmarked video cassettes piled alongside a stack of empty slip cases. They seem to have devised a new game called video roulette.

The only way to restore any semblance of order was to skim through each tape to identify the contents and match it with its slip case. Determined not to repeat this chore after subsequent visits, I decided to do what should have been done in the first place. Label the cassette as well as the case. 20/20 Hindsight!

Mailing labels are an exact fit on the side of the video cassette. Often 6 lines of text are needed for a 6 hour tape with six different programs. Rather than use an existing program, such as Disk Label, I decided to write a more flexible program which could handle Video Cassette labels as well as other types of labels.

Instead of one new program, I wound up with two, each a TIny Gram. FLEXI-LABEL's distinguishing feature is providing the user with the option to print up to 10 lines of text per label. Great for those video cassettes chock full of programs.

When first booted, you are asked to input the number of lines of text to be printed on the label. The font used is expanded compressed which enables an easily readable 28 character line. For labels with more than 7 lines, the font automatically changes to superscript. You are prompted to input the text for each line.

At any time during text entry you can change the number of label lines by entering FUNCTION C (Accent Grave); think FUNCTION C(hange). It can be entered anywhere in a line of text or by itself. The lines you have entered will not be lost, they always default to the next label. FUNCTION 3 can be used to erase unwanted lines. All editing keys are functional. If you want to redo a label, just enter zero for the quantity to be printed.

Text entry is automatically limited to 28 characters. Text can be carried over from label to label without re-entry, handy for those labels requiring only minor changes. Any line can be indented by entering spaces at the beginning of the line. I think you'll find the program as user friendly as they get.

This TIny Gram should answer most of your labeling requirements, whether they be video cassettes, return address, meeting notices, publicizing your User Group, "Property Of" labels, or simple mailing labels. Its small size makes it a candidate for your Funnelweb utility disk.


```
1 ! ***** FLEXI LABEL ******
    *      A TIny Gram      *
    *    Copyright 1988     *
    *    By Ed Machonis     *
    **QB-99ers, Bayside NY**

2 OPEN #1:"PIO.LF"

3 DISPLAY AT(8,1)ERASE ALL:"
LINES OF TEXT/LABEL?(MAX 10)
" :: ACCEPT AT(9,26)VALIDATE
(DIGIT):S :: IF S>10 THEN 3

4 E$=CHR$(27):: PRINT #1:E$&
"@"&E$&"G"&E$&"W1"&CHR$(15)&
E$&"C"&CHR$(0)&CHR$(1)&E$&"3
"&CHR$(216/(S+1)):: IF S>7 T
HEN PRINT #1:E$&"S0"

5 DISPLAY AT(1,1)ERASE ALL:"
ENTER ""`"" TO CHANGE #/LINE
S" :: FOR J=1 TO S :: DISPLA
Y AT(J*2,3):"ENTER LINE";J;L
$(J):: ACCEPT AT(J*2+1,1)SIZ
E(-28):L$(J):: IF POS(L$(J),
"`",1)THEN 3

6 NEXT J :: DISPLAY AT(23,1)
:"HOW MANY LABELS?" :: ACCEP
T AT(23,18):Q :: FOR K=1 TO
Q :: FOR L=1 TO S :: PRINT #
1:" ";L$(L);CHR$(10):: NEXT
L :: PRINT #1:CHR$(12):: NEX
T K :: GOTO 5
```

Coded for Epson RX-80

To Be Continued. (YOU HAVE BEEN WARNED!)

DISK LABEL II

Print Utility BU

A TINYGRAM From The

Library of Ed Machonis

The original DISK LABEL was a 10 Line Basic program and is on the TIMARC 4/86 disk in our library. It was written to solve the problem of the missing disk labels which were not included with packages of bulk diskettes.

I have been using mailing labels as disk labels for over two years without any problems. They are the preferred label for my disks; the "store boughten" kind are only used as temporary labels until a permanent one can be printed. I find it much easier to locate a disk in a storage case when the name is printed with an expanded type style. Colored ribbons add a nice touch.

The label used as a title for this article is an example of the labels generated by the program. The disk name appears on the first line in expanded emphasized underlined double strike type and is limited to 17 characters. The second line is available for those disks with longer titles or where two titles are appropriate (great for flippies); the same type style is used. Centering of titles is done by the program. If not required, the second line is left blank to enhance appearance and locatability.

The 3rd, 4th and last lines are limited to 28 characters, are printed expanded compressed double strike and, except for the last line, are underlined. The last line is also italicized.

The 3rd line is used for describing the disk contents, such as GAMES, UTILITIES, MP DATA, etc. The end of the line can be used to identify back ups or disk format such as BU, DSDD, etc.

The 4th line is for remarks and can be used for language, loading info, program names, etc. When required, the 3rd and even the 5th line can also be used for remarks.

The last line is used to identify the owner; handy for those round robin copy sessions, ensuring you go home at least with the disks you arrived with. Centering is automatic. It is also useful for identifying a User Group's library copies.

Soon after DISK LABEL was published, a fellow group member suggested a modification to enable text typed for a particular line to be used for the next label if desired. This was done in console basic and the original 5 sector program grew to 10 sectors.

In cleaning up and consolidating the code for this article, it was apparent that Extended Basic's "Accept At" statement would make the program a lot more user friendly. The program was rewritten and a 4 sector Tinygram is the result.

Using the program is very simple, just respond to the prompts. The program automatically limits the number of characters for the various lines of the label so that you cannot type in too long a line. If you notice a typing error after pressing enter, not to worry. Just continue with the other entries and for "How Many?" enter zero. You will be returned to the first line and need only to accept the defaults until the error is displayed. No need to retype, just correct the error.

I always enter 1 for a quantity at first and look over the label to see if it's as intended and then print the number of copies required. I often print a few extra copies for later use and either place them in the back up's jacket or in a label box. Saves reloading the blank labels at some future time just to print a label or two. If you trade many disks, the last line of the extra copies can be left blank.

Usage is not limited to disk labels. It has been used to identify binders of our User Group's newsletter library, name tags, place cards, bookplates, etc.

Epson compressed mode is 137 columns wide. Printers with other widths may change length of underlining. If so just change the TAB setting of the null strings for the respective lines. Epson's Emphasized mode takes precedence over Compressed and cancels it upon return to line 6. Your printer may require print code cancellation at the end of line 7.

The print codes are for the Epson RX-80 printer. If your printer requires different codes, the rest of characters, in order of appearance, are as follows:

[ESC=E$=CHR$(27)]

ESC&"E"	Emphasized
ESC&"8"	Double Strike
ESC&"-1"	Underline
ESC&"W1"	Expanded
ESC&"F"	Cancel Emphasized
CHR$(15)	Compressed
ESC&"-0"	Cancel Underline
ESC&"4"	Italics
ESC&"5"	Cancel Italics

```
1 ! *** DISK LABEL II ***
  A Tinygram by Ed Machonis
    QB-99ers, Bayside, NY

2 OPEN #1:"PIO"

3 DISPLAY AT(3,1)ERASE ALL:"
DISK NAME?":D$ :: ACCEPT AT(
4,1)SIZE(-17):D$ :: DISPLAY
AT(7,1):"Continued?":C$ :: A
CCEPT AT(8,1)SIZE(-17):C$

4 DISPLAY AT(11,1):"TYPE?":T
$ :: ACCEPT AT(12,1)SIZE(-28
):T$ :: DISPLAY AT(15,1):"RE
MARKS?":R$ :: ACCEPT AT(16,1
)SIZE(-28):R$ :: E$=CHR$(27)

5 DISPLAY AT(19,1):"YOUR NAM
E?":N$ :: ACCEPT AT(20,1)SIZ
E(-28):N$ :: INPUT "HOW MANY
 COPIES? ":Q :: FOR J=1 TO Q

6 PRINT #1:E$&"E";E$&"8";E$&
"-1";E$&"W1";TAB((18-LEN(D$)
)/2);D$;TAB(18);"";TAB((18-L
EN(C$))/2);C$;TAB(18);"";E$&
"F";CHR$(15);TAB(2);T$;

7 PRINT #1:TAB(30);"";TAB(2)
;R$;TAB(30);"";E$&"-0";E$&"4
";TAB((30-LEN(N$))/2);N$;E$&
"5" :: NEXT J :: GOTO 3
```

TINY CALC

A TIny Gram

By Ed Machonis

Here's the first TIny Gram for 1989 and the New Year is only 15 days old. Just in time for checking some of that creative bookkeeping on the old 1040. In fact it was written with income taxes in mind. Since retiring, I've missed access to a calculator with a printed tape. Checking calculations on tax forms with a pocket calculator often wound up in a "Best two out of three" routine.

TINY CALC will do the four basic math operations, Add, Subtract, Multiply and Divide. Output can be directed to screen or printer. (The hard copy can be annotated and placed in your tax return folder for future reference if need be.) There is also a separate register or memory for cumulative storage of your totals. Display of entries is limited to 8 digits plus 2 decimals; totals can display an additional digit. Larger numbers will not be displayed but they remain in the computer for computations. If you need greater capacity than this, you shouldn't be doing your own taxes!

Operation is very simple. Just type the first number, <Enter>; type the math operator, +, -, * or /, <Enter>; type the second number and <Enter>. The operation will be performed. If you wish to continue the operation, no need to re-enter the previous total, just type in the next operator, <Enter> and the next number. When adding a column of numbers, and directing output to a printer, I like to enter a zero as the first number so that all entered numbers appear in the second column on the hard copy.

You can of course do a series of separate operations by repeatedly entering first number, operator, second number.

Once an operator has been selected, it will remain as a default until a new operator is selected. If you multiplied two numbers and wish to do further multiplications, you only need to enter the two numbers to be multiplied. Initial default if no operator has been selected is addition.

Any time you wish to store a total in memory, just <Enter> "M". Your total will be added to whatever number is in memory at the time.

Any time you wish to clear the calculator, just <Enter> a "C". This will also clear the Memory.

The value of the number in Memory is printed with each printout. If you want to take the number in Memory to use in an operation, you will have to leave the TIny Gram world and make these changes to Line 5:

Add an "R" after MC in the VALIDATE list and add the following code at the end of Line 5:

```
 ELSE IF X=82 THEN S=M :: M=
0 :: GOTO 9
```

Pressing "R" instead of entering the second number will take the contents of the Memory and use it as the second number. Memory will be cleared. Pressing "R" instead of the first number will have the same effect as pressing "C".

In the printout, all numbers are rounded to two decimal places and to the nearest hundredth. Calculations are performed with the full decimal. In the hardcopy, all numbers are right justified. As you can see, a screenful of TI Basic code can do a lot of work.

If you want to make sense out of this dish of spaghetti, the cast of characters are:

P=File number to print to.
N$=Your entry; number or operator.
X=ASCII value of first character in N$
F=First Number
K=ASCII value of operator.
S=Second number
T=Total
L=fLag; Set to 1 when first number is entered.
M=Memory; designated M1 on printout.

Due to sales of public domain software by certain distributers, a copyright notice has been placed on this program. It may be freely distributed provided no fee of any kind is charged. This article and/or the program listing may be published in newsletters of non profit User Groups.

```
1 ! ******* TINY CALC ********
    *     A TIny Gram      *
    *    Copyright 1989    *
    *    By Ed Machonis    *
    **QB-99ers, Bayside NY**

2 DEF X=ASC(N$):: CALL CLEAR

3 INPUT "C=CLEAR  M=STORE IN
 MEMORY  SCRN=0 PIO=1 0/1?":
P :: IF P THEN OPEN #P:"PIO"

4 F,S,T,K,L,M=0

5 ACCEPT VALIDATE(NUMERIC,"*
/,MC"):N$ :: IF N$="" THEN 5
 ELSE IF X=67 THEN 4 ELSE IF
 X=77 THEN M=M+T :: GOTO 10

6 IF X<48 AND X<>46 THEN K=X
 :: IF T<>0 AND F=0 THEN F=T
 :: L=1 :: GOTO 5

7 IF L=0 AND(X>47 OR X=46)TH
EN F=VAL(N$):: L=1 :: GOTO 5

8 IF L AND(X>47 OR X=46)THEN
 S=VAL(N$)ELSE 5

9 IF K=42 THEN T=F*S ELSE IF
 K=47 THEN T=F/S ELSE IF K=4
5 THEN T=F-S ELSE T=F+S

10 PRINT #P,USING "########.
##    * ########.## = ######
###.##            M1=######
###.##":F,CHR$(K),S,T,M :: L
,F,S=0 :: GOTO 5
```

COLISTER

A TINYGRAM

by Ed Machonis

Another 28 column lister? Why not? This one happens to be my favorite and not just because I wrote it. I like it because it does the job the way I want it done, but then I wrote it that way.

At the time I wrote COLISTER, I had no access to any program that could do what I wanted done, which was to be able to list a program to disk or printer in 28 column format, the way it appears on the screen.

A 28 column listing makes it easier for the reader to type in the program with less chance for error. It also makes it simpler to check for errors should any creep in. One only has to check the end of each line as it appears on the screen against the printed listing to see if any characters were omitted or added. (Home Computer magazine never did learn this lesson.)

But the biggest reason is that it not only saves the work of typing in a program in 28 column format, but it eliminates the chance for typing errors. By letting the computer do the work, nothing can go wrong. (If you believe this, I have a fantastic deal on a Bridge I'd like to tell you about!)

Why not just LIST to Printer or Disk? It's not that simple. The computer will list the program in 80 column format. Why not set the printer's right margin at 28? It will work up to a point. The point being a program line of more than 80 characters. The computer will send a carriage return after the 80th character and start printing the rest of the code on a new line. Listing to disk will also give you an 80 column listing.

Since I originally wrote this program several years ago, two programs that do the same work have been brought to my attention. One is 28 Column Converter by Jim Peterson, published in Tigercub Tips #18, and the other is COLIST, a Fairware program by the McGoverns. Both are very nice programs and you may well find them more useful to you than the one presented here. (I had originally named my program COLIST but have since renamed it COLISTER to avoid confusion.)

COLISTER has a couple of features not available in the other programs. First, it will print a blank line between program lines. I feel this makes it easier to "read" the program, especially the spaghetti code I am prone to. It facilitates picking out a line number in the middle of the program when following those GOTOs and orELSEs.

Second, it TABs the output 6 spaces. This centers the listing when merged into 40 column text in TI-Writer's Editor, and provides a margin so hard copies can be loose leaf bound.

COLISTER does not require that a program's line numbers be resequenced in order to list it. A lot of my program lines are numbered from 1 to 10. Default resequencing (100,10) would sometimes destroy their Tinygram status. (COLISTER is a good example. One Tinygram "trick" is to use single digit line numbers to gain a few extra character spaces for your code.)

COLISTER will print to either disk or printer. Listings printed to disk can the be merged with text in TI-Writer's Editor. Do not print the listing through the Formatter unless you have modified your Formatter file to ignore the special format command characters that are also often found in programs.

This Tinygram uses only 4 sectors of disk space, which can be reduced to 3 sectors by deleting Line 1. It earns its keep on my SSSD utility disk. (Small is Beautiful)

Using COLISTER is very simple. First, load into memory the program you want to list. Next make a DV 80 listing by typing LIST "DSKn.FILENAME". Don't use the same filename as the program or the listing can overwrite the program.

Then load and RUN COLISTER. At the first prompt, enter the DSK number and the filename used above. For the second prompt, enter the print device name. This can be either PIO, RS232, or DSKn.FILENAME2. Again, use a different filename if reading from and writing to the same drive.

If you don't want the blank line between program lines, just change the FOR statement in Line 8 to read: FOR I=0 TO L-1. The TAB setting in this line can also be changed or eliminated, as desired. If for some reason you want a listing with a different width, say 40 columns for those "other" owners, just change the value of C in Line 5. (The reason it's in Line 5, and being constantly updated, is because that's where the room was. Another Tinygram "trick".)

If you prepare program listings for newsletters, I think you'll find this program useful. The algorithm used to detect a new line number is relatively unsophisticated. It hasn't failed me yet, but I'm sure that someone, someday will write code that will trip it up. For that reason it is well to always look over the output to be sure that lines have not been split or joined when they should not have been.

```
1 !   *** COLISTER ***
 A Tinygram by Ed Machonis
   QB-99ers, Bayside, NY

2 PRINT :"1st LIST your prog
ram to diskThen RUN COLISTER"

3 PRINT ::"INPUT FILENAME?
ex:DSKn.LIST" :: INPUT F$ ::
 INPUT "OUTPUT FILENAME? ex:
 PIO or DSKn.LIST28 ";P$

4 OPEN #1:F$,INPUT :: OPEN #
3:P$,OUTPUT :: ON ERROR 10

5 C=28 :: LINPUT #1:A$ :: IF
 LEN(A$)<80 THEN 8

6 LINPUT #1:B$ :: IF VAL(SEG
$(A$,1,POS(A$," ",2)))<VAL(S
EG$(B$,1,POS(B$," ",2)))THEN
 F=1 :: GOTO 8

7 A$=A$&B$ :: IF LEN(B$)>=80
 THEN 6

8 A=LEN(A$):: L=A/C+.99 :: F
OR I=0 TO L :: PRINT #3:TAB(
6);SEG$(A$,1+I*C,C):: NEXT I
 :: IF EOF(1)AND F=0 THEN CL
OSE #1 :: CLOSE #3 :: END

9 IF F=1 THEN F=0 :: A$="" :
: GOTO 7 ELSE 5

10 ON ERROR 10 :: RETURN 7
```