DR. TAMASHIRO'S

TI TRAVELS

1984 - 1988

INDEX

DIRECTORY INDEX

ELECTRONIC TYPEWRITER (DSK1.TYPEWRITER) (DSK1.TYPE/PIO)

SPEAK-N-PROOF (DSK1.SPEAK1) (DSK1.SPEAK2)

USING THE 40-COLUMN TEXT SCREEN IN EXTENDED BASIC (DSK1.FORTY)
(DSK1.FORTYSAMPL) (DSK1.FORTYTALK) (DSK1.FORTYTEST)

FIREWORKS (DSK1.FIREWORKS) (DSK1.FIRESAMPLE)

SCREENBORDER (DSK1.BORDER) (DSK1.BORDERTEST)

PLACING A BOX AROUND A WORD (DSK1.BOX) (DSK1.BOXTEST1)
(DSK1.BOXTEST2)

CENTERING A STRING (DSK1.CENTER) (DSK1.CENTERTST1)
(DSK1.CENTERTST2)

ELIMINATING SCREEN SCROLL IN TI-BASIC (DSK1.DISPLAY)
(DSK1.DISPLAYTST)

ACCEPTING INPUT ANYWHERE ON THE SCREEN IN TI-BASIC (DSK1.ACCEPT)
(DSK1.ACCEPTTEST)

AUTOMATIC WORD WRAP (DSK1.WRAP) (DSK1.WRAPTEST)

AUTOMATIC WORD WRAP ON INPUT (DSK1.WRAP) (DSK1.WRAPTEST)

THE NUMBER MOTHER (DSK1.MOTHER)

THE BASIC NOTEPAD (DSK1.NOTEPAD)

THE MESSAGE ENCODER (DSK1.ENCODER)

QUICK LABEL (DSK1.LABEL)

WHIRLS AND FLIPS: UPSIDE DOWN TYPING (DSK1.FLIPS) (DSK1.WHIRLIGIG)

ELECTRONIC DATEBOOK (DSK1.DATEBOOK) (DSK1.TDOC)

TWO-WAY COMMUNICATIONS FOR X-BASIC (DSK1.COMM)

AUTOMATED ADDRESS BOOK (DSK1.ADDRESS)

MANHATTAN STREET FINDER (DSK1.STREET) (DSK1.STREETTNDY)

THE WORD COUNTER (DSK1.WORD)

A CORONARY RISK ANALYSIS PROGRAM (DSK1.CORONARY)

COMPUTING FOR YOUR RETIREMENT INCOME (DSK1.RETIREMENT)

CONVERTING MINI-MEMORY CASSETTE PROGRAMS TO DISK (DSK1.TAPE)

ELECTRONIC TYPEWRITER

Roy T. Tamashiro, Ed.D.

When your typing job is small and you do not want to go through the many steps involved in using a word processor, the ELECTRONIC TYPEWRITER program below is handy. It is easier to learn, faster to load and simpler to operate than a regular word processor. The program emulates an electronic memory typewriter, in which you can edit a line of text before the line gets typed. Thus the program is suitable for small typing jobs(letters, notes, memos) or when one is not fully knowlegeable about using a word processor. ELECTRONIC TYPEWRITER uses the 40-column text screen, which is ordinarily not available in BASIC. This is the mode used in programs such as TI-WRITER and Multiplan. ELECTRONIC TYPEWRITER contains many features of a regular word processor, though it only handles 80 characters at a time. To run this program, the following are required: (1)TI-Extended BASIC, (2)32-K Memory Expansion, (3)Printer accessed via RS-232 interface or Thermal printer. Modify line 510 to suit your printer. When the program is RUN, type in a line of up to 80 characters. The 80 characters are displayed on two rows on the screen. The text does not get typed until you press the <ENTER> key. Before pressing <ENTER>, you can edit the line using the following function keys:

<LEFT ARROW> (FCTN-S)	Moves the cursor to the left
<RIGHT ARROW> (FCTN-D)	Moves the cursor to the right
<ERASE> (FCTN-3)	Deletes the entire line of text on the screen
<INS> (FCTN-2)	Insert space for additional characters
<DEL> (FCTN-1)	Delete a single character

If you want to skip a line on your paper, just press the <ENTER> key with no text showing on the prompt line. To exit the program type END in all upper case letters, and then press <ENTER>.

```
100 !**************************
110 !* ELECTRONIC TYPEWRITER *
120 !**************************
130 !AUTHOR: ROY T.  TAMASHIRO
140 ! 150 !DATE: SEPTEMBER 1984
160 !FOR:TI-XBASIC w/ Memory Expansion and Printer
170 CALL INIT :: CALL LOAD(8196,63,248):: CALL LOAD(16376,84,89,80,69,32,32,48,1 90)
180 CALL LOAD(12288,0,0,0,1,48,36,255,0,255,0,255,0,255,0,255,0,255,0,255,0,255, 0,49,148)
190 CALL LOAD(12312,255,0,255,0,255,0,255,0,2,48,13,0,80,32,32,32,32,32,32,32,32 ,32,32,32)
200 CALL LOAD(12336,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32, 32,32,32,32)
210 CALL LOAD(12360,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32, 32,32,32,32)
220 CALL LOAD(12384,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32, 32,0,255,0)
230 CALL LOAD(12408,42,32,69,76,69,67,84,82,79,78,73,67,32,84,89,80,69,87,82,73, 84,69,82,32)
240 CALL LOAD(12432,42,32,69,110,116,101,114,32,97,32,108,105,110,101,32,111,102 ,32,116,101,120,116,58,32)
250 CALL LOAD(12456,40,84,121,112,101,32,69,78,68,32,119,104,101,110,32,100,111, 110,101,46,41,32,2,224)
260 CALL LOAD(12480,131,224,2,1,240,129,216,1,131,212,216,1,140,2,6,193,216,1,14 0,2,2,1,244,135)
270 CALL LOAD(12504,216,1,140,2,6,193,216,1,140,2,2,224,48,0,4,192,2,1,128,0,4,3 2,32,32)
280 CALL LOAD(12528,5,128,2,128,3,192,22,250,2,0,0,7,2,1,138,0,4,32,32,32,5,128, 2,128)
290 CALL LOAD(12552,0,32,22,250,2,0,0,87,4,32,32,32,5,128,2,128,0,112,22,250,2,0 ,0,47)
300 CALL LOAD(12576,4,193,4,194,208,98,48,120,2,33,96,0,4,32,32,32,5,128,5,130,2 ,130,0,25)
310 CALL LOAD(12600,22,245,2,1,32,0,2,2,0,1,216,129,48,36,5,130,2,130,0,81,22,25 0,2,0)
320 CALL LOAD(12624,1,184,4,193,4,194,208,98,48,146,2,33,96,0,4,32,32,32,5,130,5 ,128,2,130)
330 CALL LOAD(12648,0,22,22,245,2,0,1,224,4,194,208,98,48,168,2,33,96,0,4,32,32, 32,5,130)
340 CALL LOAD(12672,5,128,2,130,0,22,22,245,2,0,2,48,200,0,48,32,6,160,50,240,19 2,32,48,32)
350 CALL LOAD(12696,2,1,126,0,4,32,32,32,2,0,5,0,216,0,131,116,4,193,2,0,32,0,2, 2)
360 CALL LOAD(12720,255,0,4,32,32,28,144,32,131,124,19,26,144,160,131,117,19,243 ,2,3,0,5,6,3)
370 CALL LOAD(12744,2,1,9,192,6,1,22,254,4,32,32,28,144,32,131,124,19,11,144,160 ,131,117,19,228)
380 CALL LOAD(12768,4,32,32,28,192,195,22,239,152,32,48,34,131,117,22,220,216,32 ,131,117,48,34,4,193)
390 CALL LOAD(12792,208,96,131,117,2,129,8,0,22,15,4,194,192,160,48,32,2,130,2,4 8,22,2,4,96)
400 CALL LOAD(12816,49,144,6,2,200,2,48,32,6,160,50,240,4,96,49,144,2,129,9,0,22 ,2,4,96)
410 CALL LOAD(12840,50,178,2,129,13,0,22,2,4,96,50,206,2,129,7,0,22,2,4,96,48,19 0,2,129)
420 CALL LOAD(12864,4,0,22,20,192,32,48,32,2,32,253,209,2,2,0,80,192,194,6,3,216 ,163,48,36)
430 CALL LOAD(12888,48,36,6,2,128,3,22,249,2,1,32,0,216,193,48,36,4,96,49,144,2, 129,3,0)
440 CALL LOAD(12912,22,20,192,96,48,32,2,33,253,209,192,193,2,0,0,80,5,131,216,9 9,48,36,48,36)
```

```
450 CALL LOAD(12936,5,129,128,3,22,249,2,2,32,0,216,194,48,36,4,96,49,144,2,129, 32,0,26,251)
460 CALL LOAD(12960,2,129,126,0,27,248,192,224,48,32,2,35,253,209,216,193,48,36, 6,160,50,240,192,96)
470 CALL LOAD(12984,48,32,2,129,2,127,17,2,4,96,49,144,5,129,200,1,48,32,4,96,49 ,144,120,32)
480 CALL LOAD(13008,131,124,131,124,4,192,2,1,0,1,2,2,48,36,4,32,32,16,120,32,13 1,124,131,124)
490 CALL LOAD(13032,2,224,131,224,4,96,0,112,2,0,2,48,2,2,0,1,4,193,208,98,48,36 ,2,33)
500 CALL LOAD(13056,96,0,4,32,32,32,5,128,5,130,2,130,0,81,22,245,4,91)
510 OPEN #1:"RS232.BA=9600.DA=8"
520 CALL LINK("TYPE",W$):: IF SEG$(W$,1,3)="END" THEN CLOSE #1 :: CALL PEEK(2,A, B):: CALL LOAD(-31804,A,B)
530 PRINT #1:W$ :: GOTO 520
```

"SPEAK-N-PROOF": THE TALKING PROOFREADER

Roy T. Tamashiro, Ed.D.

Nobody types perfectly. Whether you are writing a document on a word processor such as TI-WRITER, or typing in a BASIC program, mistakes are virtually inevitable. It is easy to omit a key word or transpose numbers. It can be quite difficult to detect typing errors in a BASIC program especially when there are long series of numbers or symbols. When you cannot find a willing friend to read back to you what you typed, you can call upon "SPEAK-N-PROOF" to help you proofread your TI-WRITER text file or BASIC (or Extended BASIC) Program. The voice synthesizer reads what you typed, while you visually check this against the original document or program listing. Since your eyes focus on one listing rather than two, you are less likely to lose your place as you check your typing.

To use SPEAK-N-PROOF, you must have the program list or text file to be proofed in DISPLAY,VARIABLE 80 format. This is the format used by programs such as TI-WRITER and EDITOR-ASSEMBLER. To proofread a TI-WRITER document, simply SAVE the file from TI-WRITER's Text Editor. To proofread a program listing, load the program to be proofed and LIST it to the disk using the following command:

```
LIST "DSK1.PROG-LIST"
```

After your file has been recorded on disk, insert the EXTENDED BASIC module and load SPEAK-N-PROOF PART 1. This program creates two new disk files which SPEAK-N-PROOF PART 2 will read, so be sure you have enough space on Disk 1. When you run the program, type the name of the file you want to proofread. (To proofread a program, note that you must type the listed filename, i.e. DSK1.PROG-LIST .) If you want the words which are typed in lower case characters to be pronounced as words, instead of pronounced as individual letters, then press <Y> when you are asked "Is text in both Upper and Lower case?(Y/N)".

When PART 1 ends, insert the Terminal Emulator II module, select TI-BASIC and load SPEAK-N-PROOF PART 2. RUN the program. When you see the question "Is text in both Upper and Lower case?(Y/N)", press <Y> only if you also pressed <Y> in Part 1. If you want to inspect each line of text, select <Y> when you see the message "Pause after each line?(Y/N)". The voice synthesizer then begins speaking your text file out loud. The text also appears on the screen. Jot down the errors or discrepancies you find. You can then make the necessary corrections using the original editor(TI-WRITER, BASIC, Extended BASIC, etc.) on which you typed.

```
100 ! *****************
110 ! * SPEAK-N-PROOF *
120 ! *    PART 1     *
130 ! *****************
140 !Author: Roy T. Tamashiro, Ed.D.
150 !September 1984/X-BASIC
160 CALL CLEAR :: A$="*****************" :: PRINT TAB(6);A$:TAB(6);"* SPEAK-N
-PROOF *":TAB(6);"*    PART 1     *":TAB(6);A$: : : : :
170 INPUT "File Name: DSK1.":B$ :: B$="DSK1."&B$
180 PRINT :"Is text in both Upper and Lower case?(Y/N)"
190 CALL KEY(0,U,S):: IF (U<>78)+(U<>89)=-2 THEN 190
200 CALL CLEAR :: PRINT "Reformatting File...":"Please wait."
210 OPEN #1:B$,INPUT ,VARIABLE 80
220 OPEN #2:"DSK1.PROOF-FILE",OUTPUT,INTERNAL,VARIABLE 90
230 IF U=89 THEN OPEN #3:"DSK1.UPPER",OUTPUT,INTERNAL,VARIABLE 90
240 LINPUT #1:W$ :: IF U=78 THEN 270
250 X$="" :: FOR I=1 TO LEN(W$):: A$=SEG$(W$,I,1):: IF (ASC(A$)>96)AND(ASC(A$)
<123)THEN A$=CHR$(ASC(A$)-32)
260 X$=X$&A$ :: NEXT I :: PRINT #3:X$
270 PRINT #2:W$ :: IF EOF(1)=0 THEN 240
280 CLOSE #1 :: CLOSE #2 :: IF U=89 THEN CLOSE #3
290 CALL CLEAR :: PRINT "Reformatting completed.": :"RUN SPEAK-N-PROOF PART 2 with Terminal Emulator." :: END
```

4

```
100 REM #################
110 REM # SPEAK-N-PROOF #
120 REM #    PART 2     #
130 REM #################
140 REM AUTHOR: Roy T. Tamashiro, Ed.D.
150 REM September 1984/Speech Synth. and Term. Emulator II required.
160 CALL CLEAR
170 A$="#################"
180 PRINT TAB(6);A$:TAB(6);"# SPEAK-N-PROOF #":TAB(6);"#    PART 2     #":TAB
(6);A$ : : : : :
190 PRINT "Is text in both Upper and   Lower case?(Y/N)"
200 CALL KEY(0,U,S)
210 IF (U<>89)+(U<>78)=-2 THEN 200
220 PRINT :"Pause after each line?(Y/N)"
230 CALL KEY(3,P,S)
240 IF (P<>89)+(P<>78)=-2 THEN 230
250 CALL CLEAR
260 IF U=78 THEN 280
270 OPEN #3:"DSK1.UPPER",INPUT ,INTERNAL,VARIABLE 90
280 OPEN #1:"DSK1.PROOF-FILE",INPUT ,INTERNAL,VARIABLE 90
290 OPEN #2:"SPEECH",OUTPUT
300 INPUT #1:W$
310 PRINT W$
320 IF U=78 THEN 340
330 INPUT #3:W$
340 PRINT #2:W$
350 IF P=78 THEN 380
360 INPUT "Press <ENTER> ":E$
370 CALL HCHAR(23,1,32,32)
380 IF EOF(1)=0 THEN 300
390 CLOSE #1
400 CLOSE #2
410 IF U=78 THEN 430
420 CLOSE #3
430 END
```

USING THE 40-COLUMN TEXT SCREEN IN EXTENDED BASIC

Roy T. Tamashiro, Ed.D.

I have often wanted to use a 40-column screen in a BASIC program because you can put 43 percent more text on it than on the normal 28-column screen in BASIC. However, the 40-column screen is not available in TI-BASIC or Extended BASIC, even though this 40-column screen is present in the TI-99/4A ROM. The program below makes it possible to create and implement programs in Extended BASIC on the 40-column screen. The 32-K Memory Expansion, the Extended BASIC cartridge, and cassette or disk system are required. To access the 40-column screen, type in and save the program "FORTY-COLUMN TEXT SCREEN FOR X-BASIC"(below), on disk or cassette. Then compose your program for the 40-column screen.

To turn on the 40-column screen in your program, use the instruction, CALL LINK("FORTY"). You may use most of the normal Extended BASIC instructions, but make the following substitutions:

EXTENDED BASIC INSTRUCTIONS:	REPLACE WITH THIS FORMAT:
CALL CLEAR	CALL LINK("CLS")
INPUT, LINPUT or ACCEPT AT	CALL LINK("INPUT",Row#(1-24),Column#(1-40),StringVariable)
PRINT or DISPLAY AT	CALL LINK("DISPL",Row#(1-24),Column#(1-40),StringVariable)
CALL COLOR, CALL SCREEN	CALL COLORS(Foreground,Background)

(If you wish to change the colors of the characters on the screen, add Lines 16000 to 16020 in the SAMPLE PROGRAM below to your program as the last routine in your program. This makes it possible to use the above CALL COLORS(Foreground,Background) instruction. Use the color codes(1-16) normally used in BASIC to designate forground and background colors. See example in the "SAMPLE PROGRAM" below.)

Do not use SPRITE instructions(CALL SPRITE, CALL MAGNIFY, CALL COINC, etc.) on the 40-column screen. Other instructions such as CALL HCHAR and CALL GCHAR work, but since they are oriented to the 32-column screen rather than the 40 column screen, the locations are confusing.

To switch back to the normal 32-column screen in Extended BASIC, use the instructions, CALL LINK("BSCRN"). Be sure to include this CALL LINK when you exit the Extended BASIC program--otherwise your program will not be visible on the screen. (See Line 200 in the "SAMPLE PROGRAM" below.)

Note that the new INPUT and DISPLAY instructions use string variables only. Thus, numeric variables must be converted before or after these CALLs. For example, to DISPLAY a numeric variable, use the following model:

```
210 N=1::N$=STR$(N)::CALL LINK("DISPL",24,1,N$)
```

(In this example, the value in N, which is 1, is converted to the string variable N$ and displayed at row 24, column 1.)

Or, to INPUT a numeric variable:

```
230 CALL INPUT(24,1,N$)::N=VAL(N$)
```

(In this example, a number is accepted at row 24, column 1 and assigned to N.)

To RUN your program, first load and RUN the program below("FORTY-COLUMN TEXT SCREEN FOR X-BASIC"), then load and RUN your program. As long as you do not use CALL INIT or load another Assembly Languge program, you can run your program without re-RUNning the "FORTY-COLUMN TEXT" Program.

```
100 !************************
110 !* FORTY-COLUMN TEXT  *
120 !* SCREEN FOR X-BASIC *
130 !************************
140 !AUTHOR: ROY T. TAMASHIRO, ED.D
150 !DECEMBER 1984,X-BASIC W/ MEMORY EXPANSION
160 ! SEE ACCOMPANYING INSTRUCTIONS FOR USING THIS PROGRAM
170 CALL INIT
180 CALL LOAD(8196,63,216):: CALL LOAD(16344,66,83,67,82,78,32,50,108,68,73,83,8 0,76,32,48,190)
190 CALL LOAD(16360,73,78,80,85,84,32,49,36,67,76,83,32,32,32,48,78,70,79,82,84, 89,32,48,38)
200 CALL LOAD(12288,8,31,16,0,50,190,0,0,0,0,1,108,51,188,0,0,0,0,0,2,12,50,11 6)
210 CALL LOAD(12312,215,32,47,190,215,32,47,191,13,0,1,108,2,107,2,224,131,224,2 ,1,240,129,216,1)
220 CALL LOAD(12336,131,212,216,1,140,2,6,193,216,1,140,2,2,1,245,135,216,1,140, 2,6,193,216,1)
230 CALL LOAD(12360,140,2,4,96,48,86,2,224,48,0,6,160;48,98,4,224,131,124,2,224, 131,224,4,96)
240 CALL LOAD(12384,0,112,4,192,2,1,128,0,4,32,32,32,5,128,2,128,3,192,22,250,4, 91,2,1)
250 CALL LOAD(12408,0,1,4,192,4,32,32,12,200,32,131,74,48,36,192,224,48,36,2,67, 0,255,2,2)
260 CALL LOAD(12432,255,216,2,34,0,40,6,3,22,252,200,2,48,34,2,1,0,2,4,192,4,32, 32,12)
270 CALL LOAD(12456,200,32,131,74,48,36,192,96,48,36,2,65,0,255,6,1,168,1,48,34, 4,91,2,224)
280 CALL LOAD(12480,48,0,6,160,48,118,2,1,255,0,216,1,50,189,2,1,0,3,4,192,2,2,5 0,189)
290 CALL LOAD(12504,4,32,32,20,4,197,209,96,50,189,6,197,2,6,50,190,192,32,48,34 ,6,160,49,28)
300 CALL LOAD(12528,4,193,192,86,2,33,96,0,4,32,32,32,5,128,6,160,49,28,6,5,19,9 ,6,193)
310 CALL LOAD(12552,2,33,96,0,4,32,32,32,5,198,5,128,6,5,22,236,4,96,48,86,2,128 ,3,192)
320 CALL LOAD(12576,21,251,4,91,2,224,48,0,2,2,1,0,2,1,32,0,216,129,50,190,6,2,2 2,252)
330 CALL LOAD(12600,6,160,48,118,2,1,0,255,192,32,48,34,160,64,2,129,3,192,18,2, 2,1,3,192)
340 CALL LOAD(12624,200,1,48,36,4,196,193,64,2,1,32,0,217,1,50,191,2,1,126,0,4,3 2,32,32)
350 CALL LOAD(12648,2,1,5,0,216,1,131,116,6,160,50,34,216,32,131,117,48,32,4,193 ,208,96,131,117)
360 CALL LOAD(12672,192,5,2,129,13,0,22,18,2,1,128,0,4,32,32,32,4,224,131,124,4, 192,2,1)
370 CALL LOAD(12696,0,3,6,196,216,4,50,190,2,2,50,190,4,32,32,16,4,96,48,86,2,12 9,7,0)
380 CALL LOAD(12720,22,13,2,1,32,0,217,1,50,191,2,33,96,0,4,32,32,32,6,0,6,4,22, 245)
390 CALL LOAD(12744,4,96,49,36,2,129,8,0,22,17,2,1,32,0,217,1,50,191,2,33,96,0,4 ,32)
400 CALL LOAD(12768,32,32,6,0,6,4,128,32,48,34,18,181,5,132,5,128,4,96,49,86,2,1 29,9,0)
410 CALL LOAD(12792,22,2,2,1,32,0,2,129,32,0,17,169,217,1,50,191,2,33,96,0,4,32, 32,32)
420 CALL LOAD(12816,5,132,5,128,136,0,48,36,18,158,6,0,6,4,4,96,49,86,4,193,2,0, 32,0)
430 CALL LOAD(12840,2,2,255,0,4,32,32,28,144,32,131,124,19,26,144,160,131,117,19 ,243,2,3,0,5)
440 CALL LOAD(12864,6,3,2,1,9,192,6,1,22,254,4,32,32,28,144,32,131,124,19,11,144 ,160,131,117)
450 CALL LOAD(12888,19,228,4,32,32,28,192,195,22,239,152,32,48,32,131,117,22,220 ,4,91,2,224,48,0)
460 CALL LOAD(12912,6,160,48,98,2,0,3,0,4,193,4,32,32,32,5,128,2,128,3,192,22,25 0,2,0)
470 CALL LOAD(12936,224,1,216,0,131,212,6,192,4,32,32,48,2,0,3,32,4,32,32,48,2,0 ,7,23)
480 CALL LOAD(12960,4,32,32,48,2,0,8,0,2,1,16,0,4,32,32,32,5,128,2,128,8,31,22,2 50)
490 CALL LOAD(12984,4,96,48,86,0,255,0,32,32)
```

Here is a sample program which demontrates the features of the 40-column access program. As noted previously, RUN the above program first, then RUN this "SAMPLE PROGRAM".

```
100 REM *SAMPLE PROGRAM *
110 CALL LINK("FORTY")
120 CALL LINK("CLS"):: CALL LINK("DISPL",1,15,"HELLO THERE")
130 CALL LINK("DISPL",10,1,"Foreground Color--Enter 1-16:")
140 CALL LINK("INPUT",10,35,F$):: F=VAL(F$)
150 CALL LINK("DISPL",12,1,"Background Color--Enter 1-16:")
160 CALL LINK("INPUT",12,35,B$):: B=VAL(B$)
170 CALL COLORS(F,B)
180 CALL LINK("DISPL",23,1,"Enter <1> to go on; or <2> to exit:")
190 CALL LINK("INPUT",23,38,K$):: IF K$="1" THEN 120
200 CALL LINK("BSCRN"):: END
16000 SUB COLORS(F,B)
16010 CALL LOAD(12350,16*(F-1)+(B-1)):: CALL LINK("FORTY")
16020 SUBEND
```

---

* Author's Note: Steve Vukelich published routines for accessing the 40-column screen in the July/August 1984 issue of the McDonnell-Douglas TI-SIG Group Newsletter. While the routines in this article are similar to Vukelich's, his routines could access only 19 of the 24 lines on the screen, with "strange characters" appearing in rows 20-24. The routines provided in this article has eliminated these "strange characters", so the entire screen is available for DISPLAY and INPUT. In addition, this set of routines provides a utility for clearing the entire screen, and for controlling the color of the characters and screen.

UTILITY SUBROUTINES IN BASIC:INTRODUCTION

by Roy Tamashiro

The subroutines I am presenting in this and future articles are versatile aids in writing programs in TI-BASIC (and TI-Extended-BASIC). Most of these routines are helpful in formatting text on the screen. To use the subroutines, follow the instructions provided with each routine. There is a "sample main program" to illustrate the application of each subroutine. You may call these subroutines, as many times in the program as you need them. Note that the variables listed in each subroutine may change after the routine is completed, so these variables should not be used elsewhere in the program indiscriminately.

FIREWORKS

by Roy T. Tamashiro, Ed.D.
Webster University

One simple yet very attractive reinforcement for correct answers in a computer-based instructional activity is the following subroutine which I call "FIREWORKS". This subroutine can be used in drill or quiz programs to indicate that a correct answer has been selected.

This subroutine works on either TI BASIC or EXTENDED BASIC.

```
2980 REM *FIREWORKS SUBROUTINE*
2990 REM *USES THE VARIABLES I,J
3000 RANDOMIZE
3010 FOR I=1 TO 15
3020 J=INT(14*RND)+2
3030 CALL SCREEN(J)
3040 CALL SOUND(50,110+J^3,2)
3050 NEXT J
3060 CALL SCREEN(4)
3070 INPUT "PRESS <ENTER> TO GO ON.  ":EN$
3080 RETURN
```

Here is a sample program which uses the fireworks subroutine. The correct answer to the problems gives you fireworks.

```
100 REM *SAMPLE PROGRAM*
110 I=I+1
120 CALL CLEAR
130 PRINT "WHAT IS 9 X";I;"?": : :
140 INPUT "ENTER YOUR ANSWER: ":AN
150 IF AN=9*I THEN 180
160 PRINT : : "NO, IT'S ";9*I
170 GOTO 190
180 GOSUB 3000
190 IF I<5 THEN 110
200 END
```

SCREEN BORDER

by Roy T. Tamashiro, Ed.D.
Webster University

This subroutine for TI BASIC or EXTENDED BASIC creates a red border around the edge of the screen after all the material to be displayed has been PRINTed to the screen. This routine uses character code 128 for the red border, so the program cannot use characters 128-135 for other purposes without changing the color definition.

```
980 REM * BORDER SUBROUTINE *
990 REM * USES THE VARIABLES K,S
1000 PRINT " PRESS <ENTER> "
1005 CALL COLOR(13,9,9)
```

```
1010 CALL HCHAR(1,1,128,32)
1020 CALL VCHAR(1,1,128,24)
1030 CALL VCHAR(1,2,128,24)
1040 CALL VCHAR(1,31,128,24)
1050 CALL VCHAR(1,32,128,24)
1060 CALL HCHAR(24,1,128,32)
1070 CALL HCHAR(23,18,30)
1080 CALL KEY(0,K,S)
1090 CALL HCHAR(23,18,32)
1100 IF (S=0)+(K<>13)THEN 1070
1110 CALL CLEAR
1120 RETURN
```

You can test this subroutine with the following sample program:

```
100 REM * TEST PROGRAM *
110 CALL CLEAR
120 A$="THIS IS A TEST"
130 PRINT TAB(7);A$: : :
140 PRINT TAB(7);A$: : :
150 PRINT TAB(7);A$: : :
160 PRINT : : :
170 GOSUB 1000
180 END
```

PLACING A BOX AROUND A WORD

by Roy T. Tamashiro, Ed.D.
Webster University

This routine places a red box around a string which you would like to highlight on the screen. You must assign your string of up to 28 characters to the variable **W$** before calling this subroutine. Also, the variable **R** must be the row number (3-21) where you wish to place the string and **C** is the column number.

```
2940 REM * TO BOX A WORD SUBROUTINE *
2950 REM * USES THE VARIABLES I,L,AS IN SUBROUTINE
2960 REM * USES VARIABLES R,C,W$ FROM MAIN PROGRAM
2970 REM *R=ROW NUMBER (3-22)
2980 REM *C=COLUMN NUMBER(MIN. 3, MAX. 28-LENGTH OF W$)
2990 REM *W$=THE STRING TO BE BOXED
3000 L=LEN(W$)
3010 CALL COLOR(13,9,9)
3020 FOR I=1 TO L
3030 AS=ASC(SEG$(W$,I,1))
3040 CALL HCHAR(R,C+I,AS)
3050 NEXT I
3060 CALL VCHAR(R-2,C-1,128,4)
3070 CALL HCHAR(R-2,C-1,128,L+4)
3080 CALL HCHAR(R+2,C-1,128,L+4)
3090 CALL VCHAR(R-2,C+2+L,128,4)
3100 RETURN
```

Of course, the color of the box may be changed if you change the color codes in line 3010. The subroutine may be called from program lines which designate the string to be displayed and its location. The following example illustrates this idea.

```
100 REM * SAMPLE CALLING PROGRAM #1 *
110 CALL CLEAR
120 W$="THIS IS A TEST."
130 R=10
```

```
140 C=12
150 GOSUB 3000
160 GOTO 160
```

Another use of this subroutine is to place a box around the word which the user enters when the program is RUN. The following sample program illustrates this.

```
100 REM * SAMPLE CALLING PROGRAM #2 *
110 CALL CLEAR
120 INPUT "ENTER A WORD(LIMIT 28 CHRS.) ":W$
130 INPUT "ROW NO.(3-22) ":R
140 PRINT "COL NO.(3-";28-LEN(W$);")";
150 INPUT C
160 CALL CLEAR
170 GOSUB 3000
180 GOTO 180
```

CENTERING A STRING

by Roy T. Tamashiro, Ed.D.
Webster University

There are many situations in which we would like to see a message displayed in the center of a line on the screen. This subroutine enables you to center a string on a line without having to figure out the starting column location. The subroutine does this calculation for you. The subroutine is as follows:

```
1980 REM * SUBROUTINE TO CENTER A STRING *
1990 REM * USES W$ FROM MAIN PROGRAM
2000 PRINT TAB(14-INT(LEN(W$)/2));W$
2010 RETURN
```

The string you want centered must be assigned to the variable W$ and it must not be longer than 28 characters. The following is a sample calling program.

```
100 REM * SAMPLE CALLING PROGRAM #1 *
110 CALL CLEAR
120 W$="THIS IS A TEST."
130 GOSUB 2000
140 END
```

This centering subroutine may be used in conjunction with strings entered when the program is RUN, such as via INPUT instructions. The followng is an illustration of this idea.

```
100 REM *SAMPLE CALLING PROGRAM #2 *
110 CALL CLEAR
120 INPUT "ENTER A WORD(MAX.: 28 CHRS.) ":W$
130 PRINT : : :
140 GOSUB 2000
150 END
```

A variation of this subroutine is its use in conjunction with the subroutine for "ELIMINATING SCREEN SCROLL IN TI-BASIC" (See separate instructions on this topic.). When these two subroutines are combined, a message can be centered on any line on the screen. To do this, add the two lines below to the "DISPLAY AT SUBROUTINE" described as Method #1 of the ELIMINATING SCREEN SCROLL subroutine. When you want to center a message, assign your message to the variable W$ and row number to R and call for the subroutine at line 1004.

```
1002 GOTO 1010
1004 C=15-INT(LEN(W$)/2)
```

UTILITY SUBROUTINES IN BASIC: INTRODUCTION

by Roy Tamashiro

The subroutines I am presenting in this and last months issue of THE BRIDGE are versatile aids in writing programs in TI-BASIC (and TI-Extended-BASIC). Most of these routines are helpful in formatting text on the screen. To use the subroutines, follow the instructions provided with each routine. There is a "sample main program"(last months issue) to illustrate the application of each subroutine. You may call these subroutines, as many times in the program as you need them. Note that the variables listed in each subroutine may change after the routine is completed, so these variables should not be used elsewhere in the program indiscriminately.

ELIMINATING SCREEN SCROLL IN TI-BASIC

by Roy T. Tamashiro, Ed.D.
Webster University

When the PRINT instruction is used in TI-BASIC, the material is written on the last row of the screen. Another PRINT statement moves the previous line display up the screen and the current string or number is displayed on the bottom of the screen. If you are displaying an entire screenful of information, it can be irritating to watch lines scroll up from the bottom of the screen, especially when there may be several pages of information such as in an instructional tutorial. The following listings illustrate two methods of eliminating this screen scroll in TI-BASIC. (NOTE: This scrolling display is not a problem in TI EXTENDED BASIC since the DISPLAY AT instruction may be used.)

METHOD 1: This approach simulates EXTENDED BASIC's DISPLAY AT instruction. In the main part of your program define the material you want to display as W$ and assign the row number to the variable R1. You can also specify a column number if you do not want the display to be at the left edge of the screen. Then call the subroutine at line 1000 in this example. (If you specify a column number, use GOSUB 1010.) Caution: W$ cannot be more than 28 characters.

```
970 REM *DISPLAY AT SUBROUTINE *
980 REM *USES VARIABLES R1,C,W$ FROM MAIN PROGRAM *
990 REM *USES THE VARIABLES I,L$ *
1000 C=2
1010 FOR I=1 TO LEN(W$)
1020 L$=SEG$(W$,I,1)
1030 CALL HCHAR(R1,C+I,ASC(L$))
1040 NEXT I
1050 RETURN
```

METHOD 2: This approach involves changing the screen background color so the scrolling is invisible, then changing the screen color again after the messages have appeared on the screen. This approach is much faster than Method 1 when you have a screenful of material. The program must be initialized with a routine to change the colors of all characters to be displayed. In this example, the characters are changed to light blue (color code 6) in line 130. Enter lines 110-150 at the beginning of your program. Then use PRINT statements to display the material for a single screen. Call for the subroutine (GOSUB 1000) after all the PRINT statements for the page.

```
100 REM * SAMPLE PROGRAM
110 CALL CLEAR
120 FOR I=1 TO 12
130 CALL COLOR(I,6,1)
140 NEXT I
150 CALL SCREEN(6)
160 A$="THIS IS A TEST!"
170 PRINT TAB(7);A$: : :
180 PRINT TAB(7);A$: : :
190 PRINT TAB(7L;A$: : :
200 PRINT : : : : : :
210 GOSUB 1000
```

The rest of your program continues between lines 210 and 980. The subroutine which switches the screen background color sted below.

```
980 END
990 REM * THE COLOR SWITCH SUBROUTINE *
1000 CALL SCREEN(16)
1010 INPUT "PRESS <ENTER> TO GO ON.  ":EN$
1020 CALL CLEAR
1030 CALL SCREEN(6)
1040 RETURN
```

ACCEPTING INPUT ANYWHERE ON THE SCREEN IN TI-BASIC

by Roy T. Tamashiro, Ed.D.
Webster University

When the INPUT instruction is given in TI-BASIC, the material is written on the last row of the screen. In designing a screen, you may want the INPUT to be accepted on the screen in locataions other than the last row. The following subroutine can be called to accept input on any row and column location on the screen you designate. In the main part of your program row number to the variable R1 and the column number to C. Then call the subroutine at line 2000 (or at 2010 if you defined your own column location). The input is returned to the main program in the variable name ND$.

```
1970 REM *ACCEPT AT SUBROUTINE*
1980 REM *USES VARIABLES C,R1 FROM MAIN PROGRAM
1990 REM *USES THE VARIABLES K,I,S,ND$
2000 C=3
2010 ND$=""
2020 K=0
2030 FOR I=C TO 30
2040 IF K=127 THEN 2190
2050 CALL HCHAR(R1,I,30)
2060 CALL KEY(0,K,S)
2070 IF K<>8 THEN 2130
2080 I=I-1
2090 CALL HCHAR(R1,I+1,32)
2100 IF I>C THEN 2050
2110 I=C
2120 GOTO 2050
2130 IF K<>9 THEN 2150
2140 K=32
2150 IF K<>13 THEN 2170
2160 K=127
2170 IF (K<32)+(S=0)THEN 2060
2180 CALL HCHAR(R1,I,K)
2190 NEXT I
2200 K=0
2210 FOR I=C TO 30
2220 IF K=127 THEN 2260
2230 CALL GCHAR(R1,I,K)
2240 IF K=127 THEN 2260
2250 ND$=ND$& CHR$(K)
2260 NEXT I
2270 RETURN
2280 STOP
```

To check if your subroutine works enter the following sample main program:

```
100 REM * SAMPLE MAIN PROGRAM *
110 CALL CLEAR
```

```
120 R1=5
130 C=10
140 GOSUB 2010
150 PRINT ND$
160 END
```

A cursor appears in row 5, column 10 and waits for you to type a message. That message is displayed at the bottom of the screen after you press <ENTER>.

AUTOMATIC WORD WRAP

by Roy T. Tamashiro
Webster University

Words split between two lines on the screen can be a nuisance, especially when a long phrase is entered by the user. The following subroutine eliminates the "split" word on the screen. This is generally referred to as "Automatic Word Wrap." The subroutine may be used in either TI BASIC or TI Extended BASIC.

```
6970 REM $SUBROUTINE FOR AUTOWORD WRAP
6980 REM $USES W$ FROM MAIN PROGRAM
6990 REM $USES THE VARIABLES L,Y1,B$(1-4)
7000 Y1=28
7010 L=LEN(W$)
7020 I=1
7030 B$(I)=W$
7040 IF L>Y1 THEN 7070
7050 PRINT B$(I)
7060 RETURN
7070 B$(I+1)=SEG$(B$(I),1,Y1)
7080 B$(I+2)=SEG$(B$(I),Y1+1,L-Y1)
7090 IF SEG$(B$(I+1),Y1,1)=CHR$(32)THEN 7140
7100 B$(I+2)=SEG$(B$(I+1),Y1,1)&B$(I+2)
7110 B$(I+1)=SEG$(B$(I+1),1,Y1-1)
7120 Y1=Y1-1
7130 GOTO 7070
7140 PRINT B$(I+1)
7150 I=I+2
7160 L=LEN(B$(I))
7170 Y1=28
7180 IF L>0 THEN 7040
7190 RETURN
```

Here's a sample main program to illustrate the subroutine.

```
100 REM $SAMPLE MAIN PROGRAM
110 CALL CLEAR
120 W$="THIS IS A TEST.  THIS IS A TEST.  SEE WHAT HAPPENS WITH A VERY LONG WORD SUCH AS
'HIPPOPOTAMUSES.'"
130 PRINT W$::::
140 PRINT "NOW, SEE THE AUTO WORD WRAP:"::
150 GOSUB 7000
160 END
```

AUTOMATIC WORD WRAP ON INPUT

by Roy T. Tamashiro

Webster University

If you found the "Automatic Word Wrap" routine helpful, You might also like to have an auto word wrap feature while the user is typing his input. The subroutine below automatically wraps a word to the next line if the last character on the right edge of the screen is not a space. You must designate the row number using the variable R and the column number (C is a number 1 less than the column number desired) before calling this subroutine. The string input is returned to the main program in the variable ND$. The routine may be used in TI BASIC or TI Extended BASIC.

```
8960 REM $AUTO-WORD WRAP ON INPUT
8970 REM $SEND R=ROW NUMBER;C=COLUMN NUMBER-1 TO SUBROUTINE
8980 REM $THE SUBROUTINE RETURNS INPUT AS ND$
8990 REM $THE SUBROUTINE USES VARIABLES: R,C,K,S,V1,I2,M1,DN
9000 ND$=""
9010 I=R+(C+1)^2
9020 CALL HCHAR(R,C+1,30)
9030 CALL KEY(0,K,S)
9040 IF K<>8 THEN 9140
9050 IF R+(C+1)/2=I THEN 9020
9060 CALL HCHAR(R,C+1,32)
9070 C=C-1
9080 L1=LEN(ND$)-1
9090 ND$=SEG$(ND$,1,L1)
9100 IF C<>1 THEN 9020
9110 C=29
9120 R=R-1
9130 GOTO 9020
9140 IF K<>9 THEN 9160
9150 K=32
9160 IF K<>13 THEN 9190
9170 CALL HCHAR(R,C+1,32)
9180 RETURN
9190 IF (K<13)+(S=0)THEN 9030
9200 CALL HCHAR(R,C+1,K)
9210 ND$=ND$&CHR$(K)
9220 C=C+1
9230 IF C<>30 THEN 9020
9240 DN=0
9250 FOR I2=30 TO 3 STEP -1
9260 IF DN=1 THEN 9380
9270 CALL GCHAR(R,I2,V1)
9280 IF V1=32 THEN 9360
9290 FOR M1=29 TO 3 STEP -1
9300 CALL GCHAR(R+1,M1,CR)
9310 CALL HCHAR(R+1,M1+1,CR)
9320 NEXT M1
9330 CALL HCHAR(R,I2,32)
9340 CALL HCHAR(R+1,3,V1)
9350 GOTO 9380
9360 DN=1
9370 C=32-I2
9380 NEXT I2
9390 R=R+1
9400 GOTO 9020
```

Here is a sample program to test the subroutine.

14

```
100 REM *SAMPLE MAIN PROGRAM*
110 CALL CLEAR
120 R=10
130 C=10
140 GOSUB 9000
150 PRINT ND$
160 END
```

One limitation of this routine is its relatively slow speed. In developing a program you should evaluate whether this slow speed will detract from the activity. If so, you should consider using the built in INPUT and using the "AUTOMATIC WORD WRAP" instructions discussed previously.

THE POWER OF FORTH

by Jim Szydlowski

I received the following program from Charles Nolan in the June issue of Bits and Bytes. It is a fine piece of work by Chris Schram. It is probably the best reason, so far, to have the TI FORTH system up and running. I have yet to find a disk that it will not copy (protected or unprotected).

```
SCR #33
  0 ( half-fast one-drive disk copier -- C.  Schram 4/28/84 )
  1 ( COLD load this screen and DUPLICATE )
  2 BASE-> DECIMAL -SYNONYMS 0 VARIABLE BIG 15358 ALLOT
  3 : ?# EMPTY-BUFFERS 0 BLOCK 10 + @ 256 1024 */MOD SWAP 0= 0= + ;
  4 : PAK CR ." PRESS ANY KEY " 52 GPLLNK KEY DROP CR CR ;
  5 : LMD ." LOAD MASTER DISK" PAK ; : LCD ." LOAD COPY DISK" PAK ;
  6 : DUPLICATE CLS 0 0 GOTOXY LMD 0 DISK_LO ! ?# DUP DUP
  7 DISK_SIZE ! DISK_HI ! LCD
  8 ." ...  FORMATTING COPY DISK ..." 0 FORMAT-DISK
  9 0 DO CLS 0 0 GOTOXY LMD
  10 I 15 0 DO DUP I + DUP .  CR BLOCK BIG I 1024 * + 1024 CMOVE LOOP
  11 CLS 0 0 GOTOXY LCD
  12 15 0 DO DUP I + DUP .  CR BLOCK BIG I 1024 * + SWAP 1024 CMOVE
  13 UPDATE FLUSH LOOP
  14 DROP 15 +LOOP 1 DISK_LO ! ; R->BASE
  15
```

For those of us with only one disk drive, this program will copy a single-sided disk in 6 passes and a double-sided disk in 12 passes. It grabs the data off the master disk and deposits it on the copy disk in 15K gulps.

That's half as efficient as the QUICK-COPYer (TM) program because there are only 17184 bytes of Dictionary/Stack space available in TI FORTH, and this program uses just about all of it. I could have used the available VDP memory or reused the disk buffers, but I wanted to keep it simple. I'm just a beginner. Anyway, unlike QUICK-COPYer, this program won't self-destruct if you mess with it.

To use this program, LOAD FORTH or type COLD it you have already been working in FORTH. Load the "Screen" on which the program is stored. (I used SCR #1) Type DUPLICATE and follow the prompts. When through copying disks, type COLD or FORGET BIG to restore the dictionary space or MON to exit FORTH. NOTE: When copying a double-sided disk, the Copy Disk will have to be previously formatted using the Disk Manager Cartridge.

A word of caution about TI FORTH's operating system. Covering the write-protect notch on a FORTH disk has no effect at all !! That means DON'T USE YOUR MASTER DISK FOR ANY REASON EXCEPT TO MAKE A BACKUP OF ITSELF!!!!! If you add a new "screen" to a FORTH disk that contains the Kernal program, the Kernal program will be written over! That means use a separate disk for SCREENS!

COMPUTER PROGRAMS WHICH GIVE BIRTH TO NEW COMPUTER PROGRAMS

Roy T. Tamashiro, Ed.D.

Even with all the wonderful things computers can do, I have often had this one wish: I wish the computer could write programs for me. Creating programs takes a lot of time, and it seems that the computer itself ought to be able to speed up this process. At first glance this idea seems exotic, if not impossible, because this would mean that the computer will do the thinking which is ordinarily done by the hBuBmBaBnB programmer. A computer program which writes other computer programs would be something that literally replaces a programmer! A radical thought!

Actually, it is really possible for programs to create new programs. It is even possible to do this on small computers. The program listed below called NUMBER MOTHER is a demonstration of this idea. This program will not put programmers out of work, but it does show how a program can generate new, free-standing programs. NUMBER MOTHER will create various simple quizzes based upon a mystery formula which you specify in the MOTHER program. Each time you use NUMBER MOTHER, you can enter a different mystery formula, making the "OFFSPRING" program different than the previous one.

The Extended BASIC cartridge and a Disk Drive system are required for using NUMBER MOTHER. Follow the steps below to use the program:

1. Type in and SAVE the program listing below for NUMBER MOTHER.

2. RUN the program. The program will ask you to select what you want the mystery formula to be in the offspring program. The formula must include an arithmetic operation (add, subtract, multiply or divide) and a factor (any number).

3. Then NUMBER MOTHER will ask you for a filename for the offspring program. You can enter any valid name (except of course the name you have given the NUMBER MOTHER program on your disk). For example, when you see the prompt, "Enter a filename for the FORMULA GUESS program," enter:

```
DSK1.CHILD
```

4. When you get the READY message, clear the computer's memory by typing NEW. (You may also type BYE or use <QUIT> (FCTN =) to return to the title screen, then reselect Extended BASIC.)

5. When the memory is clear, use the MERGE command to load the OFFSPRING program. In our example,type:

```
MERGE DSK1.CHILD
```

Do not use the usual OLD to load the offspring program the first time.

6. After completing the MERGE command, you may RUN the program, LIST the program or EDIT lines in the offspring program as you normally do with other BASIC programs. When the offspring program is RUN, the user is asked to enter a 'starting number.' Then 10 numbers will appear. These numbers change according to the operation and the factor you entered in the MOTHER program. The user tries to guess this original formula.

. Optionally, You may use the SAVE command to record the offspring program on cassette or disk. Use the OLD command to load the cassette or disk program which you have recorded using this SAVE command.

Notice that the offspring program is independent of the MOTHER program. Thus you can have NUMBER MOTHER on one disk and the offspring programs on separate disks, without ever needing NUMBER MOTHER to run the offspring programs. Note also that, in this example, the offspring program will RUN in either BASIC or Extended BASIC, if you completed step 7.

```
100 '#####################
110 !# THE NUMBER MOTHER #
120 !#####################
130 !AUTHOR: ROY TAMASHIRO.ED.D.
140 'FEBRUARY 1985.X-BASIC WITH DISK SYSTEM
150 'SEE INSTRUCTIONS IN 'THE COMPUTER BRIDGE',FEB. 1985
160 GOSUB 480 :: DISPLAY AT(3,1):"Which operation do you want in the quiz?"
170 DISPLAY AT(6,3):"1 Addition" :: DISPLAY AT(7,3):"2 Subtraction"
180 DISPLAY AT(8,3):"3 Multiplication" :: DISPLAY AT(9,3):"4 Division" :: DISPLAY AT(10,5):"Your Choice(1-4):"
190 ACCEPT AT(10,23)SIZE(1)VALIDATE("1234"):O$ :: O=VAL(O$):: ON O GOSUB 490,500 ,510,520
200 GOSUB 480 :: DISPLAY AT(3,1):"Enter a factor (a number):"
210 DISPLAY AT(4,1):O$ :: ACCEPT AT(4,LEN(O$)+2)VALIDATE(NUMERIC):F :: FA$=STR$(F)
220 GOSUB 480 :: DISPLAY AT(3,1):"Mystery Factor in FORMULA   GUESS will be: ", O$;F
230  DISPLAY  AT(7,1):"Enter  a  filename  for  the   FORMULA  GUESS  program."  ::  DISPLAY  AT(9,1):"DSK1."  ::  ACCEPT
AT(9,4)SIZE(-10):F$
240 OPEN #1:"DSK"&F$.DISPLAY .VARIABLE 163
250 GOSUB 480 :: DISPLAY AT(3,1):"FORMULA GUESS being created on DSK";F$
260 READ D :: IF D=999 THEN 460 ELSE GOSUB 380
270 IF D<>141 THEN 370
280 PRINT #1:CHR$(83);CHR$(190);CHR$(83);:: ON O GOSUB 330,340,350,360
290 PRINT #1:CHR$(200);CHR$(LEN(FA$));FA$;:: GOSUB 450
300 GOSUB 380 :: PRINT #1:CHR$(79);CHR$(36);CHR$(190);CHR$(199);CHR$(LEN(O$)); O$;:: GOSUB 450
310 GOSUB 380 :: PRINT #1:CHR$(70);CHR$(65);CHR$(36);CHR$(190);CHR$(199); CHR$(LEN(FA$));FA$;:: GOSUB 450
320 GOSUB 380 :: PRINT #1:CHR$(79);CHR$(190);CHR$(200);CHR$(1);STR$(D);:: GOSUB 450 :: GOTO 260
330 PRINT #1:CHR$(193);:: RETURN
340 PRINT #1:CHR$(194);:: RETURN
350 PRINT #1:CHR$(195);:: RETURN
360 PRINT #1:CHR$(196);:: RETURN
370 READ G :: ON G GOSUB 390,390,390,430 :: GOSUB 450 :: GOTO 260
380 N=N+1 :: PRINT #1:CHR$(1);CHR$(N);CHR$(D);:: RETURN
390 READ E$ :: ON G GOSUB 400,410,420 :: RETURN
400 PRINT #1:CHR$(200);CHR$(LEN(E$));E$;:: RETURN
410 PRINT #1:CHR$(199);CHR$(LEN(E$));E$;:: RETURN
420 READ H$ :: GOSUB 410 :: PRINT #1:CHR$(181);H$;:: RETURN
430 READ K :: IF K<>0 THEN PRINT #1:CHR$(K);:: GOTO 430
440 GOSUB 450 :: GOTO 260
450 PRINT #1:CHR$(0):: RETURN
460 PRINT #1:CHR$(255);CHR$(255):: CLOSE #1 :: CALL CLEAR
470 PRINT "Type NEW and <ENTER>, then Type MERGE DSK";F$;:"then RUN the program." :: END
480 CALL CLEAR :: DISPLAY AT(1,6):"THE NUMBER MOTHER" :: CALL HCHAR(2,8,45,17):: RETURN
490 O$="ADD" :: RETURN
500 O$="SUBTRACT" :: RETURN
510 O$="MULTIPLY BY" :: RETURN
520 O$="DIVIDE BY" :: RETURN
530 DATA 154,2,#################,154,2,# FORMULA GUESS #,154,2, #################
540                           DATA                              157,1.CLEAR,156,2,##############################,156,2,"#      FORMULA
UESS     #",156,2,"#                    #"
550 DATA 156.2,"#   A Program created by   #".156,2,"#   'THE NUMBER MOTHER'    #",156.2,"##############################"
560 DATA 156,4,0,156.4.0,156,4,0,156,4,0,156,2
570 DATA "After you give a 'starting  number', you will see 10    numbers. Try to guess the"
580 DATA 156.2."operation and factor which  account for the changes in  the 10 numbers.".156,4,0
590 DATA 146,3,"Enter your starting number: ",S
600 DATA 83,4,36,190,199,17,83,116,97,114,116.105,110,103,32,78,117
610 DATA 109,98,101,114,58,32,184,219,183,83,182,0
620 DATA 157.1,CLEAR,156.4,83,36,0,156,4,0,140,4,73,190,200,1,49.177,200,2,49,48 ,0
630 DATA 156,4,83,0,141,150,4,73,0,156,4,0
640 DATA 146.3,"What's the operaton?(Enter 1=Add;2=Subtract;3=Multiply; 4=Divide): ","O1"
650 DATA 156,4,0.146.3,"What's the factor?(Enter a number): ","F1$",156,4,0
660 DATA 132,4,183,79,49,190.79,182,193.183.70.65.36
670 DATA 190,70,49,36,182,190,194,200,1,50.176,201,1,38,0
680 DATA 156,2,"INCORRECT FORMULA.",134,4,201,1,39,0
690 DATA 156,2,"CORRECT!",146,3,"Enter: 1-To do again; 2-To see the formula; 3-To exit ","C"
700 DATA 157.1,CLEAR.155,4,67.134,201.1,18,179.201,1,42,179,201,1,45,0
710 DATA 156,2,"The formula is:",156,4,79,36,180,199,1,32,180,0.156.4,70,65,36,0 .139.4.0.999
```

THE BASIC NOTE PAD

Dr. Roy T. Tamashiro

Sometimes you need a word processor, but you do not have the necessary hardware and software handy. For example, it is inconvenient to bring the expansion system, disk drive(s), cartridge and disks to your office or to the library, where you could use the word processor to jot down some important information. Or, you may just want to show the neighbors or relatives what they can do with the TI-99/4a console which is gathering dust in the closet. The program listed below is a small word processor which will operate on just the minimal TI-system: console, cassette and monitor or TV set. No modules are needed. The program will also operate with an expanded system or with Extended BASIC. You may also convert the files you create on BASIC NOTE PAD to files you can use on a full-fledged word processor like TI-WRITER.

After you type the program and SAVE it on your cassette or disk, RUN the program. You will see a list of options. When you press 1 (to write/edit), you will see a blank screen with the cursor at the upper left corner. You may begin typing your material. On this word processor, you have almost the full screen (28 columns and 23 rows) on which to put your material. You may use the following keys for editing:

[UP-ARROW] (FCTN-E)	Moves the cursor up one line.
[DOWN-ARROW] (FCTN-X)	Moves the cursor down one line.
[LEFT-ARROW] (FCTN-S)	Moves the cursor left one character without deleting a character.
[RIGHT-ARROW] (FCTN-D)	Moves the cursor right one character without deleting a character.
[ENTER]	Moves the cursor to the leftmost position of the next row.
[DEL] (FCTN-1)	Deletes a character on a line.
[INS] (FCTN-2)	Inserts a space on a line. (All characters move over 1 space.)
[ERASE] (FCTN-3)	Aborts write/edit selection.
[PROC'D] (FCTN-6)	Accepts the completed page.

When you finish typing your screen, press [PROC'D] or (FCTN-6). There is a pause to prepare the page for saving to cassette or disk. Note that if you press [ERASE] (which is (FCTN-3)), your material has not been prepared for saving or printing. You may choose option 4 (PURGE) to remove what you typed permanently, or press 1 (WRITE/EDIT) again to get back the page.

When you get the prompt "FILENAME to SAVE or press [ENTER] to cancel.", type **CS1** if you are saving your material on tape, or a disk filename such as **DSK1.MY-NOTE** if you are saving to a disk. You may retrieve this page later by selecting33 (LOAD) and typing **CS1** for cassette or the disk filename.

Selection 5 (SAVE TO DIS/VAR 80) allows you to convert your material into the disk format readable by the TI-WRITER word processor or the EDITOR/ASSEMBER editor. When prompted, type a valid disk filename such as DSK1.NOTE-2. When you want to use the pages in TI-WRITER, enter TI-WRITER's editor and use the LF command to load the page. If you wrote and saved several pages on the BASIC NOTE PAD, you can load all of them by typing the line number where you want the pages loaded as you type the filename. For example, if you type LF [ENTER], then 24 DSK1.PAGE-2 [ENTER] the file PAGE-2 will be added to your editor after line 24.

To PRINT your page on a printer, select 6 (PRINT) from the main selection screen. Type your printer access name such as PIO or **RS232.BA=600**.

Since this word processor can only handle one screen at a time, each screen must be saved individually. On cassette, this means recording the screens one after the next on the tape (do NOT rewind the tape even though the screen tells you so). On disk systems, this means that every screen should have a different filename.

(continued)

```
100 REM *******************
110 REM * BASIC NOTE PAD *
120 REM *******************
130 REM AUTHOR-ROY TAMASHIRO
,ED.D
140 REM 1986 THE COMPUTER BR
IDGE
150 DIM R$(23)
160 CALL CLEAR
170 PRINT TAB(8);"BASIC NOTE
 PAD": :TAB(7);"Roy T. Tamas
hiro": :
180 CALL HCHAR(21,10,45,14)
190 PRINT TAB(3);"1 - WRITE/
EDIT":TAB(3);"2 - SAVE"
200 PRINT TAB(3);"3 - LOAD":
TAB(3);"4 - PURGE":TAB(3);"5
 - SAVE AS DIS/VAR 80"
210 PRINT TAB(3);"6 - PRINT"
:TAB(3);"7 - EXIT": :
220 INPUT "YOUR CHOICE(1-7):
 ":CH$
230 IF (CH$<"1")+(CH$>"7")TH
EN 160
240 CALL CLEAR
250 ON VAL(CH$)GOTO 300,960,
1070,260,1190,1190,1280
260 FOR R=1 TO 23
270 R$(R)=""
280 NEXT R
290 GOTO 160
300 FOR R=1 TO 23
310 PRINT R$(R)
320 NEXT R
330 PRINT "[PROC'D]=done;[ER
ASE]=abort.";
340 FOR R=1 TO 23
350 FOR C=1 TO 28
360 CALL GCHAR(R,C+2,G)
370 CALL HCHAR(R,C+2,30)
380 CALL KEY(5,Y,U)
390 IF U=0 THEN 380
400 IF Y>31 THEN 810
410 IF Y>13 THEN 380
420 ON Y GOTO 360,360,430,49
0,360,360,160,570,570,570,57
0,570,570
430 FOR RS=C TO 27
440 CALL GCHAR(R,RS+3,G)
450 CALL HCHAR(R,RS+2,G)
460 NEXT RS
470 CALL HCHAR(R,30,32)
480 GOTO 700
490 FOR RS=28 TO C+2 STEP -1
500 CALL GCHAR(R,RS+1,G2)
510 CALL HCHAR(R,RS+2,G2)
520 NEXT RS
530 CALL HCHAR(R,C+3,G)
540 CALL HCHAR(R,C+2,32)
550 CALL HCHAR(R,31,32,2)
560 GOTO 700
570 CALL HCHAR(R,C+2,G)
580 IF Y<>8 THEN 600
590 C=C-1
600 IF Y<>9 THEN 620
610 C=C+1
620 IF Y<>10 THEN 640
630 R=R+1
640 IF Y<>11 THEN 660
650 R=R-1
660 IF Y<>12 THEN 700
670 C=29
680 R=24
690 GOTO 850
700 IF (Y=13)+(C=29))-1 THEN
 730
710 R=R+1
720 C=1
730 IF R<>24 THEN 750
740 R=1
750 IF C>=1 THEN 780
760 R=R-1
770 C=28
780 IF R>=1 THEN 800
790 R=23
800 GOTO 360
810 IF Y>126 THEN 380
820 CALL HCHAR(R,C+2,Y)
830 NEXT C
840 NEXT R
850 IF Y<>12 THEN 340
860 FOR R=1 TO 23
870 R$(R)=""
880 NEXT R
890 FOR R=1 TO 23
900 FOR C=3 TO 30
910 CALL GCHAR(R,C,G)
920 R$(R)=R$(R)&CHR$(G)
930 NEXT C
940 NEXT R
950 CALL CLEAR
960 INPUT "FILENAME to SAVE,
 or press  [ENTER] to cancel
970 IF F$="" THEN 160
980 OPEN #1:F$,OUTPUT,INTERN
AL,FIXED 192
990 FOR R=1 TO 23
1000 IF (R/6=INT(R/6))+(R
)<=-1 THEN 1030
1010 PRINT #1:R$(R),
1020 GOTO 1040
1030 PRINT #1:R$(R)
1040 NEXT R
1050 CLOSE #1
1060 GOTO 160
1070 EF=0
1080 INPUT "FILENAME to LOA
 or press   [ENTER] to canc
l.: ":F$
1090 IF F$="" THEN 160
1100 OPEN #1:F$,INPUT ,INTER
NAL,FIXED 192
1110 FOR R=1 TO 23
1120 IF (R/6=INT(R/6))+(R=23
)<=-1 THEN 1150
1130 INPUT #1:R$(R),
1140 GOTO 1160
1150 INPUT #1:R$(R)
1160 NEXT R
1170 CLOSE #1
1180 GOTO 160
1190 PRINT "DEVICE or FILE N
AME, or":"press [ENTER] to c
ancel."
1200 INPUT "":F$
1210 IF F$="" THEN 160
1220 OPEN #1:F$
1230 FOR R=1 TO 23
1240 PRINT #1:R$(R)
1250 NEXT R
1260 CLOSE #1
1270 GOTO 160
1280 END
```

THE TI-WRITER COMPANION.

DR. BILL BROWNING
7541 JERSEY AVE.
NORTH,BROOKLYN PARK,MN.
55428
$6.50

For that price, you get 29, 8 1/2 by 11 pages, singled spaced, containing suggestions such as; How to make the most of both 40 column and 80 column capabilites of the TI-WRITER ... How to turn a black and white TV into a green-screen unit ... How to make your right-justified material more professional looking ... How to work with two columns ... how to estimate word counts ... How to type on your printer directly from the keyboard ... and much more.
Written by a professional writer with many years of writing experience, you are assured of clear, concise material. And if that is not enough folks, it comes with a MONEY BACK GUARANTEE. Sure can't beat the price.
FROM THE SUNCOAST BEEPER, ST. PETERSBURG, FL.

"XB WEE BOMBER"
TOPICS - LA 99ERS

```
100 CALL CLEAR::CALL HCHAR(2
3,2,30,30)::CALL COLOR(9,4,4
)::CALL HCHAR(24,1,96,32)::C
ALL SPRITE(#1,43,2,35,256,0,
-10)
110 CALL KEY(0,K,S)::CALL SO
UND(-1000,-3,0,200,10)::IF S
=0 THEN 110::CALL POSITION(#
1,X,Y)::A=X::CALL SPRITE(#2,
46,2,X,Y)
120 A=A+4::CALL LOCATE(#2,A,
Y)::IF A>=21*8 THEN CALL SOU
ND(100,-7,0)::CALL POSITION(
#2,R,S)::CALL HCHAR(23,S/C+1
,32)::A=0::CALL DELSPRITE(#2
)::GOTO 110
130 CALL SOUND(-100,-3,0,200
0-(4*A),10)::GOTO 120
```

PRINTER SHORTCUTS

Dr. Roy T. Tamashiro

QUESTION: "Is there a simple way to send commands to the printer when I am using TI-WRITER?"

ANSWER: Several previously published articles describe how to send commands to the printer (for example, see Castleton, 1985; Hoyt, 1986). Although they work smoothly, each method has limitations: In Castleton's method, you lose the capability of printing some characters, and in Hoyt's method, you must run the separate BASIC program before you load TI-WRITER, and you cannot send new printer commands in the middle of your document. The method described here uses the Editor in TI-WRITER, so you can include the printer commands as you are preparing your document.

You will need to know your printer's command codes which you wish to use. This is found in the appendix of your printer manual. The following are codes used in Epson MX-80, TI-99/4 and other common dot matrix printers: (Check your printer manual for these and other codes.)

(1)To Print:	(2)Code:	(3)Press these keys:	(4)Screen Shows:
Enlarged On	14	[CTRL-U] N	'e
Condensed On	15	[CTRL-U] O	'r
Condensed Off	18	[CTRL-U] R	'z
Enlarged Off	20	[CTRL-U] T	'q
Emphasized On	27;69	[CTRL-U] [FCTN-R] [CTRL-U] E	'b E
Emphasized Off	27;70	[CTRL-U] [FCTN-R] [CTRL-U] F	'bF
Double Strike On	27;71	[CTRL-U] [FCTN-R] [CTRL-U] G	'b G
Double Strike Off	27;72	[CTRL-U] [FCTN-R] [CTRL-U] H	'bH

Select the command you want from column (1). Type the characters in Column (3). [CTRL-U] means to press the [CTRL] key and U at the same time. This puts you in the special character mode which allows you to type ASCII codes 0-31. The cursor changes from a flashing rectangle to a flashing underline. The characters you type next will appear as special symbols indicate on page 146 of the TI-WRITER manual. Press [CTRL-U] again to get back the regular cursor. After you type column (3), you will see the characters in column (4) on your screen.

Here is an example. Type the following in TI-WRITER's Editor:	On the screen, the text will appear as follows:
[CTRL-U] [FCTN-R] [CTRL-U] E Jack be nimble,	'bE Jack be nimble,
[CTRL-U] [FCTN-R] [CTRL-U] F Jack be quick,	'bF Jack be quick,
[CTRL-U] O [CTRL-U] Jack jump over the	'r Jack jump over the
[CTRL-U] RN [CTRL-U] candlestick. [CTRL-U] T [CTRL-U]	'z'ecandlestick.'q

Then print the file, using the **PF** command. You will see the following:

Jack be nimble, <-----------EMPHASIZED TYPE ON.
Jack be quick, <-----------EMPHASIZED TYPE OFF.
Jack jump over the <-----------CONDENSED TYPE ON.
c a n d l e s t i c k . <--ENLARGED TYPE.

REFERENCES: Castleton, R., "Taking control or TI-Writer", MICROpendium, 1985, February, pp. 12-15
Hoyt, H., "Printer setup program", The Computer Bridge, 1986, Vol. 5, No. 4 (April), 3-4

A Swift Communication Link with Your "Other" Computer

Dr. Roy T. Tamashiro

If, like me, you work with various computers besides the TI-99/4a, you often wish there was a speedy way to transfer data between the computers. A modem is not always adequate, because it is slow--300 baud or 1200 baud seems to take forever. Also, you do not always have two telephones and separate lines to connect with the computers. The solution to these problems is to connect the two computers directly. By linking your computers together via a cable, you may transfer information as fast as 9600 baud. That is about 32 times faster than with a 300 baud modem!

The direct connect cable is possible with any computer which has a regular serial communications port. (There are other possible methods of linking the computers, though this is probably the simplest.) I connected my TI-99/4a with an Apple IIe equipped with a Super Serial Card (This description also applies to Apple II and Apple II+ computers.). The TI-99/4a requires an RS-232 card.

The cable that connected the Apple and TI used standard EIA RS232C 25-pin male connectors on both ends. The pin wirings were as follows:

RS232 ---------- Super Serial Card
1 ---------------------- 1
2 ---------------------- 2
3 ---------------------- 3
6 ---------------------- 20
7 ---------------------- 7
20 ---------------------- 6

The switches on the Super Serial Card should be set according to the software you are using for data transfer. For my purposes, I set the following:

SW1-1	SW1-2	SW1-3	SW1-4	SW1-5	SW1-6	SW1-7
OFF	OFF	OFF	ON	ON	ON	ON

SW2-1	SW2-2	SW2-3	SW2-4	SW2-5	SW2-6	SW2-7
ON	ON	ON	OFF	OFF	OFF	OFF

These settings result in the following configuration: Baud Rate=9600, Data Format=8 data bits/one stop bit/odd parity. Use the Super Serial Card manual if you want a different configuration.

Transmitting data between the two computers can be done on the TI via BASIC, Extended-BASIC, TI-Writer, Editor/Assembler, or any software which allows you to SAVE and LOAD to disk or PRINT to a printer. On the Apple, the software may be in BASIC, or communications software which is compatible with the Super Serial Card. For example, to send a TI-Writer file, load your file into the editor, select SAVE (SF) or PRINT (PF), and type RS232.BA=9600.DA=8.CR if you set the switches as noted above. To receive information from Apple, select LOAD file (LF), and use RS232.BA=9600.DA=8 as the filename.

Although only the connection between TI-99/4a and Apple IIe is discussed here, these principles can be applied to any two computers which have a serial communications port to make swift, convienient and accurate transfer of information possible.

```
************************************
*        THE MESSAGE ENCODER       *
*        Dr. Roy T. Tamashiro      *
************************************
```

Here is a paragraph describing a well-known historical figure. Notice that there are extra letters throughout the paragraph. If you find all the extra letters, they will spell the name of the mystery person:

> This famous scientistG was born near Diamoend, Missouri in 1864o. He received his degrrees in agriculture fgrom Iowa State Univeersity. He developed mWany new soil conservaation methods and wasys to improve crop prohduction. He is best iknown for his work wnith peanuts. He devegloped over 300 productts from peanuts, inclouding soap, a milk snubstitute, face powderC and printer's ink. aIn 1939, he was awarrded the Theodore Roosevvelt Medal for outsteanding contributionsr in science. A national monument in his name was established in 1951 on the Missouri farm where he was born.

In this next paragraph the extra letters spell a message. See if you can figure out this message by another famous American:

> Few people toLday are as tialented in ats many diffetrent areas als was the autheor of this msessage. He its known for rhis inventioons like the hekating stove eand he provesd that lightfning is elecetricity. He lwas a signer olf the Declarbation of Indiependence asg well as theo U.S. Constituation. He pubklished the SsAYINGS OF POOR RICHARD, where the hidden message first appeared. Of course, this man is Benjamin Franklin.

Puzzles such as these are valuable in strengthening a person's reading abilities. To solve the puzzle, one must read for meaning, know the correct spelling of the words used in the passage, and use context clues to find the letters which spell the message one is seeking.

The TI-99 4/A computer equipped with TI-Writer word processor and the Extended BASIC program listed below will allow you to build a puzzle such as those above. Here's how it is done: Start by loading TI-Writer's Editor. Press T for Tabs and use the right arrow until you are at the 70 column marker. Press R for Right Margin, and press [Enter].

Now, type the paragraph in which you wish to embed the coded message. Check to see that everything is spelled correctly. When you finish typing the paragraph, press Command/Escape [FCTN-9] and type PF for PrintFile. Type a valid disk name, such as DSK1.MY-STORY and press [Enter]. (Note that PF (PrintFile) is used instead of SF (SaveFile) so the format codes are not recorded on the disk file.)

Type the Extended BASIC program "Message Encoder" listed below, and save it on your disk. RUN the program. When you get the first prompt, "TYPE FILENAME:", enter the name of the file you used to save your paragraph. (For example: DSK1.MY-STORY) Then type the message to be embedded as directed. After the program does its work, you may choose to print the paragraph with the embedded codes, or save it disk for later work with your word processor.

(The solutions to passages are spelled backwards here: PASSAGE 1: revraC notgnihsaW egroeG PASSAGE 2: skao gib llef sekorts elttiL)

```
100 ! *******************
110 ! * MESSAGE ENCODER *
120 ! *******************
130 !AUTHOR:ROY TAMASHIRO
140 !SEPTEMBER 1986
150 DIM L$(255)
160 GOSUB 420 :: DISPLAY AT(
3,3):"This program assumes t
hat":"a passage has been cre
ated":"using TI-Writer with
the"
170 DISPLAY AT(6,1):"right m
argin set at less":"than 70.
"
180 DISPLAY AT(9,1):"TYPE FI
LENAME: DSK1." :: ACCEPT AT(
9,19)SIZE(-10):F$
190 DISPLAY AT(11,1):"ENTER
MESSAGE:" :: ACCEPT AT(12,1)
:M$
200 DISPLAY AT(14,1):"ARE TH
E ENTRIES CORRECT?":"(Y/N):"
 :: ACCEPT AT(15,7)VALIDATE(
"YN")SIZE(1):OK$ :: IF OK$<>
"Y" THEN 160
210 DISPLAY AT(24,1):"Workin
g. Please wait..." :: OPEN #
1:"DSK"&F$
220 N=I :: I=I+1 :: IF EOF(1
)=1 THEN 240
230 LINPUT #1:L$(I):: GOTO 2
20
240 CLOSE #1
250 FOR I=1 TO N :: FOR J=1
TO LEN(L$(I)):: IF SEG$(L$(I
),J,1)<>CHR$(32)THEN C=C+1
260 NEXT J :: NEXT I
270 FOR I=1 TO LEN(M$):: AA$
=SEG$(M$,I,1):: IF AA$<>CHR$
(32)THEN C$=C$&AA$
280 NEXT I
290 FA=INT(C/LEN(C$)):: IF L
EN(C$)>=C THEN DISPLAY AT(16
,1):"The passage is too shor
t.":"Retype the passage." ::
 END
300 IF C$="" THEN 340
310 IF FA>LEN(E$)THEN I=FA-L
EN(E$):: S=S+1 :: IF S>N THE
N 340
320 E$=SEG$(L$(S),I+1,LEN(L$
(S))-I):: L$(S)=SEG$(L$(S),1
,I)&SEG$(C$,1,1)&E$
330 C$=SEG$(C$,2,LEN(C$)-1):
: I=FA+I :: GOTO 300
340 GOSUB 420 :: DISPLAY AT(
3,1):"PRESS:":"1 TO PRINT":"
2 TO SAVE ON DISK":"3 TO RES
TART":"4 TO END"
350 DISPLAY AT(10,1):"YOUR C
HOICE (1-4):" :: ACCEPT AT(1
0,19)SIZE(1)VALIDATE("1234")
:C$ :: C=VAL(C$)
360 ON C GOTO 370,380,400,41
0
370 DISPLAY AT(12,1):"TYPE D
EVICENAME:" :: ACCEPT AT(13,
1):F$ :: GOTO 390
380 DISPLAY AT(12,1):"TYPE F
ILENAME: DSK1." :: ACCEPT AT
(12,19)SIZE(-10):F$ :: F$="D
SK"&F$
390 OPEN #1:F$ :: FOR I=1 TO
 N :: PRINT #1:L$(I):: NEXT
I :: CLOSE #1 :: GOTO 340
400 RUN
410 CALL CLEAR :: END
420 CALL CLEAR :: DISPLAY AT
(1,7):"MESSAGE ENCODER" :: C
ALL HCHAR(2,9,45,15):: RETUR
N
```

THE COMPUTER BRIDGE
P.O.BOX 63156 TOWER GROVE STA.
ST. LOUIS MO 63163

QUICK LABEL

Dr. Roy T. Tamashiro

Word processors are wonderful for preparing letters but they are usually not set up do the envelopes or labels for mailing. The program listed below (QUICK LABEL), which runs on TI-BASIC or TI-Extended BASIC, will prepare an envelope or address label.

Type in the program below and save it on cassette or disk. When you RUN the program, you are asked for the PRINTER NAME. Normally this is PIO or RS232(with extensions). Then select "Normal" or "Elite" characters. Select normal for most cases. If you select "elite" you must know the "printer codes" to enter at the next prompt. Press [Enter] if there are no printer codes needed. Then select envelope size: Business (9.5 X 4 inches), Small (6.5 X 3.5 inches), or none.

You are then prompted to type a label of up to six lines. Press [Enter] after each line. You may start over at any time by pressing ERASE (FCTN-3). After this, you may correct or print the label. Select the location from the choices given. Place the envelope in the printer as far to the left as possible, with the top edge of the envelope just above the print head. The program automatically calculates where to print according to the envelope size and character type you chose.

```
100 REM*****************
110 REM* QUICK LABEL *
120 REM*****************
130 REM AUTHOR-ROY TAMASHIRO
,ED.D
140 REM THE COMPUTER BRIDGE,
OCT 1986
150 GOSUB 1030
160 PRINT "Type PRINTER NAME
,"
170 INPUT "such as PIO or RS
232.BA=9600":P$
180 PRINT :"Characater size:
":" 1=Normal, 2=Elite":"YOUR
 CHOICE:"
190 CALL KEY(5,K,S)
200 IF (S=0)+(K<49)+(K>50)TH
EN 190
210 PRINT CHR$(K)
220 CLIN=40*(K-47)
230 PRINT :"Enter PRINTER CO
DES, or ":"press [Enter] if
none."
240 INPUT "(CTRL Keys appear
 as spaces)":CODE$
250 IF CODE$="" THEN 300
260 FOR I=1 TO LEN(CODE$)
270 C=ASC(SEG$(CODE$,I,1))
280 C2$=C2$&CHR$(C*ABS(C<129
)+(C-128)*ABS(C>128))
290 NEXT I
300 OPEN #1:P$
310 PRINT #1:C2$
320 PRINT :"Envelope Size:":
" 1=Business, 2=Small, 3=Non
e":"YOUR CHOICE:"
330 GOSUB 1000
340 IF (K<49)+(K>51)THEN 330
350 IF K<>50 THEN 370
360 E=.25*(CLIN/2)
370 GOSUB 1030
380 PRINT : : : : : : : : :
: : : : : : : :
390 CALL HCHAR(4,3,45,28)
400 CALL HCHAR(5,1,32,192)
410 CALL HCHAR(11,3,45,28)
420 W$="12Type your label ab
ove."
430 GOSUB 940
440 W$="13[Enter]=next, [ERA
SE]=abort."
450 GOSUB 940
460 FOR R=5 TO 10
47Q FOR C=3 TO 30
480 CALL GCHAR(R,C,G)
490 CALL HCHAR(R,C,30)
500 CALL KEY(5,Y,U)
510 IF (U=0)+(Y>126)THEN 500
520 IF Y>31 THEN 620
530 CALL HCHAR(R,C,G)
540 IF Y=7 THEN 390
550 IF Y=13 THEN 640
560 IF Y<>8 THEN 480
570 CALL HCHAR(R,C,32)
580 C=C-1
590 IF C>2 THEN 480
600 C=3
610 GOTO 480
620 CALL HCHAR(R,C,Y)
630 NEXT C
640 R$(R-4)=""
650 FOR B=3 TO C-1
660 CALL GCHAR(R,B,G)
670 R$(R-4)=R$(R-4)&CHR$(G)
680 NEXT B
690 NEXT R
700 RESTORE 1120
710 N=7
720 GOSUB 1060
730 IF (K<49)+(K>54)THEN 700
740 L=K-48
750 SPIN=0
760 IF L<4 THEN 780
770 SPIN=CLIN/2-14-E
780 ON L GOTO 390,910,790,79
0,790
790 N=2
800 GOSUB 1060
810 IF K=88 THEN 910
820 IF K<>13 THEN 790
830 ON L-2 GOTO 870,870,840
840 FOR I=1 TO 8
850 PRINT #1:
860 NEXT I
870 FOR I=1 TO 6
880 PRINT #1:TAB(SPIN);R$(I)
890 NEXT I
900 GOTO 700
910 CALL CLEAR
920 CLOSE #1
930 END
940 R=VAL(SEG$(W$,1,2))
950 CALL HCHAR(R,1,32,224)
960 FOR N=3 TO LEN(W$)
970 CALL HCHAR(R,N,ASC(SEG$(
W$,N,1)))
980 NEXT N
990 RETURN
1000 CALL KEY(3,K,S)
1010 IF S=0 THEN 1000
1020 RETURN
1030 CALL CLEAR
1040 PRINT TAB(9);"QUICK LAB
EL":TAB(9);"-----------":TAB
(6);"Roy T. Tamashiro": : :
:
1050 RETURN
1060 FOR K=1 TO N
1070 READ W$
1080 GOSUB 940
1090 NEXT K
1100 GOSUB 1000
1110 RETURN
1120 DATA "12What Now?","13
[1]Fix label, [2]Exit, or","
14Print the label at:","15 [
3] Top Left Corner"
1130 DATA "16 [4] Top Center
","17 [5] Envelope center.",
"18YOUR CHOICE (1-5): _"1140
 DATA "12Insert envelope or
label,","13Press [Enter], or
 [X]=Exit."
```

WHIRLS AND FLIPS: UPSIDE DOWN TYPING

Dr. Roy T. Tamashiro

TERRY: "How can I get text to be upside down on my computer monitor?"
ANDY: "Do a hand stand."
TERRY: "Is there another way?"
ANDY: "Sure, Turn the monitor over."
TERRY: "Yes, but that would look foolish!"

It is not clear why one would want things to appear upside down on the screen, except for fun. But "fun" is a good enough reason. So here is a routine that will allow you to turn the text on the screen upside down (and other dazzling tricks) without doing hand stands or turning over the monitor. Type the program below in Extended BASIC and SAVE it on cassette tape or disk, then RUN the program. Note that the 32-K Expansion Memory is required, and no other assembly language program can be used at the same time.

```
100 !********
110 !* FLIP *
120 !********
130 !1986,R.TAMASHIRO
140 !X-BASIC,MEMORY EXPANSIO
N REQUIRED
150 !RUN THIS FIRST.
160 CALL INIT :: CALL LOAD(1
6376,70,76,73,80,32,32,36,24
4):: CALL LOAD(8196,63,248)
170 CALL LOAD(9460,2,0,4,0,2
,5,37,72,2,2,0,2,2,4,0,4,2,1
,40,56,4,32,32,44)
180 CALL LOAD(9484,192,96,40
,56,6,5,213,65,6,193,6,5,213
,65,5,192,2,128,6,240,21,5,6
,4)
190 CALL LOAD(9508,22,239,2,
37,0,16,16,234,2,2,2,240,2,1
,37,64,2,0,4,0,4,32,32,36,4,
96,0,112)
200 Q$=CHR$(34):: CALL CLEAR
 :: PRINT Q$;"FLIP";Q$;" rou
tine is loaded.":"To use, ty
pe:":" CALL LINK(";Q$;"FLIP"
;Q$;")"
```

Now, whenever you use CALL LINK("FLIP"), the characters on the screen will turn upside down. Try the following program to see how this works:

```
100 CALL CLEAR
110 PRINT "FLIP DEMO": : :
120 FOR I=33 TO 126 :: PRINT
 CHR$(I); :: NEXT I :: PRINT
130 CALL LINK("FLIP")
140 INPUT "PRESS [ENTER]":R$
::GOTO 100
```

Press [FCTN]-4 (Clear) to exit the program. Here is a routine that allows you to print a whole screenful upside down before it is turned right-side up. Type in, SAVE (to cassette or disk), then RUN the program.

```
100 !**********************
110 !* UPSIDE-DOWN DEMO *
120 !**********************
130 DATA 1,"UPSIDE DOWN"
140 DATA 2,"-----------"
150 DATA 3,"by Roy Tamashiro
"
160 DATA 5,"In this routine,
 the text"
170 DATA 6,"first appears up
side down."
180 DATA 7,"The letters are
bottoms-up"
190 DATA 8,"and the top line
 is on the"
200 DATA 9,"bottom of the sc
reen."
210 DATA 15,"After a while e
verything"
220 DATA 16,"gets corrected.
"
230 DATA 18,"It waits for yo
u to press"
240 DATA 19,"the <ENTER> key
250 DATA 99
260 CALL UPSIDE
30000 END
31500 SUB UPSIDE
31510 DIM W$(24):: FOR I=1 T
O 24 :: W$(I)="" :: NEXT I
31520 READ R :: IF R<25 THEN
 READ W$(R):: GOTO 31520
31530 CALL LINK("FLIP"):: CA
LL CLEAR :: FOR I=1 TO 24
31540 DISPLAY AT(24-I,14-LEN
(W$(I))/2):W$(I)
31550 NEXT I :: CALL DELAY
31560 FOR I=1 TO 24 :: CALL
HCHAR(I,1,32,32):: DISPLAY A
T(I,14-LEN(W$(I))/2):W$(I)::
 NEXT I :: CALL DELAY :: CAL
L LINK("FLIP")
31570 DISPLAY AT(24,1):"PRES
S <ENTER>." :: CALL KEY(0,K,
S):: IF K<>13 THEN 315703158
0 CALL CLEAR :: SUBEND
31590 SUB DELAY :: FOR D=1 T
O 1000 :: NEXT D :: SUBEND
```

You can substitute your own DATA statements for creating your own screens by using the format: DATA row,text (See the examples.) The text may not be more than 28 characters long. To indicate the end of the screen, use DATA 99 (or any number greather than 24). Follow this by a CALL UPSIDE, the subprogram which handles the screen display. If you want another page, continue with more DATA statements, and and end with DATA 99 and CALL UPSIDE again.

Another application of the FLIP routine is to "twirl" the letters very quickly to get the effect of a "whirligig". Try the following program:

```
100 !**************
110 !* WHIRLIGIG *
120 !**************
130 DATA 1,"WHIRLIGIG"
140 DATA 2,"---------"
150 DATA 3,"by Roy Tamashiro
"
160 DATA 5,"In this routine
the text"
170 DATA 6,"twirls upside-do
wn and back"
180 DATA 7,"up again as it a
ppears on"
190 DATA 8,"the screen."
200 DATA 10,"The text on eac
h row is"
210 DATA 11,"centered."
220 DATA 15,"It waits for yo
u to"
230 DATA 16,"press <ENTER> b
efore erasing"
240 DATA 17,"the screen."
250 DATA 99
260 CALL WHIRL
30000 END
31000 SUB WHIRL
31010 DIM W$(24):: FOR I=1 T
O 24 :: W$(I)="" :: NEXT I
31020 READ R :: IF R<25 THEN
 READ W$(R):: GOTO 31020
31030 CALL CLEAR :: FOR I=1
TO 24 :: IF W$(I)="" THEN 31
050
31040 DISPLAY AT(I,14-LEN(W$
(I))/2):W$(I):: FOR L=1 TO 1
0 :: CALL LINK("FLIP"):: NEX
T L
31050 NEXT I
31060 DISPLAY AT(24,1):"PRES
S <ENTER>." :: CALL KEY(0,K,
S):: IF K<>13 THEN 31060
31070 CALL CLEAR :: SUBEND
```

Enjoy these upside - dow character routines, and inven some new ones too. But, b careful not to flip too muc over them.

****

ELECTRONIC DATEBOOK

Dr. Roy T. Tamashiro

****

With the New Year upon us, it is time for reflections on the past year, and resolutions and planning for 1987. The program below, ELECTRONIC DATEBOOK, may be helpful in keeping track of appointments, birthdates, anniversaries and other important dates. It is a monthly calendar which allows you to enter notes or memos for any day of the year.

To use ELECTRONIC DATEBOOK, type the program listed below using Extended BASIC and save the program on your disk or cassette. When you RUN the program you will see the Main Menu, which asks whether you want to (1)View the Calendar, (2)Load the Date File, (3)Save the Date File, or (4)Exit. Choose option 1. At the next prompt, select the month, you wish to view. For example, enter 1 for January.

You will see the calendar for the month you selected. You may then choose to (1)Select a date, (2)Print or (3)Return to the main menu. If you choose (1)"Select a date", you are asked which date of the month. When you do so, you are asked to enter your memo for that date. You have four screen lines to do so.

When the calendar for the month is redisplayed each date for which you have an memo entry is starred(*). If you choose option 1 again ("Select a Date"), you will see the memo displayed.

Once you have several memos in a given month, you may want to get a printout. To do so choose option 2 from the menu under the calendar, then enter your printer device name, such as "PIO" or "RS232.BA=600", etc.

When you are finished entering information for the month, choose option 3 "Go to the main menu". From the main menu, choose option 3 "Save Date File". Pick DSK1 (disk), DSK2, or CS1 (cassette) for the device you are using. If you are using a disk, be sure you have at least 185 sectors free on your disk. Use a blank cassette if you choose CS1. The entire year's memos will be save on the disk or cassette file. Be patient as this process takes several minutes.

The next time you wish to consult the ELECTRONIC DATEBOOK, run the program and choose option 2 "Load Date File" from the main menu. All of the dates for which you made entries will now be ready for you to view, to revise or to print out.

```
100 !**************
110 !* ELECTRONIC *
120 !*  DATEBOOK  *
130 !* -- 1987 -- *
140 !**************
150 !AUTHOR:ROY TAMASHIRO
160 !USES X-BASIC
170 DIM D$(11,31,4)
180 M$="5Jan311Feb281Mar314A
pr306May312Jun304Jul317Aug31
3Sep305Oct311Nov303Dec31"
190 GOSUB 500 :: W1$="Author
:Roy Tamashiro" :: R=3 :: GO
SUB 490
200 DISPLAY AT(8,7):"1 To Vi
ew Calendar":TAB(7);"2    Lo
ad Date File":TAB(7);"3    S
ave Date File":TAB(7);"4
Exit"
210 DISPLAY AT(19,5):"YOUR C
HOICE (1-4):" :: ACCEPT AT(1
9,24)SIZE(1)VALIDATE("1234")
:C$
220 GOSUB 500 :: ON VAL(C$)G
OTO 230,400,440,480
230 DISPLAY AT(4,1):"Enter M
onth (1-12):" :: ACCEPT AT(4
,21)SIZE(2)VALIDATE(DIGIT):M
 :: M=M-1 :: IF M<0 OR M>11
THEN 230
240 GOSUB 500 :: M1$=SEG$(M$
,M*6+2,3):: DISPLAY AT(4,10)
:M1$;" 1987" :: DISPLAY AT(5
,1):"Sun Mon Tue Wed Thu Fri
 Sat"
250 N=0 :: S=VAL(SEG$(M$,M*6
+1,1)):: W$="" :: FOR K=S TO
 7 :: N=N+1 :: IF D$(M,N,0)<
>"" THEN W$=W$&"*" ELSE W$=W
$&" "
260 W$=W$&STR$(N)&"  " :: NE
XT K :: DISPLAY AT(6,(S-1)*4
+1):W$
270 W$="" :: L=VAL(SEG$(M$,M
*6+5,2)):: FOR K=N+1 TO L ::
 N=N+1 :: IF D$(M,K,0)<>"" T
HEN W$=W$&"*" ELSE W$=W$&" "
280 IF N<10 THEN W$=W$&STR$(
N)&"  " ELSE W$=W$&STR$(N)&"
 "

290 NEXT K :: DISPLAY AT(7,1
):W$ :: CALL HCHAR(12,1,45,3
2):: DISPLAY AT(13,1):"* ind
icates dates noted."
300 DISPLAY AT(15,3):"1 to S
elect a Date":TAB(3);"2 to P
rint":TAB(3);"3 to go to men
u":TAB(9);"Your Choice (1-3)
:"
310 ACCEPT AT(18,28)SIZE(1)V
ALIDATE("123"):C$ :: GOSUB 5
10 :: ON VAL(C$)GOTO 320,350
,190
320 DISPLAY AT(15,1):"Enter
Date:(1-";STR$(L);")" :: ACC
EPT AT(15,19)SIZE(2)VALIDATE
(DIGIT):DD :: IF DD<1 OR DD>
L THEN 320
330 GOSUB 510 :: DISPLAY AT(
15,1):"Type memo for ";M1$;D
D;"below." :: FOR I=0 TO 3 :
: DISPLAY AT(I+18,1):D$(M,DD
,I):: NEXT I
340 CALL VCHAR(18,2,62,4)::
FOR I=0 TO 3 :: ACCEPT AT(I+
18,1)SIZE(-28):D$(M,DD,I)::
NEXT I :: GOTO 240
350 DISPLAY AT(15,1):"Enter
Device Name:":"PIO" :: ACCEP
T AT(16,1)SIZE(-28):P$
360 OPEN #1:P$ :: PRINT #1:T
AB(30);M1$;" 1987"
370 FOR I=1 TO L :: IF D$(M,
I,0)<>"" THEN PRINT #1:M1$;I
;"1987" ELSE 390
380 FOR J=0 TO 4 :: PRINT #1
:D$(M,I,J):: NEXT J :: PRINT
 #1:""
390 NEXT I :: CLOSE #1 :: GO
TO 240
400 W1$="Load Date File" ::
R=3 :: GOSUB 490 :: GOSUB 52
0 :: IF C$="4" THEN 190
410 OPEN #1:F$,INPUT ,INTERN
AL,FIXED 128
420 FOR I=0 TO 11 :: FOR J=1
 TO VAL(SEG$(M$,I*6+5,2))::
FOR K=0 TO 2 :: INPUT #1:D$(
I,J,K),:: NEXT K :: INPUT #1
:D$(I,J,3):: NEXT J
430 NEXT I :: CLOSE #1 :: GO
TO 190
440 W1$="Save Date File" ::
R=3 :: GOSUB 490 :: GOSUB 52
0 :: IF C$="4" THEN 190
450 OPEN #1:F$,OUTPUT,INTERN
AL,FIXED 128
460 FOR I=0 TO 11 :: FOR J=1
 TO VAL(SEG$(M$,I*6+5,2))::
FOR K=0 TO 2 :: PRINT #1:D$(
I,J,K),:: NEXT K :: PRINT #1
:D$(I,J,3):: NEXT J
470 NEXT I :: CLOSE #1 :: GO
TO 190
480 CALL CLEAR :: END
490 DISPLAY AT(R,14-LEN(W1$)
/2):W1$ :: RETURN
500 CALL CLEAR :: W1$="ELECT
RONIC DATEBOOK: 1987" :: R=1
 :: GOSUB 490 :: CALL HCHAR(
2,4,45,25):: RETURN
510 CALL HCHAR(15,1,32,288):
: RETURN
520 DISPLAY AT(6,3):"1 for D
SK1":TAB(3);"2     DSK2":TAB
(3);"3     CS1":TAB(3);"4
  None of the above"
530 DISPLAY AT(15,5):"Your C
hoice: (1-4)" :: ACCEPT AT(1
5,25)SIZE(1)VALIDATE("1234")
:C$
540 IF C$="3" THEN F$="CS1"
ELSE F$="DSK"&C$&".DATEFILE"
550 RETURN
```

MEETING DATES

JANUARY 27
FEBRUARY 24
MARCH 24

HAPPY NEW YEAR

TWO-WAY COMMUNICATIONS FOR X-BASIC

Dr. Roy T. Tamashiro

Have you ever tried to use BASIC or Extended BASIC to communicate with another computer? If so, you soon discover that you can send and receive data (using PRINT # and INPUT # or LINPUT # instructions), but there are no instructions which allow simultaneous two-way communications. This "chat" mode is not available in the ordinary BASICs because you may not interrupt the PRINT # and (L)INPUT # instructions which are waiting for carriage returns. This is like having a walkie-talkie conversation in which each person must say "OVER" and release a switch before the other person can talk.

The "COMMUNICATIONS TERMINAL" program (See listing below and footnote) allows simultaneous two-way interaction. To use the program, you must have Extended BASIC, 32-K Memory Expansion, and an RS-232 Interface connected to a modem and phone line or direct wire to another computer. Type the program and SAVE it to your disk or cassette.

When you RUN the program, the screen clears, and whatever you type appears on your screen and is sent to the remote computer at 300 Baud. Whatever the remote computer sends to your computer appears on your screen also. When you or the remote computer sends a carriage return i.e. [Enter], a new line begins. The [left-arrow] key (CHR$(8)) acts as a Backspace on the screen.

To exit the routine, press [FCTN-9]. The remote computer can also allow you to exit the terminal send a [Control-O] or CHR$(15). To return to the terminal, type CALL LINK("START") and press [Enter].

```
100 !##################
110 !# COMMUNICATIONS #
120 !#    TERMINAL    #
130 !##################
140 !AUTHOR:ROY TAMASHIRO
150 !FEBRUARY 1987,X-BASIC,
MEMORY EXPANSION & RS232
160 CALL INIT :: CALL LOAD(8
196,63,248):: CALL LOAD(1637
6,83,84,65,82,84,32,48,0)
170 CALL LOAD(12288,2,224,50
,114,4,192,2,1,96,0,4,32,32,
32,5,128,2,128,3,0,22,250,2,
0)
180 CALL LOAD(12312,15,128,2
,1,50,96,2,2,0,16,4,32,32,36
,2,6,15,137,200,6,131,86,4,3
2)
190 CALL LOAD(12336,49,118,0
,8,4,192,2,12,19,64,31,21,22
,14,4,32,48,174,192,32,50,14
6,152,32)
200 CALL LOAD(12360,50,178,5
0,181,22,6,4,224,131,124,2,2
24,131,224,4,96,0,112,4,196,
216,4,131,124)
210 CALL LOAD(12384,4,32,32,
28,216,32,131,124,131,124,19
,229,208,96,131,117,152,1,50
,181,19,236,6,160)
220 CALL LOAD(12408,48,206,1
93,0,2,0,16,0,4,32,32,32,2,0
,15,128,2,1,3,0,4,32,32,32)
230 CALL LOAD(12432,2,0,15,1
33,2,1,1,0,4,32,32,32,2,6,15
,137,200,6,131,86,4,32,49,11
8)
240 CALL LOAD(12456,0,8,192,
4,16,214,50,146,48,178,192,3
2,50,114,2,12,19,64,31,21,22
,252,54,32)
250 CALL LOAD(12480,50,178,2
08,96,50,178,29,18,6,160,48,
206,3,128,152,1,50,180,22,5,
2,64,255,224)
260 CALL LOAD(12504,2,32,0,3
2,16,17,152,1,50,183,22,5,2,
128,0,0,19,37,6,0,16,35,216,
1)
270 CALL LOAD(12528,50,182,2
,33,96,0,4,32,32,32,208,96,5
0,182,5,128,2,128,3,0,17,23,
2,0)
280 CALL LOAD(12552,0,32,2,1
,50,184,2,2,2,224,4,32,32,44
,4,192,4,32,32,36,2,0,2,224)
290 CALL LOAD(12576,2,1,96,0
,4,32,32,32,5,128,2,128,3,0,
22,250,2,0,2,224,4,91,0,0)
300 CALL LOAD(12600,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0)
310 CALL LOAD(12624,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0)
320 CALL LOAD(12648,0,0,0,0,
0,0,0,0,0,100,32,0,46,170,49
,68,49,122,193,126,83,224,49
,114)
330 CALL LOAD(12672,192,32,1
31,86,194,64,2,41,255,248,4,
32,32,40,208,193,9,131,7,4,2
,2,49,100)
340 CALL LOAD(12696,5,128,5,
132,128,196,19,6,4,32,32,40,
220,129,152,1,49,116,22,246,
193,4,19,82)
350 CALL LOAD(12720,2,132,0,
7,21,79,4,224,131,208,200,4,
131,84,200,4,49,62,5,132,168
,4,131,86)
360 CALL LOAD(12744,200,32,1
31,86,49,64,2,224,131,224,4,
193,2,12,15,0,195,12,19,1,30
,0,2,44)
370 CALL LOAD(12768,1,0,4,22
4,131,208,2,140,32,0,19,50,2
00,12,131,208,29,0,2,2,64,0,
152,18)
380 CALL LOAD(12792,49,117,2
2,238,160,160,49,78,16,3,192
,160,131,210,29,0,192,146,19
,230,200,2,131,210)
390 CALL LOAD(12816,5,194,19
4,114,209,96,131,85,19,9,156
,133,22,242,9,133,2,6,49,100
,156,182,22,237)
400 CALL LOAD(12840,6,5,22,2
52,5,129,200,1,49,66,200,9,4
9,60,200,12,49,58,6,153,16,2
26,30,0)
410 CALL LOAD(12864,2,224,49
,68,192,9,4,32,32,40,9,209,2
2,4,3,128,2,224,49,68,4,193,
6,193)
420 CALL LOAD(12888,215,65,2
43,224,49,114,3,128,0,0,16,0
,40,0,0,0,0,6,82,83,50,51,50
,46)
430 CALL LOAD(12912,50,46,47
,188,5,136,152,56,53,30,19,6
,6,8,16,4,4,224,47,186,2,8,5
5,6)
440 CALL LOAD(12936,7,32,47,
184,2,12,41,172,200,12,52,16
,168,12,52,10,2,12,41,252,13
1,32,52,10)
450 CALL LOAD(12960,17,5,200
,32,52,16,52,10,5,160,47,174
,6,12,200,12,52,14,0,0,13,15
,0,8)
460 CALL LOAD(12984,19,52,6,
8,152,56,47,182,19,13,6,8,6,
160,53,44,195,76,5,141,6,8,1
52,56)
470 CALL LINK("START")
480 END
```

If your line out of the RS-232 interface has a Y-cable and you are using port 2, insert the following line to run the program via RS232/2:

```
465 CALL LOAD(12321,18):: CA
LL LOAD(12345,128):: CALL LO
AD(12905,8):: CALL LOAD(1291
1,47):: CALL LOAD(12473,128)
```

NOTE: The program is adapted from notes published by Texas Instruments Inc. on "9902 Asynchronous Communications Controller" and "DSRLNK Routine for XB".

MEETING DATES

FEBRUARY 24
MARCH 24
APRIL 28

COMPUTING CALORIES

I'm indebted to an amateur computer-users' group for a tongue-in-cheek estimate of the calories you can burn per hour with the following activities:

- Deciphering unfriendly software manuals/150
- Making back-up copies/50
- Finding the cause of a glitch/300

To which I would add: Listening to computer jargon from dedicated users/500!

A STUDENT'S SATELLITE COMPUTER

Dr. Roy T. Tamashiro

Terrilyn Morris has become a "star" in her eighth grade class. She brings a computer to school everyday. No, it is not the TI-99/4a, which she has loved using for school work in the last seveal years. She especially relies on the TI for doing her writing assignments. She seems so attached to her TI that she proably would not part with it for anything...except maybe if she could get Bruce Springsteen's autograph.

The computer Terri carries around with her is a Tandy Model 102 Portable Computer. It is about the size of a small notebook and fits easily in her backpack. It runs on batteries, but she also carries a power adapter to use when she can plug it in because that will keep the batteries from dying sooner. The Model 102 has 24K of RAM, and that is just enough to store a day's worth of notes she takes in class and in the library.

Terri's teachers and her classmates are really impressed when she takes out the Model 102 in the social studies class or the biology class, and she starts typing as the teacher is explaining some new material on the board. She can take just as much notes on it as the other students writing on paper. Actually she seems to write more quickly than the others, even though she is not a fast typist. This is because the word processor built into the computer allows her to do things like "copy text" and cutting and pasting very quickly and neatly.

Mr. and Mrs. Morris, Terri's parents, are pleased, but they were not sure it would turn out like this. They were worried that Terri's classmates would cast her as a snob or an "egghead" by taking notes on a computer. They were worried too about a school rule that electronic equipment like Walkmans and boom boxes were not allowed. So before they permitted Terri to take the computer to school they phoned the school, and discussed their concerns with the principal and the teacher. Both the principal and the teacher were reassuring. They said it was fine for Terri to bring the computer to school and that they would observe what happened and inform them if any problems came up.

When Terri gets home, she connects the Model 102 to her TI using an RS-232 cable and sends all her notes to a disk in the TI. Later she loads her notes into the TI-Writer word processor and does all the editing she needs to do. She "merges" the days notes with the notes she took on previous days, organizes them according to the study units and saves them back to the TI disk. From time to time she gets a printout of her notes. She uses the printout to study for tests. She says it is much easier to study from typed notes than from handwritten notes.

Some of Terri's friends also think it is easier to read and study from printout notes. Several have secretly offered to pay her a dollar for a copy of her printed out unit notes. Terri is not sure this is right, so she hasn't collected money from them, but she has let them study from her notes when she's not using them.

Terri is most thrilled about how much time she saves when she does research reports. When she goes to the library, she types information from her sources like reference materials, books and magazines into her Model 102. When she gets home, she transfers the file onto TI-Writer like she did with class notes. She now has the information ready to create her report. She need not retype the notes she took at the library as her classmates must do. Even her friends who have word processors at home, must take notes by hand in the library, then retype their notes into the word processor. Terri is a step ahead of them too.

In the last month, two other students in Terri's school have started to bring Model 102 portable computers to school. Terri, her teachers, and her parents are delighted to have stimulated a new approach to studying in the school.

If you would like to set up a satellite computer system like Terri did, or if you would like to help a student do so, you will need the following equipment: TI-99/4a Computer with 32K Memory Expansion, RS-232, at least one disk drive, printer and cable, Radio Shack/Tandy Model 100, 102 or 200 portable computer, and a standard DB-25 RS-232-C cable. The necessary software (word processor and communications) for the Model 100/102/200 is built-in the computer. For the TI, a communications software, such as Terminal Emulator II, Fast-Term or 4A-Talk, and a word processor like TI-Writer are required. To transfer files between the Model 100/102/200 and the TI, connect them with the RS-232 Cable, match the parameters on both communications software and use the normal commands for uploading and downloading text (ASCII) files on the two computers.

MEETING DATES

APRIL 28
MAY 26
JUNE 23

Using the Word Processor to Communicate with the Student's Satellite Computer

Dr. Roy T. Tamashiro

<u>NOTE:</u> In the April 1987 issue of <u>The Computer Bridge</u>, we introduced Terrilyn Morris, an eighth grade student who used a portable notebook computer (Tandy Model 102) in school and worked with the material on her TI-99/4a at home.

Several months have passed now since Terrilyn Morris first started bringing her notebook portable computer to school with her. The small Tandy 102 has become a faithful companion to her, and it works handily with the TI-99/4a she has at home. As she relies more on the Tandy as a satellite computer for her TI, she has run into a few problems. Now, with at least 10 other students in the school who also carry around notebook portables, she can enlist their help in solving some of these problems. But she solves many problems on her own too. Terry is especially proud of this one:

One afternoon, Terry was about to transfer the notes she took at the library from her Tandy to a TI disk. She had done this transfer with an RS-232 cable many times with no problem. (See <u>The Computer Bridge,</u> April 1987.) But then she discovered that the communication program she uses on the TI was gone! Her cousin Barbara was visiting over the weekend, and the disk must have gotten misplaced in the excitement. It was important that to get these notes transferred because she had already organized a lot for her paper on a TI-Writer word processor file.

How was Terry going to solve this problem? She looked around for a while, but couldn't find the disk. She tried to phone Marvin Lenworth, who also has a TI computer system, so she could borrow his copy of "4a-Talk." But there was no answer. In a way Terry was glad he didn't answer because she thinks he is sort of wierd, and she wasn't sure whether she wanted to talk to him.

After thinking a while, she wondered if TI-Writer might be able to receive her file directly from the Tandy 102, without a communications program. She tried several ways of doing this, getting all kinds of errors in the process. But her perseverance paid off. She got it to work. She learned that she could send the notes on the Tandy directly to TI-Writer, without a communications program!

Terry was so thrilled she didn't bother to look further for her lost disk. She soon discovered that, in addition to sending files from Tandy to TI, she could send files from TI to Tandy as well. So Terry ended up with a method for sending AND receiving files between TI-Writer and the Tandy portable, without a communications program! Of course, the next day, Terry was so excited about her discovery, she was talking more about this than about the research paper she was writing.

Here is how you can link TI-Writer and the Tandy like Terry did: Connect the RS-232 cable between the TI and the Tandy as usual. On TI-Writer, enter the Editor and type LF (load file) and [Enter]. At the prompt, "Load file, enter filename:," type RS232.PA=N.LF and press [Enter]. On the Tandy, select TELCOM, then press F3 (Stat) and use "37N1E" for the settings. Then press F4 (Term) to enter the terminal program, and F3 (Upload) and enter the name of the file on the Tandy. After the file is sent, press CLEAR (FCTN-4) on the TI. Although an error is shown, type E to edit, press [Enter] and the entire file will appear.

To send a document from TI-Writer to the Tandy, you may use either the Editor or the Formatter. On the Tandy, select F2 (Download) and answer the prompts. From TI-Writer's Editor, select PF (Print file), type "RS232.PA=N.LF" and press [Enter]. From the Formatter, use "RS232.PA=N.LF" as the "Print devicename."

AUTOMATED ADDRESS BOOK

Dr. Roy T. Tamashiro

Several readers who liked the QUICK LABEL program (See THE COMPUTER BRIDGE, October 1986) suggested that the program would be more useful if you did not need to type in frequently used addresses. The AUTOMATED ADDRESS BOOK program (listing below) allows you to keep addresses quite permanently on disk or cassette, and print them on regular-size or business envelopes or labels. On envelopes, the program will properly position your return address and the addressee. The program also contains a "search" feature which allows you to find records using any part of a name or address. For example, you can search for all JIMs, or all 631 zip codes.

Type in the program below using Extended BASIC and save it on cassette or disk. When you RUN the program, you see a menu. Start a new address file by selecting option 4 (Work with Return Address). You have four lines to type your return address. Press [Enter] if a line is blank. Select option 2 (Add Names) to enter your addressees. Again you have four lines to type the address. When you choose option 3 (See/Edit/Print Names) you are prompted for a SEARCH string. Type the item you want to find. (Be sure you match upper or lower cases.) To see all the records, leave the prompt blank and press [Return]. Use [PROC'D] (FCTN-6) to go on, [BACK] to go the menu, [ENTER] to edit or [BEGIN] (FCTN-5) to print the address that is shown. When print is chosen, you must enter your printer name, such as PIO or RS232 (and extensions). Then the various printing options are displayed. Use option 5 from the main menu to save your work on cassette or disk. When you start the next session, option 1 (Load Names) will allow you to retrieve your address file.

Note that you can organize your addresses using this program by having separate files for different groups of people. For example, one file might be for co-workers, another for relatives, and another for the cooking club members.

```
100 !****************
110 !*  AUTOMATED    *
120 !* ADDRESS BOOK *
130 !****************
140 !AUTHOR:ROY TAMASHIRO
150 !1987, THE COMPUTER BRID
GE
160 DIM A$(255):: P$="PIO" :
: E=1
170 B$(1)="Load Names" :: B$
(2)="Add Names" :: B$(3)="Se
e/Edit/Print Names" :: B$(4)
="Work with Return Address"
:: B$(5)="Save Work" :: B$(6
)="Exit"
180 CALL CLEAR :: DISPLAY AT
(1,4):"AUTOMATED ADDRESS BOO
K": :TAB(5);"1987, Roy Tamas
hiro" :: CALL HCHAR(2,6,45,2
2)
190 FOR I=1 TO 6 :: DISPLAY
AT(9+I,1):I;B$(I):: NEXT I
200 DISPLAY AT(24,1):"YOUR C
HOICE (1-6):" :: ACCEPT AT(2
4,20)VALIDATE("123456")SIZE(
1):C$ :: C=VAL(C$)
210 CALL CLEAR :: DISPLAY AT
(1,14-LEN(B$(C))/2):B$(C)::
ON C GOTO 220,250,270,440,45
0,460
220 GOSUB 470 :: OPEN #1:F$,
INPUT ,INTERNAL,FIXED 128 ::
 A=0
230 INPUT #1:A$(A):: IF EOF(
1)THEN 240 ELSE A=A+1 :: GOT
O 230
240 CLOSE #1 :: GOTO 180
250 W$="" :: GOSUB 500 :: GO
SUB 530 :: IF W$<>RPT$(CHR$(
32),84)THEN A=A+1 :: A$(A)=W
$
260 GOTO 180
270 DISPLAY AT(20,1):"Enter
SEARCH STRING:":"(Enter blan
k to see all.)";S$
280 ACCEPT AT(22,1)SIZE(-28)
:S$ :: IN=1
290 FOR SR=IN TO A :: IF S$=
"" OR POS(A$(SR),S$,1)>0 THE
N 300 ELSE 430
300 GOSUB 560 :: W$=A$(SR)::
 GOSUB 510 :: DISPLAY AT(20,
1):"[BEGIN]=Print [PROC'D]=N
ext [BACK]=Menu   [ENTER]=Ed
it"
310 CALL KEY(0,K,S):: IF S=0
 OR K<12 OR K>15 THEN 310
320 IF K<>13 THEN 350 ELSE G
OSUB 500 :: GOSUB 530 :: A$(
SR)=W$
330 IF W$<>RPT$(CHR$(32),84)
THEN 430 ELSE IF A=SR THEN A
=A-1 :: GOTO 430
340 FOR I=SR TO A-1 :: A$(I)
=A$(I+1):: NEXT I :: A$(A)="
" :: A=A-1 :: IN=SR :: GOTO
290
350 IF K=15 THEN SR=A
360 IF K<>14 THEN 430 ELSE G
OSUB 560 :: DISPLAY AT(15,1)
:"Enter Printer devicename:
  (Blank=Abort)":P$ :: ACCEP
T AT(17,1)SIZE(-28):P$
370 IF P$="" THEN 430 ELSE D
ISPLAY AT(15,1):"PRINT ON:":
" 1 Regular Envelope":" 2 Bu
siness Envelope":" 3 Label":
" 4 Plain Paper":" 5 Cancel"
380 DISPLAY AT(21,1):"Your C
hoice (1-5):";E :: ACCEPT AT
(21,20)VALIDATE("12345")SIZE
(-1):E$ :: E=VAL(E$)
390 IF E=5 THEN 430 ELSE DIS
PLAY AT(21,1):"Set printer a
nd press ENTER to start prin
ting."
400 ACCEPT AT(22,20)SIZE(1):
E$ :: OPEN #1:P$ :: W=1 :: I
F E<3 THEN W$=A$(0):: GOSUB
550 :: PRINT #1:CHR$(10),CHR
$(13)
410 W$=A$(SR):: IF E=1 THEN
W=20 ELSE IF E=2 THEN W=30
420 GOSUB 550 :: CLOSE #1 ::
 GOSUB 560 :: GOTO 300
430 NEXT SR :: GOTO 180
440 W$=A$(0):: GOSUB 500 ::
GOSUB 530 :: A$(0)=W$ :: GOT
O 180
450 GOSUB 470 :: OPEN #1:F$,
OUTPUT,INTERNAL,FIXED 128 ::
 FOR I=0 TO A :: PRINT #1:A$
(I):: NEXT I :: CLOSE #1 ::
GOTO 180
460 END
470 DISPLAY AT(4,1):"Enter D
evice/Filename:":"(Blank=Abo
rt.)":F$
480 ACCEPT AT(6,1)SIZE(-28):
F$
490 IF F$="" THEN 180 ELSE R
ETURN
500 DISPLAY AT(20,1):"Enter
information on 4 lines above
:"
510 CALL HCHAR(5,1,45,32)::
CALL HCHAR(10,1,45,32)
520 FOR L=0 TO 3 :: DISPLAY
AT(L+6,1):SEG$(W$,L*28+1,28)
:: NEXT L :: RETURN
530 FOR L=1 TO 4 :: ACCEPT A
T(L+3,1)SIZE(-28):N$(L):: NE
XT L
540 W$=N$(1)&RPT$(CHR$(32),2
8-LEN(N$(1)))&N$(2)&RPT$(CHR
$(32),28-LEN(N$(2)))&N$(3)&R
PT$(CHR$(32),28-LEN(N$(3)))&
N$(4):: RETURN
550 FOR I=0 TO 3 :: PRINT #1
:TAB(W);SEG$(W$,1+I*28,28)::
 NEXT I :: PRINT #1:CHR$(10)
,CHR$(13):: RETURN
560 CALL HCHAR(15,1,32,320):
: RETURN
```

MEETING CHANGE — SEPTEMBER 29

MANHATTAN STREET FINDER

Dr. Roy T. Tamashiro

Terrilyn Morris is now in the ninth grade. Remember when Terrilyn became a star in her eighth grade class when she used a Tandy Model 102 computer as a satellite computer to her TI-99/4a, which she used at home. (See THE COMPUTER BRIDGE, April 1987 and May 1987 issues.)

Over the summer, Terri got a chance to visit her favorite cousins in New York City. With their help, she learned her way around Manhattan. She got to be pretty resourceful, using the phone book to find out where "important" places like record stores and movie theaters were located. She even found a "Key to Manhattan Street Numbers," which is a formula for locating the cross-street of almost any address in Manhattan. Terri got tired of figuring out cross-streets using a pencil and paper, so she typed the following TI-BASIC program called "Manhattan Street Finder" to help her.

```
100 REM*****************
110 REM*  MANHATTAN    *
120 REM* STREET FINDER *
130 REM*****************
140 REM AUTHOR:ROY TANASHIRO
 ,ED.D
150 REM 1987,THE COMPUTER BR
IDGE
160 PRINT TAB(3);"MANHATTAN
STREET FINDER"
170 PRINT TAB(3);"----------
--------------"
180 PRINT "ENTER ADDRESS (AL
L CAPS.) "
190 INPUT A$
200 B$=""
210 C$=""
220 K=0
230 V=0
240 A=POS(A$," ",1)
250 IF A=0 THEN 850
260 B$=SEG$(A$,1,A-1)
270 B=VAL(B$)
280 S=LEN(B$)+2
290 C$=SEG$(A$,S,LEN(A$)-S+1
)
300 L=LEN(C$)
310 RESTORE
320 FOR I=1 TO 15
330 READ D$,V1
340 IF C$<>D$ THEN 360
350 V=V1
360 NEXT I
370 IF C$<>"CENTRAL PARK WES
T" THEN 400
380 K=1
390 V=60
400 IF ((SEG$(C$,1,5)="AVE.
")+(L=6))=-2 THEN 410 ELSE 4
20
410 V=3
420 IF C$<>"5TH AVE." THEN 5
40
430 IF B>200 THEN 450
440 V=13
450 IF (B<201)+(B>400)THEN 4
70
460 V=16
470 IF (B<401)+(B>600)THEN 4
90
480 V=18
490 IF (B<601)+(B>775)THEN 5
10
500 V=20
510 IF (B<776)+(B>1286)THEN
540
520 K=1
530 V=-18
540 IF ((C$="7TH AVE.")+(B<1
801))=-2 THEN 550 ELSE 560
550 V=12
560 IF ((C$="7TH AVE.")+(B>1
800))=-2 THEN 570 ELSE 580
570 V=20
580 IF ((C$="RIVERSIDE DR.")
+(B<568))=-2 THEN 590 ELSE 6
10
590 V=73
600 K=1
610 IF ((C$="RIVERSIDE DR.")
+(B>567))=-2 THEN 620 ELSE 6
40
620 V=78
630 K=1
640 IF C$<>"BROADWAY" THEN 7
50
650 IF B>754 THEN 680
660 A=8
670 GOTO 810
680 IF (B<755)+(B>858)THEN 7
00
690 V=-29
700 IF (B<859)+(B>958)THEN 7
20
710 V=-25
720 IF (B>958)+(B<1000)=-2 T
HEN 850
730 IF B<1000 THEN 750
740 V=-31
750 IF V=0 THEN 850
760 A=INT(B/10)
770 IF K=1 THEN 790
780 A=INT(A/2)
790 A=A+V
800 IF (A<1)+(A>250)THEN 850
810 PRINT "CROSS STREET IS:"
;A;"ST."
820 PRINT "ANOTHER STREET(Y/
N)";
830 INPUT I$
840 IF I$<>"N" THEN 170 ELSE
 920
850 PRINT "ADDRESS NOT FOUND
."
860 GOTO 170
870 DATA "1ST AVE.",3,2ND AV
E.,3,3RD AVE.,10,4TH AVE.,8,
8TH AVE.,9
880 DATA "AVE. OF THE AMERIC
AS",-12,9TH AVE.,13,10TH AVE
.,13
890 DATA "11TH AVE.",15,AMST
ERDAM AVE.,59,COLUMBUS AVE.,
59
900 DATA "LEXINGTON AVE.",22
,MADISON AVE.,27,PARK AVE.,3
4
910 DATA "WEST END AVE.",59
920 END
```

Although the "Manhattan Street Finder" worked fine, Terri could make a lot more use of it if the program was in her portable Tandy computer that she carried around in her knapsack. Then if she needed to know the location of an address while she was riding the bus or train, she could do so handily just by pulling out her computer from her knapsack and typing in the address. She learned how to send and convert the "Manhattan Street Finder" program to her Tandy Computer. Here is how she did it:

1. She connected the TI to the Tandy with an RS-232 cable.
2. She loaded "Manhattan Street Finder" into the TI.
3. On the Tandy, she selected TELCOM, then pressed [F3] to reset the communications parameter. She typed 38N1E and [Enter], then pressed [F4] to go into the terminal. She pressed [F2] to "download" the program. At the filename prompt, she typed "MANH" (short for Manhattan).
4. On the TI, she typed LIST "RS232.DA=8.PA=N" and [Enter].
5. After the program was sent, she pressed [F2] on the Tandy, then [F8] to disconnect, and exit TELCOM.
6. From the main menu, she selected BASIC, typed in the following conversion program, and ran it.

```
100 REM -CONVERT FILE FOR TANDY
 MODEL 100/102
110 MAXFILES=2
120 INPUT  "ENTER FILE  TO CON
VERT";F$:OPEN F$ FOR INPUT AS 1
130 OPEN "TEMP.DO" FOR OUTPUT
AS 2
140 LINEINPUT #1,A$
150 IF  LEFT$(A$,4)="240  "THEN
A$="240 A=INSTR(1,A$,CHR$(32))"
160 A=INSTR(1,A$,"SEG$")
170 IF A>0 THEN MID$(A$,A)="MID
$":GOTO 160
180 PRINT #2,A$
190 IF EOF(1) THEN 200 ELSE 140
200 CLOSE 1,2
210 KILL F$+".DO"
220 NAME "TEMP.DO" AS F$+".DO"
```

7. Then she pressed [F8] for the menu, and selected the MANH.DO file from the main menu. She then pressed [F7], [Ctrl-Down Arrow], [F5], and [F8] in sequence.
8. She then entered BASIC, and pressed the PASTE key. She could then RUN the program in her Tandy.
9. Of course, she also saved it to the Tandy's memory by typing, SAVE "MANH" and pressing [Enter].

The Word Counter

Dr. Roy T. Tamashiro

Many people feel so good about their writing when they learn to use a word processor, they believe their writing deserves to be published. And they are often right. Some of their writing does deserve to be published. Are you one of these hopeful writers? If so, you have probably searched through the publication guidelines of periodicals or journals, and run across a statement like, "We invite writers to submit unsolicited manuscripts."

Your fantasy goes wild. "Unsolicited could mean me. This might be a chance for someone to discover that I'm the next Ernest Hemingway."

You read on: "The manuscript length should be 8,000-10,000 words." You're puzzled. You ask yourself, "How much is 8,000 or 10,000 words?" You search your memory. You remember that your high school English teacher mentioned some sort of formula for estimating how many words there were on a typed, double-spaced page. But you can't remember the formula. Was it 300 words per page, or 250, or 600?

The program listed below, THE WORD COUNTER, will count the exact number of words in a TI-Writer document, or other documents saved in the "Display/Variable 80" format. To get a word count of a manuscript, type the program using Extended BASIC and save it. When you RUN the program, you must indicate whether you are working with a TI-Writer document or another D/V 80 document. The program filters out "formatter commands" as it analyzes TI-Writer files. No filtering is done on other D/V 80 files. The total word count is shown after the file has been analyzed. In addition, the number of syllabified words are given -- these are words which are hypenated at the end of one line and continued on the next line.

```
100 !****************
110 !* WORD COUNTER *
120 !****************
130 !AUTHOR:ROY TAMASHIRO,ED
.D
140 !APRIL 1988
150 CALL CLEAR :: S2$=RPT$("
 ",2):: WC=0 :: HY=0
160 DISPLAY AT(1,9):"WORD CO
UNTER":TAB(9);"------------"
:TAB(5);"1988, ROY TAMASHIRO
"
170 DISPLAY AT(6,1):"PRESS:"
: :TAB(4);"1 FOR TI-WRITER F
ILE": :TAB(4);"2    OTHER D
/V 80 FILE": :TAB(4);"3 TO
QUIT"
180 DISPLAY AT(22,7):"YOUR C
HOICE (1-3):" :: ACCEPT AT(2
2,26)VALIDATE("123")SIZE(1):
C$
190 ON VAL(C$)GOTO 200,200,4
00
200 DISPLAY AT(22,1):"ENTER
FILENAME:":"DSK1." :: ACCEPT
 AT(23,4)SIZE(-12):F$ :: F$=
"DSK"&F$ :: OPEN #1:F$
210 DISPLAY AT(6,1):"Analyzi
ng..."
220 LINPUT #1:A$ :: CALL HCH
AR(8,3,32,532):: DISPLAY AT(
8,1):A$
230 IF EOF(1)<>0 AND C$="1"
THEN 370
240 IF A$="" THEN 360
250 IF C$="2" THEN 280
260 IF SEG$(A$,1,1)="." THEN
360
270 T=POS(A$,"^",1):: IF T>0
 THEN A$=SEG$(A$,1,T-1)&" "&
SEG$(A$,T+1,LEN(A$)-T):: GOT
O 270
280 IF (SEG$(A$,1,1)<"!" OR
SEG$(A$,1,1)>"~")AND LEN(A$)
>1 THEN A$=SEG$(A$,2,LEN(A$)
-1):: GOTO 240
290 IF (SEG$(A$,LEN(A$),1)<"
!" OR SEG$(A$,LEN(A$),1)>"~"
)AND LEN(A$)>1 THEN A$=SEG$(
A$,1,LEN(A$)-1):: GOTO 240
300 IF A$=" " THEN 360
310 T=POS(A$,S2$,1):: IF T>0
 THEN A$=SEG$(A$,1,T)&SEG$(A
$,T+2,LEN(A$)-1):: GOTO 310
320 FOR I=1 TO LEN(A$)
330 IF SEG$(A$,I,1)=" " THEN
 WC=WC+1
340 IF I=LEN(A$)AND SEG$(A$,
I,1)="-" THEN HY=HY+1 :: WC=
WC-1
350 NEXT I :: WC=WC+1
360 IF EOF(1)=0 THEN 220
370 CLOSE #1
380 DISPLAY AT(6,1):"TOTAL W
ORD COUNT IS:";WC: :"INCLUDI
NG";HY:"SYLLABIFIED WORD(S).
"
390 DISPLAY AT(22,1):"ANOTHE
R COUNT? (Y/N)" :: ACCEPT AT
(22,22)VALIDATE("YN")SIZE(1)
:N$ :: IF N$="Y" THEN 150
400 CALL CLEAR :: END
```

A Coronary Risk Analysis Program

Roy Tamashiro, Ed.D.

With heart disease and cancer as the two leading causes of death in America, the media and many institutions tell us how to reduce the risk of these diseases. There are diets to follow, exercise programs to do, and warnings to stop smoking and limit alcohol use. But how do you decide how strictly you should follow these guidelines and warnings? A competent physician is probably your best resource on this question. In addition, published guidelines, like one by the American Health Foundation (AHF, 320 East 43rd Street, New York, NY 100017) to estimate one's risk of heart disease, may be helpful. The program below, CORONARY RISK RATE, computes your risk for heart disease according to the AHF guidelines.

To use the CORONARY RISK RATE program, type the program in Extended BASIC, and save it to disk or cassette. When you RUN the program, enter the information requested. (Systolic blood pressure refers to the top number in the ratio, 120780.) After you answer the questions, the program reports your level of risk according to the AHF guide, and suggests what you should do.

You may notice that other risk analysis guides differ from this one assembled by AHF. The reason is that each guide is based on research which has differing results. The AHF guide uses the calculations from the "Framingham Study 26 Year Followup," and a height/weight table from Metropolitan Life Insurance.

[NOTE: If you prefer not to type the program in yourself, send me $3.00 (or $5.00 from outside USA) (My address is: 1767 Robin Knoll Court; St. Louis, MO 63146) and I will provide a disk containing the program, as well as details on the method AHF used to calculate heart disease risk.]

```
100 !***********************
110 !* CORONARY RISK RATE *
120 !***********************
130 !AUTHOR:ROY TAMASHIRO
140 !JUNE 1988
150 ON WARNING NEXT
160 CALL CLEAR :: DISPLAY AT
(1,5):"CORONARY RISK RATE" :
: CALL HCHAR(2,7,45,18)
170 DISPLAY AT(3,1):"Program
 by: Roy Tamashiro":"Source:
 American Health":TAB(9);"Fo
undation"
180 DISPLAY AT(24,1):"Answer
 the Questions Above:" :: CA
LL HCHAR(23,1,45,32)
190 DISPLAY AT(7,1):"Blood C
holesterol Level:":"Systolic
 Blood Pressure:":"Cigarette
s Smoked Daily":" (Number of
 packs):"
200 DISPLAY AT(11,1):"Sex (M
/F):":"Height:" :: DISPLAY A
T(12,11):"ft." :: DISPLAY AT
(12,18):"in."
210 DISPLAY AT(11,15):"Age:"
 :: DISPLAY AT(13,1):"Weight
 (lbs.):":"Do you have diabe
tes?(Y/N):"
220 P=0 :: ACCEPT AT(7,25)SI
ZE(3)VALIDATE(DIGIT):CH
230 IF CH>299 THEN P=P+10 EL
SE IF CH>249 THEN P=P+7 ELSE
 IF CH>199 THEN P=P+3 ELSE I
F CH>179 THEN P=P+1
240 ACCEPT AT(8,25)SIZE(3)VA
LIDATE(DIGIT):BP
250 IF BP>159 THEN P=P+6 ELS
E IF BP>139 THEN P=P+4 ELSE
IF BP>119 THEN P=P+2
260 ACCEPT AT(10,21)SIZE(2)V
ALIDATE(NUMERIC):CG
270 IF CG>=2 THEN P=P+5 ELSE
 IF CG>=1 THEN P=P+3 ELSE IF
 CG>0 THEN P=P+1
280 ACCEPT AT(11,11)SIZE(1)V
ALIDATE("MFmf"):SX$ :: IF AS
C(SX$)>96 THEN SX$=CHR$(ASC(
SX$)-32)
290 IF SX$="" THEN 280
300 ACCEPT AT(11,19)SIZE(3)V
ALIDATE(DIGIT):AGE
310 IF AGE>59 THEN P=P+7 ELS
E IF AGE>49 THEN P=P+3
320 ACCEPT AT(12,9)SIZE(1)VA
LIDATE("456"):FT$ :: FT=VAL(
FT$)
330 ACCEPT AT(12,15)SIZE(2)V
ALIDATE(DIGIT):IC :: IF IC>1
1 THEN GOSUB 590 :: GOTO 330
340 IF SX$="M" AND(FT<5 OR(F
T=6 AND IC>4))THEN GOSUB 590
 :: GOTO 320
350 IF SX$="F" AND((FT=6)OR(
FT=5 AND IC=11)OR(FT=4 AND I
C<7))THEN GOSUB 590 :: GOTO
320
360 ACCEPT AT(13,16)SIZE(3)V
ALIDATE(DIGIT):WT
370 IF SX$="F" THEN 460
380 IF FT=6 THEN 430
390 IF IC<6 THEN X=116+3*IC
400 IF IC>5 THEN X=140+(IC-7
)*4
410 IF IC=11 THEN X=157
420 GOTO 490
430 IF IC<2 THEN X=161+5*IC
440 IF IC>1 THEN X=170+5*(IC
-2)
450 GOTO 490
460 IF FT=4 THEN X=73+3*IC
470 IF FT=5 AND IC>0 THEN X=
108+4*IC
480 IF FT=5 AND IC=0 THEN X=
109
490 IF WT>1.39*X THEN P=P+2
500 IF WT>1.19*X THEN P=P+2
510 IF WT>X THEN P=P+2
520 ACCEPT AT(14,28)SIZE(1)V
ALIDATE("YNyn"):DB$ :: IF DB
$="Y" OR DB$="y" THEN P=P+5
530 IF P<5 THEN R$="LOW" ELS
E IF P<14 THEN R$="MODERATE"
 ELSE IF P<19 THEN R$="HIGH"
 ELSE R$="VERY HIGH"
540 DISPLAY AT(16,1):"You ar
e at ";R$;" risk":"for heart
 disease." :: IF P<5 THEN 57
0
550 DISPLAY AT(18,1):"You sh
ould:" :: IF CG>0 THEN DISPL
AY AT(18,14):"Quit smoking,"
560 DISPLAY AT(19,1):"Reduce
 your weight,":"cholesterol
& blood pressure":"through d
iet & exercise."
570 DISPLAY AT(24,1):"Do ano
ther?(Y/N)" :: ACCEPT AT(24,
18)SIZE(1):K$
580 IF K$="Y" OR K$="y" THEN
 150 ELSE CALL CLEAR :: END
590 CALL SOUND(100,990,0)::
RETURN
```

COMPUTING FOR YOUR RETIREMENT INCOME

Dr. Roy Tamashiro

Although many of us are quite a few years away from retirement, we might be hoping to retire early, or to spend more years in retirement. To get a truly realistic picture of your financial future, you should consult an expert in this field. The computer program below ("RETIREMENT INCOME ANALYSIS") gives a very rough estimate of how you might be able enjoy a financially comfortable retirement.

"RETIREMENT INCOME ANALYSIS" works in TI-BASIC or EXTENDED BASIC. Type in the program, proofread it, and SAVE it to a tape or disk. When the program is run you are prompted for the following items:

This year is: 19__ [Look at a calendar if you don't know.]
Current Monthly Income: $___ [Enter your gross monthly income.]
Inflation Rate (%) ____ [Make an educated guess. The inflation rate is about 4 percent now.]
Years to Retirement:___ [Subtract your present age from your retirement age.]
Amount Invested for Retirement $____ [Include all of your retirement assets including IRAs, pension plans, retirement annuities, etc., but do NOT include non-retirement investments, checking or savings accounts, or Social Security.]
Assumed Interest Rate (%) ___ [Estimate an annual average of interest or dividends on your retirement investments between now and your retirement age.]
Annual Additions to Retirement Investments: $_____ [Estimate how much you will add per year to your investments.]

The program calculates how much income you will need when you retire. Many experts say that you need about 75 percent of your pre-retirement income, but with inflation included in the calculation, the monthly figure at retirement is likely to be higher than your present income.

Social Security benefits are not included in the analysis, nor are the taxes you will owe on your various tax-deferred investments. You may want to adjust the recommendations given to account for these and other omitted factors.

```
100 REM **************
110 REM * RETIREMENT *
120 REM *   INCOME   *
130 REM *  ANALYSIS  *
140 REM **************
150 REM 1988, ROY TAMASHIRO
160 DEF RATE=IT*.01
170 GOSUB 410
180 INPUT "This Year is: 19"
:NOW
190 INPUT "Current Monthly I
ncome:$":MONTHLY
200 INPUT "Inflation Factor
(%):":IT
210 INPUT "Years to Retireme
nt: ":YEARS
220 NEEDED=INT(((.75*MONTHLY
)*(1+RATE)^YEARS)+.5)
230 INPUT "Amount Invested f
or retire- ment: $":B
240 INPUT "Assumed Interest
Rate(%):":IT
250 INPUT "Annual additions
to retire- ment investment $
":ADD
260 FUTURE=INT(((B*(1+RATE)^
YEARS+ADD*(((1+RATE)^YEARS-1
)/RATE))*RATE)/12+.5)
270 GOSUB 410
280 PRINT "You will need abo
ut $";NEEDED:"per month when
 you retire in";1900+NOW+YEA
RS
290 PRINT :"You can expect t
o receive":"about $";FUTURE;
"per month on"
300 PRINT "your investment i
nterest or dividends."
310 IF FUTURE>=NEEDED THEN 3
90
320 PRINT :"If you will have
 no other":"income, then you
 should":"increase your inve
stments"
330 FIX=INT(((12*(NEEDED-FUT
URE))/RATE)/(1+RATE)^YEARS+.
5)
340 PRINT "by about $";FIX;"
to":"$";FIX+B;"this year, or
"
350 MORE=INT((((NEEDED-FUTUR
E)*12)/RATE)/(((1+RATE)^YEAR
S-1)/RATE)+.5)
360 PRINT "raise your annual
 additions by $";MORE;
370 PRINT "to total ":"$";MO
RE+ADD;"per year."
380 GOTO 400
390 PRINT :"You are in an ad
equeate     investment posit
ion for your retirement."
400 END
410 CALL CLEAR
420 PRINT " RETIREMENT INCOM
E ANALYSIS"
430 PRINT " ================
=========="
440 RETURN
```

REMINDER:

CHICAGO/MILWAUKEE FAIR
NOVEMBER 11-13
CONTACT TOM and/or JAN KNAPP FOR FURTHER DETAILS

COMPUTER BRIDGE SEPTEMBER 1988, P.5

CONVERTING MINI-MEMORY CASSETTE PROGRAMS TO DISKS

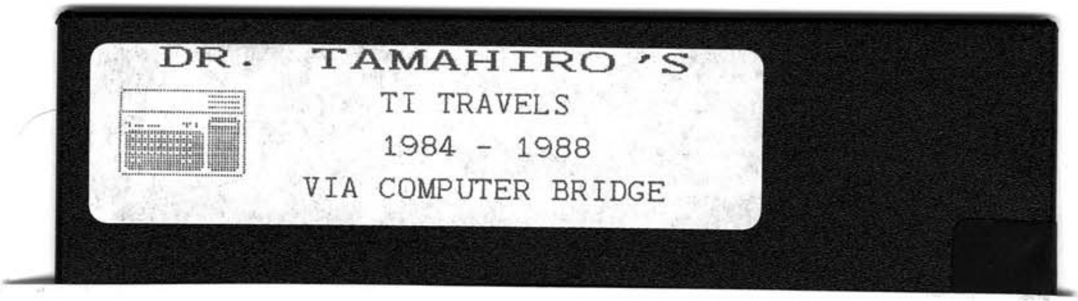
Roy T. Tamashiro, Ed.D.

The Mini-Memory Cartridge is an inexpensive way to be able to run Assembly Language programs if you do not have the Editor-Assembler package and Disk Drive system. For example, TI-MINIWRITER is a word processing program which will operate just on the console with Mini-Memory Cartridge and cassette tape recorder. For those who have disk drives, however, it would be handy to be able to load cassette-Based-Mini-Memory programs from disk rather than from cassette tape. The manual for Mini-Memory does not give a way to copy an assembly language program made for tape to a disk. The program below will allow you to do just that.

To use MINI-MEMORY TAPE-TO-DISK CONVERTER, the system must contain the 32-K Memory expansion and Disk Drive. Load the cassette program into the cartridge using the EASY-BUG "L" operation. Then exit using <QUIT> to the main title screen, and select TI-BASIC. Load and run the program below. At the FILE NAME prompt, type the disk filename under which you wish to save the program (for example, DSK1.PROGRAM). The program will begin to convert the contents of the cartridge at >7000 to >7FFF to an assembly language object code. This process will take about 25 minutes. A tone will sound when the job has been completed.

When you want to load the program from disk to the Mini-Memory Cartridge, select option 3 (Mini Memory) from the main selection list, then choose the "Load and Run" option(#1). At the prompt "FILE NAME?" type the name you gave your program (for example, DSK1.PROGRAM). Then press <ENTER>. After the file has loaded press <ENTER> again, and you will see the Mini-Memory Menu. If the program operates from Mini-Memory, press 2(Run) and type the name of the program. If the program operates from TI-BASIC via CALL LINK, then press <QUIT> (FCTN-=) and re-select TI-BASIC from the main selection list. Then load your BASIC program and RUN it.

```
100 REM *****************
110 REM * MINI-MEMORY  *
120 REM * TAPE-TO-DISK *
130 REM *  CONVERTER   *
140 REM *****************
150 REM    AUTHOR:
      ROY TAMASHIRO,ED.D.
         APRIL 1985
160 CALL CLEAR
170 EL=2
180 INPUT "FILE NAME: ":F$
190 OPEN #1:F$,DISPLAY ,FIXE
D 80
200 PRINT #1:"00000        9
A0009A020B02E0BA000B0201B000
0B0203B1000B80C1B13057F35DF
      0001"
210 PRINT #1:"9A030BD861BA04
6B7000B0581B10F9B0300B0002B0
2E0B83E0B0460B00707F2E6F
      0002"
220 FOR I=28672 TO 32767 STE
P 22
230 A$="9A"
240 L=I+12358
250 IF L<45056 THEN 280
260 L=L-4096
270 A$="9B"
280 L=L-40960
290 A=INT(L/256)
300 IF L<256 THEN 320
310 L=(L/256-A)*256
320 GOSUB 610
330 A$=A$&SEG$(H$.2.1)
340 A=L
350 GOSUB 610
360 A$=A$&H$
370 FOR J=0 TO 21 STEP 2
380 IF I+J+K<32768 THEN 410
390 J=21
400 GOTO 470
410 A$=A$&"B"
420 FOR K=0 TO 1
430 CALL PEEK(I+J+K,A)
440 GOSUB 610
450 A$=A$&H$
460 NEXT K
470 NEXT J
480 GOSUB 840
490 NEXT I
500 A$="9B042B4544B71B6"
510 GOSUB 840
520 A$="1A020"
530 GOSUB 840
540 A$="6A020LOAD0 7FD50F"
550 GOSUB 850
560 A$=":        99/4 AS"
570 GOSUB 850
580 CLOSE #1
590 CALL SOUND(3000,700,0)
600 END
610 H$=""
620 IF A<>0 THEN 650
630 H$="00"
640 RETURN
650 IF A>15 THEN 690
660 GOSUB 750
670 H$="0"&H$
680 RETURN
690 A2=(A/16-INT(A/16))*16
700 A=INT(A/16)
710 GOSUB 750
720 A=A2
730 GOSUB 750
740 RETURN
750 IF A<>0 THEN 780
760 H$=H$&"0"
770 RETURN
780 ON A GOSUB 800,800,800,8
00,800,800,800,800,800,820,8
20,820,820,820,820
790 RETURN
800 H$=H$&STR$(A)
810 RETURN
820 H$=H$&CHR$(A+55)
830 RETURN
840 A$=A$&"8F2E6F"
850 EL=EL+1
860 EL$=STR$(EL)
870 IF EL>9 THEN 900
880 EL$="00"&STR$(EL)
890 GOTO 920
900 IF EL>99 THEN 920
910 EL$="0"&STR$(EL)
920 FOR M=LEN(A$)+1 TO 76
930 A$=A$&CHR$(32)
940 NEXT M
950 A$=A$&"0"&EL$
960 PRINT #1:A$
970 RETURN
```



DR. TAMAHIRO'S
TI TRAVELS
1984 - 1988
VIA COMPUTER BRIDGE