

036
8706

From the Prez...

Have you ever heard the saying, The more you learn, the more you find out what you don't know. Well I have found this particularly true in working with computers. The better handle I get on what I'm doing, the more uninformed I feel. Even when it's a piece of common software, that I have used hundreds of times, I always seem to pick up something that I didn't know before.

For example, sometimes when I only have a daisy-wheel printer, I will print from the formatter to a disk file instead of the printer. This is a very fast way of previewing the text format. Once in the editor again I can view what its going to look like on paper. It is then possible to print it right from the editor, however sometimes the control characters mess things up. I found, while looking for something else, if you enter "C PIO", after entering PF for Print File, all characters below ASCII 32 will not be sent to the printer. Also, you can have it print the line numbers by entering "L PIO", or print in display fixed 80 format by entering "F PIO".

This may be common knowledge to most of you, but for me, every piece of software I use tends to be a constant learning experience. Maybe that's why it seems to hold my attention...

--Keith Fast

Club News & Views

There is an additional \$30.53 in the clubs treasury as a result of the paper drive last month. This gives our clubs general fund a balance of \$1330.00. So you see your effort did pay off! Voice tape recordings have been made of the programs the past few months and they are available from the librarians. They can be helpful in understanding certain important points. This month the program will be "Music and the TI, conducted by our own Dan Hawes. He will explain many of the procedures that he uses in his musical programs and he has written some really good ones. The workshops this month will be on Multiplan and more of the fine details on music programming for those with particular interest. Your editor was in Seattle last week and stopped in at both "Bits & Chips" and the "Queen Anne Computer Shoppe". The Queen Anne Computer Shoppe has moved out to 62nd. & Roosevelt Way N.E. but they plan to keep their old name. They are both preparing for the TI Faire next September (26th & 27th). with all the things happening in the TI world. Don't forget our swap meet on Saturday June 27th - you might find just what you are looking for. The board meeting on June 16th will be at the home of Ron Mayer (umm, strawberry shortcake). programs will include Forth, C-99, Fun'l Writer, and how to install a quiet fan in your P.E. Box. Remember this is your newsletter and if you have something to contribute or want to see certain things in it you are more than welcome. As mentioned elsewhere in this issue the library offerings this month will include three fine disks of tutorials from Tigercub Software.

* Murphy's Rule: *
* A shortcut is the longest *
* distance between two points. *
* *

What's Inside

Presidents Report Page 1
Club News 1
CALL KEY Commands 2
The Game Room "Catch it" 3
Electronic Datebook 4
Easy to use Mail List 5
PRINT USING 6
Slash Zero Program 6
Basic Programming 7

Club Officers

President Keith Fast 777-1531
Vice-President Dale Kirkwood 646-4354
Secretary Don Barker 223-1749
Treasurer Mike King 357-4413

PUNN Staff

Librarians Ron Mayer 232-7363
Walt Morey 239-5105
Jim Thomas 284-2425
Hardcopy Mike Calkins 636-1839
Program Chairman Ted Peterson 244-1587
Workshop Chairman (open)
Membership Chairman Terry Priest 649-9583
Newsletter Editor Charles Ball 639-0466
Ass't Editor Dan Hawes 620-9725

BBS Committee

Chairman: Al Kinney 640-5860
Ron Mayer 232-7363
Mike King 357-4413

BBS Phone Number 233-6804

CALL KEY Commands

The CALL KEY command in Basic and Extended Basic is one whose complete power may not be appreciated by many programmers. This article and list of examples is an attempt to explain some of the "hidden" capabilities of the CALL KEY statement so that you can get the most out of it in your programs.

The information in this article was collected from several sources including: an excellent summary of the CALL KEY options, written by Joyce Corker of Waltham, MA, and an article by Glenn Davis of the MSP 99 newsletter.

CALL KEY, as implemented on the TI 99/4A has six possible modes in which to operate. These modes are summarized below.

CALL KEY(0,KEY,STATUS)

When the mode specified is "0", the keyboard is scanned in the same mode it was in previously. (The normal Basic mode is Mode 5--see below--so when a CALL KEY(0,K,S) statement is used in Basic or Extended Basic, we are really telling the computer to scan using Mode 5).

CALL KEY(1,KEY,STATUS)

Mode 1 scans the left side of the keyboard only.

CALL KEY(2,KEY,STATUS)

Mode 2 scans the right side of the keyboard only.

CALL KEY(3,KEY,STATUS)

Mode 3 is the "99/4" mode. In this mode values for upper case letters are returned in "KEY" even if a lower case letter is pressed. (In other words, in this mode it doesn't matter whether the ALPHA LOCK key is up or down, all you get is upper case letters.)

This mode is particularly useful where upper case letters are important. For example, it is recommended that disk file names be all upper case letters. By putting a CALL KEY(3,K,S) statement before the INPUT or ACCEPT statement, the name typed in by the user will be all in upper case letters. (TI-Writer uses this mode when accepting file names.)

CALL KEY(4,KEY,STATUS)

Mode 4 (Pascal Mode) allows upper and lower case letters and all control and function keys. However, some of the "codes" are different than in BASIC. For example, FCIN 4 will not "break" a program on an INPUT or ACCEPT statement and FCIN 5 will not backspace. This is because these combinations of key strokes generate different codes than in BASIC. (See the complete list in the Users Guide Appendix, pages III-3, III-4.)

CALL KEY(5,KEY,STATUS)

Mode 5 is normal BASIC mode and allows for both upper and lower case

letters.

HERE ARE SOME EXAMPLES:

Yes or no answers using CALL KEY- - - -

```
100 CALL CLEAR
110 PRINT "Y OR N"
120 CALL KEY (0,K,S)
130 IF K=78 THEN 170
140 IF K<>89 THEN 120
150 PRINT "YES"
160 GOTO 180
170 PRINT "NO"
180 END
```

Space bar or ENTER answers using CALL KEY
5- - - -

```
100 DISPLAY AT(3,3)ERASE ALL:
    "PRESS SPACE BAR TO CONTINUE":
    "PRESS ENTER TO PRINT"
110 FOR DELAY=1 TO 600 :: NEXT DELAY
120 CALL KEY(5,K,S)
130 IF K=32 THEN PRINT "SPACE BAR
    PRESSED" :: GOTO 150 ELSE
    IF K<>80 THEN 120
140 PRINT "HERE YOU WOULD GO TO
    YOUR PRINT SUB"
150 END
```

Alphabet answers that are forgiving of
wrong case using CALL KEY 3- - - -

```
100 DISPLAY AT(3,3)ERASE ALL:
    "PRESS R TO REPEAT":
    "PRESS F TO PRINT"
110 FOR DELAY=1 TO 600 ::
    NEXT DELAY
120 CALL KEY(3,K,S)
130 IF K=82 THEN PRINT "HERE YOU
    WOULD GOTO YOUR REPEAT
    SUBPROGRAM" :: GOTO 150 ELSE
    IF K<>80 THEN 120
140 PRINT "HERE YOU WOULD GO TO
    YOUR PRINT SUB"
150 END
```

Accessing Function and Control Keys using
CALL KEY 5- - - -

```
100 DISPLAY AT(3,3)ERASE ALL:
    "PRESS CONTROL KEY AND COMMA"
110 FOR DELAY=1 TO 600 ::
    NEXT DELAY
120 CALL KEY(5,K,S)
130 IF K=138 THEN PRINT "CONTROL
    AND COMMA PRESSED" ELSE 120
140 END
```

OR

```
100 DISPLAY AT(3,3)ERASE ALL:
    "PRESS FUNCTION 8"
110 FOR DELAY=1 TO 600 ::
    NEXT DELAY
120 CALL KEY(5,K,S)
130 IF K=6 THEN PRINT "FUNCTION
    8 PRESSED" :: GOTO 140 ELSE
    120
140 END
```

As you can see, the CALL KEY command gives you a great deal of control over the input you are accepting.

 #####
 # The Game Room #
 # # # # #
 #####

by Dan Hawes

Mystery Program

Here's an interesting program that has been adapted from the Bits and Bytes newsletter. It requires speech and Extended BASIC. That's all I can tell you. You'll have to find out what it does!

```

100 ! CATCH "IT"
110 ! FAMILY COMPUTING
120 ! FEBRUARY 1987
130 ! TRANSLATED BY W. BLOOD
160 CALL CLEAR :: FOR T=1 TO 12 :: CALL COLOR(T,16,1)::
NEXT T :: CALL SCREEN(5)170
A$="*" :: C$=CHR$(32):: DN$=
"X" :: UP$="E" :: N1$=C$&C$
:: F$=RPT$(F",16)
180 CALL CHAR(136,F$*FF7F3F
1F0F070301FF0000000000000000
):: CALL COLOR(14,7,1)
190 CALL CHARFAT(73,E$,84,B$
):: CALL CHAR:128,E$&B$&F$):
CALL COLOR(13,11,1):: I$=C
HR$(128)&CHR$(129)
200 P$=CHR$(138)&CHR$(138)&C
HR$(137)&CHR$(136):: M2$=N1$
&N1$
220 DISPLAY AT(2,3)ERASE ALL
:"Welcome to CATCH IT!"
230 DISPLAY AT(4,1):"Use the
<"UP$;"> and <"DN$;"> key
s"
240 DISPLAY AT(5,2):"to move
your ";CHR$(34);"catcher";C
HR$(34)
250 DISPLAY AT(6,6):"up and
down."
260 DISPLAY AT(8,1):"The goal
is to catch "&CHR$(34)&I$&
CHR$(34)
270 DISPLAY AT(9,1):"before
it hits the right"
280 DISPLAY AT(10,1):"border
of the screen."
290 DISPLAY AT(12,1):"Please
select the level"
300 DISPLAY AT(13,1):"of dif
ficulty you prefer."
310 DISPLAY AT(15,1):"Would
you like"
320 DISPLAY AT(17,1):"(1) ea
sy,"(2) moderate, or"
330 DISPLAY AT(19,1):"(3) ha
rd?";" "
340 ACCEPT AT(21,5)BEEP SIZE
(1)VALIDATE("123"):K$
350 DD=VAL(K$)*.5
360 RD=1 :: SC=0
380 DISPLAY AT(11,9)ERASE AL
L:"Get Ready!"
400 FOR DE=1 TO 1000 :: NEXT
DE :: CALL CLEAR
410 CALL HCHAR(3,3,130,28)
420 CALL HCHAR(23,3,130,28)
430 CALL HCHAR(2,2,136,30)
440 CALL HCHAR(24,2,136,30)
450 CALL VCHAR(2,31,136,30)
460 CALL VCHAR(2,2,136,23)
470 CALL VCHAR(3,3,130,21)
480 CALL VCHAR(3,30,130,21)
490 DISPLAY AT(1,1):"Round:"
:RD
500 DISPLAY AT(1,13):"Score:
":SC
510 X=10 :: 6A=4
520 Y=4 :: Z=INT(RND*16)+4 ::
6A=6A-(6A<100)
530 DISPLAY AT(X,24)SIZE(4):
P$
550 H1=Z :: HY=Y :: RANDOMIZ
E :: N=RND :: IF Y=22 THEN 6
90
560 Z=Z+DD*((N).5)-(N<.5))
570 Y=Y+1 :: Z=Z-DD*((Z<4)-(
Z>22))
580 DISPLAY AT(H1,HY)SIZE(2)
:N1$
590 DISPLAY AT(Z,Y)SIZE(2):I
$
610 CALL KEY(S,K,S):: IF S=0
THEN 550
620 K$=CHR$(K)
63: IF K$<>UP$ AND K$<>DN$ T
HEN 550
640 H2=X :: X=X-(K$=DN$)+(K$
=UP$)
650 X=X-(X<4)+(X>20)
660 DISPLAY AT(H2,24)SIZE(4)
:N2$
670 DISPLAY AT(X,24)SIZE(4):
P$ :: GOTO 550
690 DISPLAY AT(H1,HY)SIZE(2)
:N1$
700 IF Z-X>2.5 OR Z-X<0 THEN
Y=Y+2 :: GOTO 810
720 DISPLAY AT(Z,23)SIZE(1):
A$
730 FOR T=5 TO 6A
740 CALL COLOR(2,RND*10+6,RN
D*5+1)
750 CALL SOUND(250,T*25+110,
0):: SC=SC+10*DD
760 DISPLAY AT(1,19):SC :: N
EXT T
770 CALL COLOR(2,16,1)
780 DISPLAY AT(Z,23)SIZE(1):
C$
790 FOR DE=1 TO 200 :: NEXT
DE :: GOTO 520
810 HY=Y :: Y=Y+.5 :: DISPLA
Y AT(Z,Y)SIZE(2):I$
820 DISPLAY AT(Z,HY)SIZE(2):
N1$ :: IF Y>26 THEN 810
840 RD=RD+1 :: IF RD<4 THEN
380
860 DISPLAY AT(8,5)ERASE ALL
:"Sorry, you, missed "&CHR$(
34)&I$&CHR$(34)
870 DISPLAY AT(10,9):"Three
times!"
88: DISPLAY AT(12,1):"Your s
core was";SC;"points."
89: DISPLAY AT(15,1):"Would
you like to"
900 DISPLAY AT(17,1):"(1) p)
ay again:" at the same l
evel."
910 DISPLAY AT(19,1):"(2) se
lect a new level, or"
920 DISPLAY AT(20,1):"(3) qu
it/return to BASIC?"
930 DISPLAY AT(22,1):"Enter
your choice."
940 DISPLAY AT(24,3):" "
950 ACCEPT AT(24,5)BEEP SIZE
(1)VALIDATE("123"):K$
960 IF K$="3" THEN CALL CLEA
R :: END
970 IF K$="2" THEN CALL CLEA
R :: GOTO 290
980 GOTO 360
    
```

```

100 REM SAVE DSK1.GAME1
110 REM
120 REM Mystery Program
130 REM by Chris Schram
140 REM
150 REM Requires Memory Expans
ion
160 REM and Speech Synthesiz
er
170 REM
180 REM Runs in TI Extended
BASIC
19: REM or Console BASIC
200 REM with Editor/Assemble
r
210 REM or Mini-memory
220 REM
230 DATA 71,64,72,65
240 DATA 70,75,73,70
250 DATA 76,67,66,66
260 DATA 65,68,76,68
270 DATA 77,68,78,71
280 DATA 77,66,68,66
290 DATA 66,67,74,67
300 DATA 74,77,74,68
310 DATA 73,71,64,67
320 DATA 72,68,76,65
330 DATA 72,68,76,65
340 CALL INIT
350 CALL PEEK(-28672,A)
360 IF A<>96 THEN 460
370 FOR Z=1 TO 11
380 FOR X=1 TO 4
390 READ A
400 CALL LOAD(-27648,A)
410 NEXT X
420 CALL LOAD(-27648,64)
430 CALL LOAD(-27648,80)
440 NEXT Z
450 STOP
460 PRINT "You don't have a
Speech"
470 PRINT "Synthesizer attac
hed"
    
```

* Catch It! *

This month's game is one that was taken from the Kansas City 99'er newsletter. It runs in Extended BASIC. The game is fairly simple. You are a hand, trying to catch "IT" which moves randomly about the screen. You are allowed three misses before the game is over. There are three levels of difficulty, with one being the easiest and three being the hardest. ENJOY!

* Cheat Hints *

Here are some methods of "cheating" on various TI game modules. Some cartridge games like Moon Mine, Alpinar, and others go into a test mode when you press SHIFT B3B at the very beginning of the game. (The game title screen.) After you depress SHIFT B3B the game will ask you questions like what level would you like to skip to, etc... Also, there is a method in which one can "Time Warp" in Parsec. To do this type in the following procedure:

1. Crash one ship before firing.
2. Work up to the Bynites (the Bynites come before the asteroids).
3. Crash one ship after destroying each Bynite. If the asteroids start coming, crash all remaining ships.
4. Push "REDO" before the "GAME OVER" appears.
5. Crash one ship before firing. (again)
6. After the swoopers come the killer satellites, so be ready to break your joystick!

After the above procedure is done, you will advance from level 1 to level 4, skipping levels 2 and 3. Although the ground color will not change like it normally would, you will still be advanced to the forth level. (Sometimes this can also bring you down levels, so use it with caution!)

HAVE FUN!

(Continued on page 4)

Electronic Datebook

This program ELECTRONIC DATEBOOK may be helpful in keeping track of appointments, birthdays, 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 using XBasic. When you run the program you will see the Main Menu, which asks whether you want to (1) View the Calendar; (2) Load the data file; (3) Save the data 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 re-displayed each date for which you have a 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 Data File". Pick DSK1, or other choice for the device you are using. If you are using a disk, be sure you have at least 185 sectors free on the disk as the entire years memos will be stored on that disk. It is best to use a blank disk or a new blank tape. The recording takes several minutes.

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

If you can make use of this program you will need to make some changes for 1988 and subsequent years. Change the year in the lines: 130, 240, 360, 370, and 500. Also change the day of the week number appearing before each month in line 180. For instance the first of January in 1988 will be on a Friday the 6th day of the week. Thus you would want to insert a 6 prior to January. You'll need to do that for all the other months also.

(Continued from page 3)

* SORCERER HINTS II *

This is a continuation of Sorcerer hints from last months newsletter.

6. Do you not have enough zorkoids to go everywhere you want?? Try searching the troll at the bridge after you give him the money.

7. Missing a bat spell?? After pulvering the river, go find the hidden cave, and get the pile of bat guano. Put the bat guano in the cannon in the gun emplacement and try getting a scroll from the cannon.

```

100 !*****
110 !* ELECTRONIC *
120 !* DATEBOOK *
130 !* -- 1987 -- *
140 !*****
150 !AUTHOR:ROY TAMASHIRO
160 !USES EXTENDED BASIC
170 DIM D$(11,31,4)
180 M$="5Jan31Feb28Mar31Apr30
pr306May312Jun304Jul317Aug31
3Sep305Oct311Nov303Dec31"
190 GOSUB 500 :: M1$="Author
:Roy Tamashiro" :: R=3 :: GO
SUB 490
200 DISPLAY AT(8,7):"1 to Vi
ew Calendar":TAB(7):"2 Lo
ad Date File":TAB(7):"3 S
ave Date File":TAB(7):"4
Exit"
210 DISPLAY AT(19,5):"YOUR C
HOICE (1-4):" :: ACCEPT AT(1
9,24)SIZE(1)VALIDATE("1234")
:C$
222 GOSUB 500 :: ON VAL(C$)G
O 230,400,400,400
230 DISPLAY AT(4,1):"Enter M
onth (1-12):" :: ACCEPT AT(4
,21)SIZE(2)VALIDATE(DIGIT):M
:: M=M-1 :: IF M<0 OR M>11
THEN 230
240 GOSUB 500 :: M1$=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)<>" I
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)S:TE(1)V
ALIDATE("123"):C$ :: GO 350
10 :: ON VAL(C$)GOTO 120,350
,190
320 DISPLAY AT(15,1):"Enter
Date:(1-";STR$(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 W10="Load Date File" ::
R=3 :: GO 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 M10="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(W10)
/2):W10 :: RETURN
500 CALL CLEAR :: W10="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
    
```

8. Instead of trying to take the scroll with you from the top of the hovel, (inside the glass maze) drop it down the small hole, then gaspar yourself and die. Go back to the stone room and look in the chimney.

9. If you can't seem to hit anything in the gallery in the amusement park, try taking a potion.

10. Missing the potion needed in step 9?? (The potion is in an aqua vial) Lower the flag in the parade ground, and search the flag. You should find the vial needed.

Easy to Use Mail List

This mail listing program is very easy to use and was designed by George Steffen of the LA 99ers. You can use TI Writer to make up your mail list in a Display Variable 80 format. Then with this program you will be able to either print out either labels or a list.

The program is written for Epson compatible printers, however I'm sure with a little work in lines 130 thru 160 that it could be made to work with other printers.

The program has provision for entering label height and width and single or double line feeds. Of course, with one inch labels and double line feeds you would be restricted to three lines per label. The program will discard any portion of a line that will not fit on the size label you have entered.

With the program you can print labels from files prepared in the edit section of TI Writer. You must cancel Word Wrap and use the hollow cursor. The first line in the file is reserved for your file title and will not be printed. You must also put in a New Page (Control 9) before each label. For example when starting a new file line one would be your File Name, line two would be a New Page Control and the first line of your first label would be on line three. If you wish to add any printer controls to your data you may do so between the File Title and the first New Page entry.

The program includes a stop label printing feature so that entries may remain on the list, but labels will not be printed out. This would be handy if you had a

mailing list with delinquencies. The labels would print out for current members, but the others would remain on your file but not print out. If you use the listing feature of this program the entire file would print out.

(See the sample below to use this feature)

The program as written runs in Extended Basic, but you can run it in Basic with a simple change in line 660. Change LINPUT to INPUT in this line.

You can use leading spaces, trailing spaces and commas when using the Extended Basic version but you can't use any of these in the Basic version.

The PC in the example below indicates New Page and Carriage Return.

Membership Test List

```
PC
Jane Doe
123 E. Main Street
Anytown, CA 98765
PC
John Q Public
P.O. Box 75
Los Angeles, CA 90001
PC
END END END END
Members who have not renewed
PC
Richard Roe
3298 Elm Street
Old Town, NY 12345
PC
```

```
100 REM LIST LABEL
110 FE= SEE INSTRUCTIONS
120 DATA PRINT LABELS,PRINT
LISTS,END SESSION
130 FL$=CHR$(27)&"C"
140 CT$(1)=CHR$(27)&"E"&CHR$(
27)&"G"
150 CT$(2)=CHR$(15)
160 RST$=CHR$(18)&CHR$(20)&C
HR$(27)&"F"&CHR$(27)&"H"&FL$
&CHR$(66)
170 FOR I=1 TO 3
180 READ M$(I)
190 NEXT I
200 CALL CLEAR
210 PRINT TAB(8);"MAIL PRINT
ER"
220 FOR I=1 TO 3
230 PRINT :STR$(I);", ";M$(
I)
240 NEXT I
250 PRINT
260 INPUT "CHOICE? ";S
270 IF (S(1)+(S>3) THEN 260
280 ON S GOTO 290,440,990
290 PRINT
300 INPUT "LABEL HEIGHT (IN)
? ";L
310 L=L#6

320 IF L=INT(L) THEN 350
330 PRINT "ERROR IN LABEL HE
IGHT."
340 GOTO 300
350 PRINT
360 INPUT "LABEL WIDTH (IN)?
";C
370 C=C#10
380 IF C<80 THEN 410
390 PRINT "ERROR IN LABEL WI
DTH."
400 GOTO 360
410 PRINT : "! SINGLE LINEFE
ED";"2. DOUBLE LINEFEED"
420 INPUT "CHOICE? ";D
430 IF (D<1)+(D>2) THEN 420 E
LSE 460
440 L=66
450 C=132
460 PRINT
470 INPUT "FILE NAME FOR YOU
R PRINTER? ";P$
480 OPEN #2:P$,DISPLAY ,VARI
ABLE C,OUTPUT
490 PRINT #2:FL$;CHR$(L);CT$
(S);
500 PRINT : "FILE(S) TO BE PR
INTED?";"NULL ENTRY WILL TER
MINATE": ;

510 I=1
520 INPUT STR$(I)&". ";F$(I
)
530 IF F$(I)="" THEN 570
540 PRINT
550 I=I+1
560 GOTO 520
570 PRINT : :TAB(11);"PRINTI
NG"; ;
580 L=0
590 FOR J=1 TO I-1
600 PRINT STR$(J);", ";F$(J
);
610 E=0
620 K=0
630 OPEN #1:F$(J),DISPLAY ,V
ARIABLE 80,INPUT
640 INPUT #1:L$
650 IF EOF(1) THEN 740
660 LINPUT #1:L$
670 IF ASC(L$)>127 THEN 740
680 IF ASC(L$)<>12 THEN 720
690 ON S GOSUB 860,880
700 K=K+1
710 GOTO 650
720 ON S GOTO 900,970
730 IF E=0 THEN 650
740 K=K-1
750 L=L+K

760 IF S=1 THEN 780
770 PRINT #2: :TAB(7);K;"ENT
RIES": ;
780 PRINT K;"ENTRIES": ;
790 CLOSE #1
800 NEXT J
810 PRINT :TAB(6);"PRINTING
COMPLETED";L;"TOTAL ENTRIES"
: ;
820 IF S=1 THEN 840
830 PRINT #2: : :TAB(7);"Tot
al Entries";L;RST$
840 CLOSE #2
850 GOTO 220
860 PRINT #2:L$
870 RETURN
880 PRINT #2
890 RETURN
900 IF SEG$(L$,1,3)<>"END" T
HEN 930
910 E=1
920 RETURN
930 PRINT #2:L$
940 IF D=1 THEN 960
950 PRINT #2
960 RETURN
970 PRINT #2:L$;" ";
980 RETURN
990 END
```

PRINT USING

One of the more obscure statements available with TI Extended BASIC is one called PRINT USING. Even more obscure is the fact that this statement can be used to format variables and constants that will be dumped to your printer. The Extended BASIC manual, on page 150 shows several examples of how PRINT USING can be used to format data for screen display, but not a word of how to do the same with open files. It can be done, and is much more powerful than you may realize.

Any discussion of PRINT USING will require an understanding of the IMAGE statement, so if you are not familiar with it, you better brush up on it first. The PRINT USING statement uses IMAGE in one of two ways, either with a string expression, or a line number reference. I prefer the latter, as it allows for more flexibility, but since these different methods are explained in the manual, I will limit this to a few simple examples that are not shown there.

```
100 TCOST=19.95 110 IMAGE ##.## 120 OPEN
#1:"FIO" 130 PRINT #1,USING 110:TCOST
```

Running this sample program will effectively show how the PRINT USING statement will work with an open file. Of course, there are many other variations of IMAGE that can be used, so experiment with them and watch how it performs when line 130 dumps it to the printer. Shown below are a few more examples for use with an open file.

```
110 IMAGE "###.## ##.##" 130 PRINT #1,USING
110: COST1,COST2
```

This IMAGE statement will allow you to print two (or more) variables at a pre-determined spot on the same line. The length of the string expression in the IMAGE statement can be as long as you wish, up to the limit of an Extended BASIC line.

```
110 IMAGE "##### ##.##" 130 PRINT #1,USING
110:"TOTAL COST", TCOST
```

This version shows how you can format the printed line for string data as well as numerical data. A string variable could be used in place of the string constant, as below.

```
105 A$="TOTAL COST" 110 IMAGE
"##### ##.##" 130 PRINT #1,USING
"###.##":TCOST
OR 130 PRINT #1,USING "##### ##.##
":A$,TCOST
```

A few other points to remember include the fact that IMAGE and PRINT USING can be used to round off calculated variables. A single string expression such as "#####.##" will round off and decimal align numbers as small as 0.01 up to 999999.99 and print the number at any designated location. This function could save many hours of algorithm development for accomplishing the same thing. So, in the long run, the PRINT USING statement is one that any programmer should be very familiar with and use as much as possible.

Tutorials Offered

Jim Peterson from Tigercub Software has sent us three disks full of his Tips. He tells us in his letter that he has gathered many of his tips and tutorials and offered them to the various user groups around the country. These have been placed in the club library and you can have any or all of them for the usual copying fee.

The disks just arrived in time to report on them in this newsletter and I have not had time to look at them completely. Basically they are good sound tutorials that will help you understand how your computer works. All you have to do is call up the particular area of interest and print it out. Or if you don't have a printer you can look at the information on the screen.

This has to be one of the best things that has come our way for a long time. Plan to get your copies at the next meeting.

Slash Zero

I have decided that from now on as long as I am the newsletter editor that I will use a (SLASH-ZERO) when I list programs. Even though my Gemini Printer does a pretty good job of distinguishing between O's and 0's I think you will find it easier when you type programs in.

I am also listing this little program in the Bulletin so you can use it in your own work.

Type it in, and save it to a file such as (/ZERO). Then before you send anything to the printer, run the program and everything that the printer prints will slash zeros as long as you leave the printer turned on.

I don't like to see slashed zeros in ordinary printing but when it comes to reading program lines, I think they are helpful.

```
100 ON BREAK NEXT :: CALL CL
EAR :: CALL INIT
110 OPEN #1:"FIO"
120 PRINT #1:CHR$(27);CHR$(4
2);CHR$(0)
130 PRINT #1:CHR$(27);CHR$(4
2);CHR$(1);CHR$(48);CHR$(0);
CHR$(92);CHR$(34);CHR$(0);CH
R$(81);CHR$(8);CHR$(69);CHR$(
0);CHR$(34);CHR$(29)
140 PRINT #1:CHR$(27);CHR$(3
6);CHR$(1)
150 CLOSE #1 :: CALL PEEK(2,
A,B):: CALL LOAD(-31804,A,B)
```

TIGERCUB SOFTWARE

Tigercub Software has released Nuts & Bolts #3, containing another 140 subprograms in MERGE format. Contents include 19 screen character fonts, etc., 17 screen display routines, 6 screen formatting, 8 plotting, 6 date, 10 input and accept, 9 string handling, 15 file handling, and 9 miscellaneous routines. The 11 pages of documentation contain a

programming example to demonstrate the use of each routine.

The three Nuts & Bolts Disks now provide a total of 348 subprograms which programmers can merge into their own programs and use. It's like having 348 CALLS available in Extended Basic. The price of all three of these disks has been reduced to \$15.00 each, postpaid.

Send for their catalog. The \$1.00 charge is refundable on your first order.

TIGERCUB SOFTWARE
156 Collingwood Avenue
Columbus, OH 43213
Phone 614/235-3545

Basic Programming

Most of us have read many articles about programming. Some good ones and some not so good. Yet there are still some of us who can always use more information on good fundamental programming information.

This article is directed at that group who still are looking for fundamentals of programming.

As you know there are two basic languages that you can use. Console Basic which is built into your computer and Extended Basic which uses a module and gives the programmer many functions and commands not available in Console Basic.

Regardless of which language you use, there are some basic steps that need to be taken to write a useful program. One thing you must keep in mind is that writing a program will task your powers of reason as well as your ability to think logically.

Many of you may think that programming is a special gift or talent which only a few select people possess. Not completely true, because what really makes a good programmer is the ability to take a problem or task and write a set of instructions and commands to make their computer solve the problem or perform the task at hand.

There is a definite flow of events which must occur in sequence in order to end up with a solution to the current problem. One of the first steps is to identify and define the problem as briefly as possible. The next thing that we must do is outline the solution as to the best approach to the solution. The word "Algorithm" is often used when speaking of programming. This word is used to refer to a specific method of solving a certain kind of problem. If you were to ask a dozen programmers to solve a given problem, you would probably get a dozen different algorithms depending on that programmer's approach to problem solving. In other words there are no hard and fast rules as to the way a problem may be solved with your computer.

The next step is to write the program, which is not as hard as it may seem once you have gone through the previous steps. One of the things to use when stumped by a particular area of the process of problem solving, is to look at other programs for segments which may help in my final solution. Two favorite sources are the TI Learning Basic and Teach Yourself Extended Basic lesson programs. Many routines to do some of the things you may want to do are contained in these lessons.

There are also a number of books available that were written for the TI and if you will look through these books you can find many of the solutions and routines to use in your programs.

Suppose we have a simple task such as adding two numbers and obtaining the result. We must first assign the given values, perform the necessary mathematical operation, and then be able to see the results of that operation. The following is but one approach:

```
100 REM ADD TWO NUMBERS
AND PLACE RESULT ON SCREEN
110 INPUT A,B
120 LET C=A+B
130 PRINT C
140 END
```

Line 100 is a remark statement. The computer ignores all information contained in a REM statement, but this information is often essential to the documentation of the program. It allows you and others to see what a given set of commands or statements do in any given instance. Line 110 allows you to input the two values and assigns the values to two variables named A and B. Line 120 adds the two numbers and places the result in a variable named C. The LET statement is used for convenience mostly, but is not required. Line 130 Prints the result on the screen for you to see and Line 140 tells the computer that the program has ended.

Another task you may at one time wish to accomplish is the squaring of a set of numbers. This can be done as follows:

```
100 REM SQUARE THE NUMBERS
AND PRINT RESULTS
110 N=N+1
120 S=N 2
130 PRINT N,S
140 IF N<99 THEN 110
150 END
```

Line 100 is the familiar REM Statement. Line 110 adds one to the value of the variable N. Line 120 Squares the number N and assigns the result to the variable S. Line 130 prints to the screen the value of N and it's square S. Line 140 tells the computer that if the value of N is less than 99 then repeat the process on the next number determined by N=N+1. This value would be the highest number you wish to be squared by the program. Line 150 is the old END statement to stop execution. On the TI the end statement is not necessary but it is good practice. Note that in lines 110 and 120 we omitted the LET statement.

One of the better tools in good programming and the solution of problems is the use of flowcharts. You can find information on flow charts in many computer books. This article is necessarily quite basic but it is hoped that it will give many of our newer members the incentive to try some programming on their own.

P.O. Box 15037
Portland, OR 97215



DISCLAIMERS: The PUNN User's Group is not affiliated with or sponsored by TI and has no relationship with them, implied or otherwise.

Mention of a company or product is not an endorsement of that company or product.

We are not a subsidiary or branch of any other User's Group and any relationship we may have with other groups is on the basis of equals.

ALL GENERAL MEETINGS ARE HELD
ON THE FIRST TUESDAY OF EACH
MONTH, AT THE PGE BUILDING
3700 SE 17TH. PORTLAND, OREGON

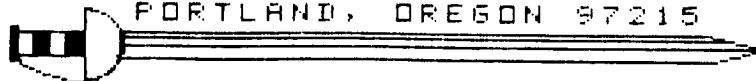
!! NEXT MEETING DATE !!
JUNE 2ND. 1987

THE **PUNN** NEWSLETTER

Wordplay

P.O. BOX 15037

PORTLAND, OREGON 97215



POINTING THE WAY FOR
USERS OF TI'S 99/4 COMPUTERS

JUNE 1987-VOLUME VI ISSUE 6