

SGV 99/4 USERS' GROUP

PRINTOUT



VOL. 2 NO. 7

PUBLISHED BY THE SAN GABRIEL VALLEY 99/4 USERS GROUP

MINUTES...

The meeting of August 1st was short, since most of the members wanted to get home and watch the Olympics. Topics briefly discussed were:

- 1) The enhancement for TI Writer and Multiplan. TI Writer can now provide true lower case letters on the screen and no longer needs the form feed when using the Formatter. Other files will allow some printer defaults.
- Multiplan is faster and has auto repeat when moving the cursor around the screen.
- 2) Source code for Forth. See Paul Schippnick for details.
- 3) Super Debugger - a program that a few members have gotten.
- 4) the picnic of August 18th.
- 5) Rumors regarding the International User Group lawsuit against the Atlanta UG. (As if TI's pulling out of the market wasn't enough of a stress - now we're trying to do ourselves in!)
- 6) Bugs in the CorComp products on the market. If you've bought one and are having problems, call the distributor and get the solution, or get on the list so there won't be a delay in your getting the improvement when the bug is exterminated.
- 7) The Olympics.

The Thursday meeting was taken up by copying the Super Debugger for members and answering questions about the TI.

There was no executive meeting.

(THE CLUB THANKS TI FOR THEIR CONTINUED SUPPORT IN THE FORM OF PROGRAMS AND TIPS FOR THE 99 - THE 'CARE PACKAGES' GIVE US IDEAS FOR CLUB OBJECTIVES.)

MAINTAINING YOUR DISK DRIVE

Do not keep your drive next to the CRT, a refrigerator or any electrical equipment that would generate a magnetic field. (The house of the future, with a computer doing the menu planning and cooking is still in the future.)

If your area is subject to blackouts and brownouts, it would be wise to invest in a surge protector. Also, if you are not going to use your computer daily, unplug the transformer.

Keep track of your computer use; every few hundred hours, clean your disk drive heads. Be delicate - use isopropyl alcohol (not rubbing alcohol) and lint free cloth or a foam applicator. If you use a kit, buy the 'wet' type.

Check the drive for loose screws, chips, dust, etc.; like you do for the stereo and TV. Remember to be delicate - drives are the piece of equipment most often repaired.

Do not force a disk into the machine - you could damage the heads.

When moving your drive, put either the cardboard 'disk' that came with the machine, or put an error-ridden disk in the drive to protect the heads.

On the TI, close the door of drive when not in use, to keep out dust.

COMPUTERESE...

For those of you who are new to the computer culture, here are a few terms. Like any language, you become fluent with use!

Network - An interconnection of computer systems, terminals & communication facilities. Usually done with modems.

Program listing - A printout that lists the source language statements and contents of a program. What you use when debugging a program or when you want to study to decipher how someone else wrote a program.

Cursor - A pointer that shows the location on the screen your input will affect.

Mouse - A cursor pointing device which is operated by moving it on a surface. The Macintosh uses one.

LASER - as in laser printer. A device that emits a beam of coherent light that forms the image that is transferred to the paper.

Non-impact printer - Printing is not result of mechanical impacts, like the laser, thermal, electrostatic or photographic printer.

Impact printer - A printer which strikes the paper, like a typewriter.

(Dot) Matrix printer - Each character is represented by a pattern of dots.

Line printer - A device that prints a line of characters as a unit.

CRT - A television-like picture tube used in visual display terminals like the TI.

Hardware - The physical equipment like the console, drive, printer, etc.

Software - Programs, procedures, rules and documentation that pertains to the operation of a computer system.

Firmware - A computer program or software stored permanently in the computer.

CPU - Central processing unit. The part of a computer that interprets the program and does arithmetic and logical operations.

Peripherals - Any equipment that communicates with the CPU. The drive, printer, recorder, etc.

Binary - A base 2 numbering system - just as decimal is a base 10 numbering system - which is used by computers because it can represent an OFF or 0 condition and an ON or 1 condition.

Bit - A single binary digit, either 0 or 1.

Byte - Eight bit binary number that can represent a number, a character, a memory address, etc.

K - Usually an abbreviation of kilo (1000). In computers, it is 2 to the 10th power or 1024 bits or bytes.

Modem - Translates the computer's binary language into tones that the telephone can transmit. One reconverts incoming tones into binary language that the receiving computer can understand. Short for modulator-demodulator.

Interface - Hardware or software used to connect two devices.

RS-232 - Standard interface.

That's it for this month. To learn more, come to the meetings. (Thursdays especially)

USING THE TI WRITER WITH A GEMINI 10X

Following are some of the codes you use to get what you want from the 10X. These are not by any means all of them; only the ones I use the most.

First, to access the codes, you use 'Control U'. Your cursor will change from a solid black square or open square, depending if you're using Formatter or Editor, to a black line. A capital letter (B) in a command means that is what you should see on the screen. For example, to print in pica, a size of print, the command is ESC B 1. What you do on the computer is strike Control U, Function R, Control U, Shift A, Control U, then Shift A again. This is what you get on the screen:

The most important thing to do: Get page 146 of your TI Writer manual and copy it. Put it where you can see it when using TI Writer. Mine is so dog-eared, I need to make another copy and cover it!

Standard print: ESC 5 - CTRL U, FCTN R, SHIFT E
Italic Print: ESC 4 - CTRL U, FCTN R, SHIFT D
Pica: DC2 - CTRL U SHIFT R (another way)
Condensed: SI - CTRL U, SHIFT O
Enlarged: SO - CTRL U, SHIFT N
Double strike: ESC G - CTRL U, FCTN R, CTRL U, SHIFT G
Underline: ESC - 1 - CTRL U, FCTN R, CTRL U, SHIFT SLASH, CTRL U, SHIFT A
Disregard paper out: ESC B - CTRL U, FCTN R, CTRL U, B

The ASCII Codes are the key to transliterating the TI commands to the 10X or any other printer's commands. It is rather boring work to look up the commands which give you what you want. However, it is worth it - you get maximum use of your printer's capabilities.

THE PICNIC...

Of course it rained the day of the picnic. A few hardy souls braved the scattered sprinkles and talked computers in the great outdoors. We had plenty of food, and held a drawing for 2 solar calculators (donated by member JACK SEIDEL), and Donna Roquemore's version of the 10 commandments for the computer, framed. Since all of the officers showed, and two members, naturally, the officers won the prizes! Cheryl won the commandments, Floyd won a calculator, as did Tom's children. Maybe next year we should hold the picnic in April!

THE LIBRARY...

First, those of you who want the Debugger, the Forth Source Code, and/or the enhancements for Multiplan & TI Writer, must bring your own disks (one for each program except the source code which requires two disks) to the Thursday meetings only. They are only available at the meeting. There is a \$5 charge that covers our costs for copying the documentation. Orders for other programs in the Library will be as usual - order one month and receive the next.

Now for the programs:

The first two are from PAUL SCHIPPNICK and are two more versions of a screen dump. See Paul for further explanations.

```
>32728 REM 8407101854 PRO-SC
REEN DUMP\3 MODULE VERSION--
-----BY P.E.SCHIPPNICK
>32729 \$$="123456789ABCDEF"
>32730 \\\=159
>32731 CALL PEEK(24577,\)
>32732 IF (\=0)+(\=255)THEN 3
2734
>32733 \\\=143
>32734 OPEN #1:"PI0.CR"
>32735 PRINT #1:CHR$(27)&"T"&
"16"
>32736 FOR I=32 TO 1 STEP -1
>32737 PRINT #1:CHR$(27)&"S"&
"0192";
>32738 FOR I=1 TO 24
>32739 CALL GCHAR(I,1,\)
>32740 IF (\<32)+(\>\\)THEN
32757
>32741 CALL CHARPAT(\,\$$)
>32742 IA=POS(\$,SEG$(\$,1,1
),1)*16+POS(\$,SEG$(\$,2,1
),1)
>32743 IB=POS(\$,SEG$(\$,3,1
),1)*16+POS(\$,SEG$(\$,4,1
),1)
>32744 IC=POS(\$,SEG$(\$,5,1
),1)*16+POS(\$,SEG$(\$,6,1
),1)
>32745 ID=POS(\$,SEG$(\$,7,1
),1)*16+POS(\$,SEG$(\$,8,1
),1)
>32746 IE=POS(\$,SEG$(\$,9,1
),1)*16+POS(\$,SEG$(\$,10,1
),1)
>32747 IF=POS(\$,SEG$(\$,11,
1),1)*16+POS(\$,SEG$(\$,12,
1),1)
>32748 IG=POS(\$,SEG$(\$,13,
1),1)*16+POS(\$,SEG$(\$,14,
1),1)
>32749 IH=POS(\$,SEG$(\$,15,
1),1)*16+POS(\$,SEG$(\$,16,
1),1)
>32750 PRINT #1:CHR$(IA)&CHR$(
IB)&CHR$(IC)&CHR$(ID)&CHR$(
IE)&CHR$(IF)&CHR$(IG)&CHR$(
IH);
>32751 NEXT I
>32752 PRINT #1:CHR$(13)&CHR$(
10)
>32753 NEXT J
>32754 PRINT #1:CHR$(27)&"A"
>32755 CLOSE #1
>32756 END
```

```

>32757 JA=0
>32758 JB=0
>32759 JC=0
>32760 JD=0
>32761 JE=0
>32762 JF=0
>32763 JG=0
>32764 JH=0
>32765 GOTO 32750

>32693 REM 8406132349*SCREEN
DUMP--BY P.E.SCHIPPNICK
>32694 DIM \$(23)
>32695 OPEN #1:"PIO.CR"
>32696 FOR I=0 TO 23
>32697 CALL PEEKV()*32,\1,\2,
\3,\4,\5,\6,\7,\8,\9,\0,\A,\
B,\C,\D,\E,\F,\G,\H,\I,\J,\K
,\L,\M,\N,\O,\P,\Q,\R,\S,\T,
\U,\V)
>32698 \$(I)=CHR$(\1)&CHR$(\2
)&CHR$(\3)&CHR$(\4)&CHR$(\5)
&CHR$(\6)&CHR$(\7)&CHR$(\8)&
CHR$(\9)&CHR$(\0)&CHR$(\A)&C
HR$(\B)&CHR$(\C)&CHR$(\D)&CH
R$(\E)&CHR$(\F)&CHR$(\G)&CHR
$(\H)&CHR$(\I)&CHR$(\J)&CHR$(
\K)&CHR$(\L)&CHR$(\M)&CHR$(
\N)
>32699 \$(I)=\$(I)&CHR$(\O)&C
HR$(\P)&CHR$(\Q)&CHR$(\R)&CH
R$(\S)&CHR$(\T)&CHR$(\U)&CHR
$(\V)
>32700 NEXT I
>32701 PRINT #1:CHR$(27);"A";
CHR$(8)
>32702 FOR I=32 TO 1 STEP -1
>32703 PRINT #1:CHR$(27);"K";
CHR$(192);CHR$(0);
>32704 FOR J=1 TO 24
>32705 I=ASC(SEG$(\$(I)-1),(I
))
>32706 I=\*(\>125)-127*(\<126
)
>32707 CALL PEEKV(1016+(\-127
)*8,JA,JB,JC,JD,JE,JF,JG,JH)
>32708 GOSUB 32716
>32709 PRINT #1:CHR$(JA)&CHR$(
JB)&CHR$(JC)&CHR$(JD)&CHR$(
JE)&CHR$(JF)&CHR$(JG)&CHR$(J
H);
>32710 NEXT J
>32711 PRINT #1:CHR$(13);CHR$(
10)
>32712 NEXT I
>32713 PRINT #1:CHR$(27);"@"
>32714 CLOSE #1
>32715 END

>32716 \I=JA
>32717 GOSUB 32741
>32718 JA=\@
>32719 \I=JB
>32720 GOSUB 32741
>32721 JB=\@
>32722 \I=JC
>32723 GOSUB 32741
>32724 JC=\@
>32725 \I=JD
>32726 GOSUB 32741
>32727 JD=\@
>32728 \I=JE
>32729 GOSUB 32741
>32730 JE=\@
>32731 \I=JF
>32732 GOSUB 32741
>32733 JF=\@
>32734 \I=JG
>32735 GOSUB 32741
>32736 JG=\@
>32737 \I=JH
>32738 GOSUB 32741
>32739 JH=\@
>32740 RETURN
>32741 \@=0
>32742 IF \I<128 THEN 32745
>32743 \@=\@+1
>32744 \I=\I-128
>32745 IF \I<64 THEN 32748
>32746 \@=\@+2
>32747 \I=\I-64
>32748 IF \I<32 THEN 32751
>32749 \@=\@+4
>32750 \I=\I-32
>32751 IF \I<16 THEN 32754
>32752 \@=\@+8
>32753 \I=\I-16
>32754 IF \I<8 THEN 32757
>32755 \@=\@+16
>32756 \I=\I-8
>32757 IF \I<4 THEN 32760
>32758 \@=\@+32
>32759 \I=\I-4
>32760 IF \I<2 THEN 32763
>32761 \@=\@+64
>32762 \I=\I-2
>32763 IF \I<1 THEN 32765
>32764 \@=\@+128
>32765 RETURN

```

CONTINUED ON PAGE 6

***** RELATIONAL OPERATORS *****

an article on TI 99/4(A) programming
By: Curtis Garcia
10202 Forum Park Drive #310
Houston, Texas 77036
(713)774-6881

The *Relational Operator (RO)* is a powerful programming tool which is not well explained by TI in the 99/4(A) manual. I will attempt to explain what a *RO* is and it's possible applications in this article.

A *RO* is a relational expression such as $X=5$ or $Y>A$. I'm sure you've seen this type of expression in *IF-THEN-ELSE* statements. The purpose of this article is to show new applications for this type of statement, and ways to conserve program space by using *RO's*.

The first thing that must be understood is how the computer interprets it in a program. The computer effectively replaces a *RO* expression with either a 0 if the expression is FALSE, or a -1 if the expression is TRUE. It is this fact which gives the *RO* the potential to do so much for us in a program, as I will demonstrate in this article.

The simplest form of the *RO* is the simple IF statement:

```
200 IF X=5 THEN 390
```

Then there is the OR statement (for TI Basic):

```
200 IF (X<1)+(X>9)THEN 390
```

And, there is the AND statement (for TI Basic):

```
200 IF (X=5)*(Y<5)THEN 390
```

REPRINTED FROM:
Hustan Users Group
Newsletter

The above expressions are all simple *RO's*. In each case, the computer will evaluate the *RO* and, if the result is other than zero, perform the command that follows the *RO*. If the evaluation results in a value of zero, control passes to the next line in the program. As you can see in the OR statement example, there are two relationships expressed, with a + between them. This causes the results of the evaluations to be added together. If the total is not zero (one or more of the expressions are TRUE), the command that follows is executed.

In the AND statement example, the results of the evaluations are multiplied. Thus, if either expression were FALSE, the total expression would also be FALSE $(0)*(-1)=0$ and control would pass to the next line. If both of these were TRUE, the result of the evaluation is not zero $(-1)*(-1)=1$ and the command that follows the *RO* would execute.

We've looked at the simplest application of the *RO* now. In Extended Basic you should use the OR and AND commands that are provided, since they are more space efficient than the methods shown above. In the next section I will show applications of *RO's* for Basic or Extended Basic with large potential savings of program space outside of the *IF-THEN-ELSE* statement.

The fact that the computer evaluates and equates *RO's* as a 0 if FALSE, and a -1 if TRUE, has many other applications when programming either in TI Basic or Extended Basic. Consider the following equivalent statements:

```
190 X=X+1
200 IF X<5 THEN 220
210 X=0
```

The above can be replaced with the statement below using a *RO*:

```
190 X=X+1+5*(X=4)
```

The above *RO* looks at the value of *X* BEFORE BEING MODIFIED BY THE CURRENT LINE. So, if *X* is 4, the *RO* is TRUE(-1), and the 5*(X=4) evaluates to a -5, which is added to the first part of the expression; X=X+1. Thus, X=4+1+(-5)=0.

This is the power of the *RO*, that you can set up relationships almost anywhere in your program, that can save memory space. The *RO* cannot be used in DIM statements or OPTION BASE statements, but may be used in all other programming applications where a numeric value is expected. These include FILE NUMBERS, CALL COLOR, HCHAR, VCHAR, DISPLAY, RECORD NUMBERS (in a file), FOR-NEXT LOOPS, ect.

A very graphic example of the space savings possible with *RO's* is shown in the following equivalent examples. In each case, a CALL KEY statement is used to detect any of the four arrow keys, which are being used to set the *X* and *Y* values. The *X* and *Y* values represent the row and column locations of a graphic character, with its travel limited to rows 1 thru 24 and columns 3 thru 30.

200 CALL KEY(0,KEY,STATUS)	310 IF X>0 THEN 500
210 IF STATUS=0 THEN 200	320 X=1
220 IF KEY=68 THEN 260	330 GOTO 500
230 IF KEY=69 THEN 300	340 Y=Y-1
240 IF KEY=83 THEN 340	350 IF Y>2 THEN 500
250 IF KEY=88 THEN 380 ELSE 200	360 Y=3
260 Y=Y+1	370 GOTO 500
270 IF Y<31 THEN 500	380 X=X+1
280 Y=30	390 IF X<25 THEN 500
290 GOTO 500	400 X=24
300 X=X-1	410 GOTO 500

The above can be replaced with the following equivalent statements:

```
200 CALL KEY(0,KEY,STATUS)
210 IF (KEY<>68)*(KEY<>69)*(KEY<>83)*(KEY<>88) THEN 200
220 Y=Y+(KEY=68)*(Y<30)-(KEY=83)*(Y>3)
230 X=X+(KEY=88)*(X<24)-(KEY=69)*(X>1)
240 GOTO 500
```

The above example uses the relational AND to check both the KEY and the value of the variable (either *X* or *Y*) BEFORE MODIFICATION; this prevents the variable from being modified to a value outside the desired range. You will notice that if either of a pair of *RO's* with a * between them is FALSE, the result is zero, so no change is made to the variable. Remember, when using the relational AND, if TRUE, the value is +1. If using the relational IF, the value is -1 if TRUE, so watch the signs you place in front of the *RO's* to get the desired results.

The last one is from Mike Young. It is a versatile program that can be used for graphics, as a subroutine in a word-processing program, etc. When you run it, let your imagination run wild.

The program was originally written to scroll a message across the screen, but I got carried away. If you don't want colors then delete lines 11,12,&13. To really get the effect of scrolling delete lines 18&21 and let B=3 in lines 19&20.

```
10 A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
11 FOR A=1 TO 14
12 CALL COLOR(A,A,A)
13 NEXT A
14 CALL CLEAR
15 FOR A=1 TO 28
16 C$=SEG$(A$,1,A)
17 D$=SEG$(A$,29-A,A)
18 FOR B=1 TO 23 STEP 2
19 DISPLAY AT(B,28-A):C$
20 DISPLAY AT(B+1,1):D$
21 NEXT B
22 NEXT A
23 FOR DELAY=1 TO 50
24 NEXT DELAY
25 GOTO 14
```

NEW MEMBERS...

Prisco J Serrano
Edwin Soledad
Laurel Webb
Jack Seidel

RENEWING MEMBERS...

Mike Archer
Joe King
Calvin Caldwell

DISCLAIMER:

The SGV 99/4 User's Group is not affiliated with or sponsored by TI or by any other User's Group. We have no relationship with TI, implied or otherwise.

Mention of a company or product in the PRINTOUT is not an endorsement. The opinions expressed are those of the individual and not necessarily that of the User's Group.

SGV 99/4 User's Group
1008 Dore Street
W. Covina, CA 91792

President:
Tom Fadilla

Vice-President:
Bob Roquemore

Treasurer:
Floyd Wilson

Secretary & PRINTOUT Editor:
Cheryl Ahrens Young

SEND TO

EDMONTON USERS GROUP
P.O. BOX 11983
EDMONTON, ALBERTA
CANADA T7E 7E1

MEMBERSHIP FEES:

Adult Individual: \$15/year
Junior Individual: \$10/year
Family: \$20/year
Sponsor: \$50/year
Non-participating: \$ 5/year

A junior is anyone under the age of 18; a non-participating member is someone out of the area who would like to receive the PRINTOUT. For renewing members, look for the asterisk by your name on the mailing label. Send your checks or money orders to:

SGV User's Group
1008 Dore Street
West Covina, California 91792

MEETING PLACES:

First wednesday of the month for the general meeting :: West Covina Library at 1601 W. Covina Parkway, off the 10 (San Bernardino Freeway) at 6:30 pm.

Second Thursday of the month for classes & library business: Pomona First Federal Savings & Loan at 18220 E. Colima Road; Rowland Heights (Take the Fullerton exit of the 60 freeway) 7:00 pm.

