

SUPER 99 MONTHLY

This month we have an education issue for you. Now that students have settled back into school and vacations are remembrances, it's time to put the computer back to use for learning.

Of course, there are a number of topics not devoted to education and some programming examples that can be used for both education and other areas of interest.

This month we've decided to make a few small but useful changes in the STANDARD LINE as follows:

ABBREVIATION	ITEM OPTIONAL	BRAND NOT OPTIONAL
o	X	
no		X
ono	X	X

This change was necessary to identify articles written specifically for the CorComp 9900 Disk Controller and may be of benefit in other instances.

Fun With Spatial Relationships

STANDARD: 1A 9A

Spatial Relationships Fun is an educational game for preschoolers to explore their ability to distinguish shapes using four shapes - triangle, square, heart, and diamond. A shape is shown at the top of the screen. The child is to select the shape at

the bottom of the screen that matches the shape at the top. Instructions are included in the game.

Keep in mind that this is only a game. To have your child tested in this important facet of development, consult a kindergarten, health center, or doctor.

As for the programming, several of the tips we introduced last month are used, including DISPLAY AT and instant screen formatting. Study the program for other techniques you may not be familiar with. Here is the listing:

```
>1 REM SUPER 99 SPATIAL
    RELATIONSHIPS FUN
>2 REM COPYRIGHT 1984
    SUPER 99 MONTHLY
>100 DIM SHAPE$(16)
>110 FOR I=1 TO 16
>120 READ SHAPE$(I)
>130 CALL CHAR(131+I,SHAPE$(I
))
>140 NEXT I
>150 CALL COLOR(16,5,1)
>160 FOR I=152 TO 155
>170 CALL CHAR(I,"0")
>180 NEXT I
>190 GOSUB 1000
>200 ON CHECK+1 GOTO 190,210,
    210,220,220
>210 GOSUB 2000
>220 FOR B=1 TO 3
>230 FOR A=1 TO 4
>240 SELECT=1
>250 GOSUB 4000
>260 GOSUB 5000
>270 GOSUB 6000
>280 GOSUB 7000
```

```

>290 CALL CHAR(152,"FFFFFFFF
FFFFFF")
>300 CALL KEY(5,K,8)
>310 IF S<1 THEN 300
>320 IF K<>32 THEN 380
>330 SELECT=SELECT+1
>340 IF SELECT<=4 THEN 360
>350 SELECT=1
>360 GOSUB 8000
>370 CALL CHAR(SELECT+151,"FF
FFFFFFFFFFFFFF")
>380 IF K<>13 THEN 300
>390 GOSUB 9000
>400 GOSUB 8000
>410 NEXT A
>420 NEXT B
>430 GOSUB 12000
>999 END
>1000 REM * INSTRUCTIONS ? *

>1010 CALL CLEAR
>1020 PRINT "DO YOU WANT INST
RUCTIONS?"
>1030 PRINT "PRESS Y OR N"
>1040 CALL KEY(5,K,S)
>1050 IF S<1 THEN 1040
>1060 A$="YyNn"
>1070 CHECK=POS(A$,CHR$(K),1)
>1080 RETURN
>2000 REM * INSTRUCTIONS *

>2010 CALL CLEAR
>2020 FOR A=1 TO 11
>2030 READ WORD$,X,Y
>2040 GOSUB 3000
>2050 NEXT A
>2060 CALL KEY(5,K,S)
>2070 IF S<1 THEN 2060
>2080 RETURN
>3000 REM * DISPLAY AT *

>3010 FOR I=1 TO LEN(WORD$)
>3020 CALL VCHAR(X,Y+1+I,ASC(
SEG$(WORD$,I,1)))
>3030 NEXT I
>3040 RETURN
>4000 REM * SET-UP BORDERS *

>4010 CALL CLEAR
>4020 FOR I=1 TO 4
>4030 CALL HCHAR(16,I*6-1,I+1
51,6)
>4040 CALL VCHAR(16,I*6-1,I+1
51,6)
>4050 CALL HCHAR(21,I*6-1,I+1
51,6)
>4060 CALL VCHAR(16,I*6+4,I+1
51,6)

```

```

>4070 NEXT I
>4080 RETURN
>5000 REM * SOLUTION SHAPE *

>5010 CALL HCHAR(3,16,128+4*A
)
>5020 CALL HCHAR(3,17,129+4*A
)
>5030 CALL HCHAR(4,16,130+4*A
)
>5040 CALL HCHAR(4,17,131+4*A
)
>5050 RETURN
>6000 REM * RANDOM & UNIQUE*

>6010 RANDOMIZE
>6020 FOR I=1 TO 4
>6030 C(I)=INT(4*RND)+1
>6040 IF I=1 THEN 6090
>6050 J=1
>6060 IF C(I)=C(J) THEN 6030
>6070 J=J+1
>6080 IF J<I THEN 6060
>6090 NEXT I
>6100 FOR I=1 TO 4
>6110 C(I)=C(I)*6+1
>6120 NEXT I
>6130 RETURN
>7000 REM * DISPLAY SHAPES *

>7010 N=132
>7020 FOR I=1 TO 4
>7030 CALL HCHAR(18,C(I),N)
>7040 CALL HCHAR(18,C(I)+1,N+
1)
>7050 CALL HCHAR(19,C(I),N+2)
>7060 CALL HCHAR(19,C(I)+1,N+
3)
>7070 N=N+4
>7080 NEXT I
>7090 RETURN
>8000 REM * ERASE BORDERS *

>8010 FOR I=152 TO 155
>8020 CALL CHAR(I,"0")
>8030 NEXT I
>8040 RETURN
>9000 REM * ANSWER RIGHT? *

>9010 CALL GCHAR(3,17,CORRECT
)
>9020 CALL GCHAR(18,SELECT*6+
2,ANSWER)
>9030 IF CORRECT-ANSWER=0 THE
N 9050
>9040 CORRECT=ANSWER+1
>9050 ON CORRECT-ANSWER+1 GOS
UB 10000,11000

```

```

>9060 RETURN
>10000 REM * RIGHT ANSWER *

>10010 RIGHT=RIGHT+1
>10020 FOR I=170 TO 1700 STEP
100
>10030 CALL SOUND(-1,I,0)
>10040 NEXT I
>10050 RETURN
>11000 REM * WRONG ANSWER *

>11010 FOR I=1700 TO 170 STEP
-100
>11020 CALL SOUND(-1,I,0)
>11030 NEXT I
>11040 RETURN
>12000 REM * PRINT SCORE *

>12010 CALL CLEAR
>12020 PRINT "SCORE = ";RIGHT
>12030 RETURN
>15000 REM * SHAPE PATTERNS*

>15010 DATA FFFFFFFFFFFFFFFFFF
>15020 DATA FFFFFFFFFFFFFFFFFF
>15030 DATA FFFFFFFFFFFFFFFFFF
>15040 DATA FFFFFFFFFFFFFFFFFF
>15050 DATA 80C0E0F0F8FCFEFF
>15060 DATA 0
>15070 DATA FFFFFFFFFFFFFFFFFF
>15080 DATA 80C0E0F0F8FCFEFF
>15090 DATA 00001C3E7F7F7F7F
>15100 DATA 0000387CFEFEFEFE
>15110 DATA 3F1F0F070301
>15120 DATA FCF8F0E0C0B
>15130 DATA 000103070F1F3F7F
>15140 DATA 0080C0E0F0F8FCFE
>15150 DATA 7F3F1F0F070301
>15160 DATA FEFCF8F0E0C0B
>16000 REM * INSTRUCTIONS *

>16010 DATA SPATIAL RELATIONS
HIPS FUN,3,1
>16020 DATA INSTRUCTIONS:,5,1
>16030 DATA USE THE SPACE BAR
TO MOVE,7,1
>16040 DATA FROM ONE SELECTIO
N TO,9,1
>16050 DATA ANOTHER. USE THE
ENTER KEY,11,1
>16060 DATA TO MAKE A SELECTI
ON.,13,1
>16070 DATA MATCH THE SHAPE A
T THE TOP,15,1
>16080 DATA OF THE SCREEN TO
ONE OF THE,17,1
>16090 DATA SHAPES AT THE BOT
TOM OF THE,19,1

```

```

>16100 DATA SCREEN.,21,1
>16110 DATA PRESS ANY KEY TO
CONTINUE,24,2

```

The game can be made more challenging by changing the shapes. Try forming the four shapes by rotating a triangle.

Program Development: The Power
of MERGE

STANDARD: 1A 2A 4A 5A 6A 7A 9A

Last month we outlined the proper structure of an Extended BASIC program, emphasizing the importance of the use of subroutines. When using Extended BASIC and disk drive(s), program development can be greatly simplified using MERGE.

Have you ever keyed in many program lines, only to determine later that they must be removed, line by line? It is a very slow process!

Try keying in your program as follows:

```

>100 GOSUB 1000
>110 GOSUB 2000
>120 END
>SAVE "DSK1.1-999",MERGE
>NEW
>1000 ! SET UP SCREEN
>1010 CALL CLEAR
>1020 CALL HCHAR(1,3,42,28)
>1030 CALL HCHAR(24,3,42,28)
>1040 CALL VCHAR(1,3,42,24)
>1050 CALL VCHAR(1,30,42,24)
>1060 RETURN
>SAVE "DSK1.1000-1999",MERGE
>NEW
>2000 ! DISPLAY MESSAGE
>2010 DISPLAY AT(12,5)SIE(19
);"MERGE DEMONSTRATION"
>2020 RETURN
>SAVE "DSK1.2000-2999",MERGE
>NEW

```

Now you can MERGE selected subroutines. For example, if you choose to omit the subroutine at line 1000, proceed as shown on the next page:

```
>MERGE "DSK1.1-999"  
>MERGE "DSK1.2000-2999"  
>100
```

Or, you can completely change a subroutine by writing over the program on disk, as follows:

```
>1000 REM SET UP SCREEN  
>1010 CALL CLEAR  
>1020 RETURN  
>SAVE "DSK1.1000-1999",MERGE  
>NEW
```

If you use a subroutine in many of your programs, SAVE the subroutine using MERGE and avoid otherwise using the program lines at which that subroutine will reside.

If a subroutine does not contain references to lines outside the subroutine, MERGE can be used to RESEQUENCE a portion of a program:

```
>MERGE "DSK1.1000-1999"  
>RES 3000,10  
>SAVE "DSK1.3000-3999",MERGE  
>DELETE "DSK1.1000-1999"  
>MERGE "DSK1.1-999"  
>100 GOSUB 3000  
>SAVE "DSK1.1-999"
```

After completing your program, do a normal SAVE, as follows:

```
>SAVE "DSK1.MERGE_DEMO"
```

Use either DELETE or the Disk Manager to eliminate subroutines that are no longer needed on your disk.

Parallel Default for
TI-Writer Fix 1

```
STANDARD: 1A 2CE 3B 4B 5A 6B 7A 9A  
10A
```

Last month we reported that the printer default provided by the fix to TI-Writer (recently released by TI to users groups) was an advantage only for serial printer users. So, we set about to find a way for parallel users to share the convenience of having a printer default. FORTH is required to

install our revision. We used an unaltered version of TI FORTH. Here are the steps to be followed:

1. Initialize a SS/SD diskette. Copy FORM1 from TI-Writer Fix 1 onto the diskette.

2. Insert the Editor/Assembler cartridge and FORTH System Disk. Boot FORTH (Load and Run "DSK1.FORTH"). Insert the FORM1 diskette set up in step 1 above. Key:

```
-EDITOR  
10 EDIT
```

Move the cursor to Row 12 Column 29. Change "RS232.BA=1200.LF" to match your printer requirements. For example, parallel users would change to "PIO.LF" or "PIO.CR". Press BACK (<FCTN> <9>). Key:

```
FLUSH
```

3. Copy the new version of file FORM1, along with the other fix files, to your TI-Writer diskette.

Now, when you use TI-Writer, your word processing will go faster since there will be a default for your own printer!

Multiple Choice Test Generator

```
STANDARD: 1A 2A 3B 4B 5A 6B 7A 9A  
10A
```

The program which is listed below will allow you to create multiple choice tests, take the test on the screen, and/or print the test on your printer. Questions and answers are linked and scrambled randomly for each test. Printouts include 2 test sheets and 1 answer sheet. Each printout has a control number so that study groups and/or instructors can match answer sheets to the corresponding test sheets. No questions or answers will be duplicated. The <ALPHA LOCK> key must be depressed except when creating questions and answers.

->

The program will prompt you to enter 20 questions and 80 answers to create a test. Each question can be up to 4 lines long, 28 characters per line. You must key <ENTER> after each line. One 24 character line is allowed for the correct answer. Three 24 character lines are provided for wrong answers.

It is advisable to catalog your disk to have a list of available tests as the program will "crash" if the name of the test you select is not the name of a file on the diskette.

If you are not interested in the educational possibilities of the program, it can be used to ask trivial questions, etc.!

Listing (TEST PRINTER COMMANDS!):

```
>1 MULTIPLE CHOICE TEST
  GENERATOR
>2 COPYRIGHT 1984
  SUPER 99 MONTHLY
>100 OPTION BASE 1
>110 DIM QUESTION$(20),ANSWER
$(20),CHOICE$(4),QSCRAMBLE(2
0),SCRAMBLE(4),LINE$(16)
>120 GOSUB 1000 ! MENU
>130 CALL KEY(5,K,S):: IF S<1
OR K<49 OR K>52 THEN 130 EL
SE IF K=52 THEN 999
>140 ON K-48 GOSUB 2000,3000,
4000
>150 RUN 100
>999 END
>1000 ! MENU
>1010 DISPLAY AT(1,3)ERASE AL
L:"MULTIPLE CHOICE TESTS"
>1020 DISPLAY AT(4,1):"1. CR
EATE TEST"
>1030 DISPLAY AT(6,1):"2. TE
ST"
>1040 DISPLAY AT(8,1):"3. PR
INT TEST AND ANSWERS"
>1050 DISPLAY AT(10,1):"4. E
ND"
>1060 DISPLAY AT(20,8)BEEP:"S
ELECT NOW."
>1070 RETURN
>2000 ! CREATE TEST
>2010 DISPLAY AT(1,8)ERASE AL
L:"CREATE TEST"
>2020 DISPLAY AT(3,1):"NAME O
F TEST:" :: ACCEPT AT(4,1)BE
```

```
EP SIZE(10):TESTNAME$
>2030 FOR A=1 TO 20
>2040 CALL HCHAR(9,1,32,480):
: DISPLAY AT(6,1):"QUESTION"
:A:"(FOUR LINES OF INPUT ALL
OWEDFOR QUESTIONS)" :: LETTE
RS=28 :: QUESTION$(A)=" "
>2050 FOR B=1 TO 4 :: ACCEPT
AT(B+9,1)BEEP:TEMP$ :: GOSUB
13000 :: NEXT B
>2060 DISPLAY AT(15,1):"CORRE
CT ANSWER:" :: LETTERS=24 ::
ACCEPT AT(16,1)BEEP SIZE(24
):TEMP$ :: GOSUB 13000 !
CONCATENATE STRING
>2070 DISPLAY AT(17,1):"WRONG
ANSWERS (3):" :: FOR B=1 TO
3 :: ACCEPT AT(B+17,1)BEEP
SIZE(24):TEMP$ :: GOSUB 1300
0 :: NEXT B
>2080 DISPLAY AT(22,1)BEEP:"D
ID YOU ENTER THE QUESTION P
ROPERLY (Y OR N)?"
>2090 CALL KEY(5,K,S):: IF S<
1 THEN 2090
>2100 Z$="YyNn" :: ON POS(Z$,
CHR$(K),1)+1 GOTO 2080,2110,
2110,2040,2040
>2110 NEXT A
>2120 OPEN #1:"DSK1."&TESTNAM
E$,RELATIVE,DISPLAY ,UPDATE,
FIXED 208
>2130 FOR A=1 TO 20
>2140 PRINT #1,REC A:QUESTION
$(A)
>2150 NEXT A
>2160 CLOSE #1
>2170 RETURN
>3000 ! TEST
>3010 DISPLAY AT(1,12)ERASE A
LL:"TEST"
>3020 DISPLAY AT(3,1):"SELECT
TEST:" :: ACCEPT AT(4,1)BEE
P SIZE(10):TESTNAME$
>3030 OPEN #1:"DSK1."&TESTNAM
E$,RELATIVE,DISPLAY ,UPDATE,
FIXED 208
>3040 GOSUB 11000 :: GOSUB 14
000 :: GOSUB 7000
>3050 FOR A=1 TO 20
>3060 CALL HCHAR(6,1,32,608):
: DISPLAY AT(6,1):"QUESTION"
:A
>3070 FOR B=1 TO 4 :: DISPLAY
AT(B+7,1):SEG$(QUESTION$(A)
,B*28-27,28):: NEXT B
>3080 Q=A :: GOSUB 8000
>3090 FOR B=1 TO 4 :: DISPLAY
```

```

AT(B+12,1):CHR$(64+B)&". "&
CHOICE$(SCRAMBLE(B)):: NEXT
B
>3100 DISPLAY AT(18,1):"ANSWE
R:" :: ACCEPT AT(18,10)BEEP
VALIDATE("ABCD")SIZE(1):RESP
ONSE$
>3110 IF CHOICE$(SCRAMBLE(ASC
(RESPONSE$)-64))=SEG$(QUESTI
ON$(A),113,24)THEN GOSUB 500
0 ELSE GOSUB 6000
>3120 GOSUB 10000 ! ANY KEY
>3130 NEXT A
>3140 DISPLAY AT(20,1)ERASE A
LL:"YOUR SCORE IS";RIGHT#5
>3150 GOSUB 10000 :: GOSUB 11
000
>3160 RETURN
>4000 ! PRINTOUT: 80 COLUMN
PRINTER REQUIRED
>4010 DISPLAY AT(1,3)ERASE AL
L:"PRINT TEST AND ANSWERS"
>4020 DISPLAY AT(3,1):"SELECT
TEST:" :: ACCEPT AT(4,1)BEE
P SIZE(10):TESTNAME$
>4030 DISPLAY AT(6,1):"PRINTE
R:" :: ACCEPT AT(7,1)BEEP:PN
AME$
>4040 DISPLAY AT(9,1):"NUMBER
OF COPIES TO PRINT" :: ACCE
PT AT(10,1)BEEP VALIDATE(DIG
IT):COPIES
>4050 OPEN #2:PNAME$ :: OPEN
#1:"DSK1."&TESTNAME$,RELATIV
E,DISPLAY,UPDATE,FIXED 208
:: GOSUB 11000
>4060 PRINT #2:CHR$(27);CHR$(
108);CHR$(3);CHR$(27);CHR$(8
1);CHR$(77);! SET MARGINS:
LEFT 8 RIGHT 77
>4070 PRINT #2:CHR$(27);CHR$(
68);CHR$(37);CHR$(0);! SET
TAB TO CENTER FOR RIGHT
PORTION OF PAGE - CHR$(9) IS
TAB
>4080 FOR D=1 TO COPIES
>4090 GOSUB 9000 :: GOSUB 140
00 :: GOSUB 7000
>4100 FOR E=1 TO 10
>4110 FOR F=1 TO 2
>4120 Q=QSCRAMBLE(E#2+F-2)::
GOSUB 8000
>4130 FOR G=1 TO 4
>4140 LINE$(G+8*(F-1))=SEG$(Q
UESTION$(Q),26#G-27,28)
>4150 LINE$((G+4)+8*(F-1))=CH
OICE$(SCRAMBLE(G))
>4160 IF SEG$(QUESTION$(Q),11

```

```

3,24)=CHOICE$(SCRAMBLE(G))TH
EN ANSWER$(E#2+F-2)=CHR$(G+6
4)
>4170 NEXT G
>4180 NEXT F
>4190 FOR F=1 TO 8
>4200 IF E<6 THEN G=E ELSE G=
E+5
>4210 IF F=1 THEN PRINT #2:ST
R$(G)&". ";LINE$(F);CHR$(9)
;STR$(G+5)&". ";LINE$(F+8)
>4220 IF F>1 AND F<5 THEN PRI
NT #2:RPT$(" ",3+LEN(STR$(G)
));LINE$(F);CHR$(9);RPT$(" "
,3+LEN(STR$(G+5)));LINE$(F+8
)
>4230 IF F>4 THEN PRINT #2:"
";CHR$(F+60)&". ";LINE$(F)
;CHR$(9);" ";CHR$(F+60)&".
";LINE$(F+8)
>4240 NEXT F
>4250 PRINT #2 :: PRINT #2:"A
NSWER ___";CHR$(9);"ANSWER
___" :: PRINT #2
>4260 IF E=5 THEN PRINT #2:CH
R$(12):: GOSUB 9000 ! GOTO
NEXT PAGE AND PRINT HEADING
>4270 NEXT E
>4280 PRINT #2:CHR$(12)
>4290 GOSUB 9000
>4300 FOR F=1 TO 10 :: IF F<6
THEN G=F ELSE G=F+5
>4310 PRINT #2:RPT$(" ",2-LEN
(STR$(G)));STR$(G)&". ";ANS
WER$(F#2-1);
>4320 PRINT #2:CHR$(9);RPT$("
",2-LEN(STR$(G+5)));STR$(G+
5)&". ";ANSWER$(F#2):: PRIN
T #2 :: PRINT #2 :: NEXT F
>4330 PRINT #2:CHR$(12)
>4340 NEXT D
>4350 CLOSE #1 :: CLOSE #2
>4360 RETURN
>5000 ! CORRECT RESPONSE
>5010 RIGHT=RIGHT+1
>5020 FOR B=170 TO 1700 STEP
100 :: CALL SOUND(-1,B,0)::
NEXT B
>5030 DISPLAY AT(20,8):"VERY
GOOD!!!"
>5040 RETURN
>6000 ! INCORRECT RESPONSE
>6010 FOR B=1700 TO 170 STEP
-100 :: CALL SOUND(-1,B,0)::
NEXT B
>6020 DISPLAY AT(20,1):"THE C
ORRECT ANSWER IS":SEG$(QUEST
ION$(A),113,24)

```

```

>6030 RETURN
>7000 ! INPUT FROM DISK
>7010 FOR B=1 TO 20 :: LINPUT
#1,REC QSCRAMBLE(B):QUESTIO
N$(B):: NEXT B
>7020 RETURN
>8000 ! SCRAMBLE ANSWERS
>8010 FOR B=1 TO 4 :: CHOICE#
(B)=SEG$(QUESTION$(Q),113+(B
-1)*24,24) :: NEXT B ! GET
ANSWERS FROM STRING
>8020 RANDOMIZE
>8030 FOR B=1 TO 4
>8040 C=1 :: SCRAMBLE(B)=INT(
4*RND)+1
>8050 IF B=1 THEN 8080
>8060 IF SCRAMBLE(B)=SCRAMBLE
(C)THEN 8040
>8070 C=C+1 :: IF C<B THEN 80
60
>8080 NEXT B
>8090 RETURN
>9000 ! PAGE HEADER
>9010 FOR B=1 TO 4 :: PRINT #
2 :: NEXT B
>9020 PRINT #2:RPT$(" ",INT((
74-LEN(TESTNAME$))/2));TESTN
AME$
>9030 PRINT #2:RPT$(" ",INT((
74-LEN("CONTROL NO. "&STR$(D
)))/2));"CONTROL NO. "&STR$(
D)
>9040 PRINT #2
>9050 RETURN
>10000 ! ANY KEY
>10010 DISPLAY AT(24,2)BEEP:"
PRESS ANY KEY TO CONTINUE"
>10020 CALL KEY(5,K,S):: IF S
<1 THEN 10020
>10030 RETURN
>11000 ! USER MESSAGE
>11010 DISPLAY AT(24,1):"WORK
ING...ONE MOMENT PLEASE!"
>11020 RETURN
>13000 ! BUILD RECORD
>13010 QUESTION$(A)=QUESTION#
(A)&TEMP#&RPT$(" ",LETTERS-L
EN(TEMP#))
>13020 RETURN
>14000 ! SCRAMBLE QUESTIONS
>14010 RANDOMIZE
>14020 FOR A=1 TO 20
>14030 B=1 :: QSCRAMBLE(A)=IN
T(20*RND)+1
>14040 IF A=1 THEN 14070
>14050 IF QSCRAMBLE(A)=QSCRAM
BLE(B) THEN 14030
>14060 B=B+1 :: IF B<A THEN 1

```

```

4050
>14070 NEXT A
>14080 RETURN

```

DISPLAY AT USING in Console BASIC

STANDARD: 1A 9A

This article shows how to convert numbers for use by the DISPLAY AT routine we covered last month. Plus, the example program simulates the Extended BASIC statement DISPLAY AT USING. That is to say, numbers are formatted. The example program rounds to 2 decimal places, adds a leading dollar sign if desired, and, if required, adds a comma for numbers up to 999,999.99 and trailing decimal point and Zero(s). After the formatting, the number is ready for display using the DISPLAY AT routine, with the decimal point being the point of reference. In other words, the column selected is the column in which the decimal point will be placed - not the column at which the "\$" will appear. Five numbers will be displayed. After the fifth number appears, press any key to end the program. DISPLAY AT USING is useful in many situations, including for education, business, and recreation. Here is the example program listing:

```

>100 CALL CLEAR
>110 REM READ TEXT TO BE
DISPLAYED AND ROW
AND COLUMN
>120 FOR A=1 TO 6
>130 READ WORD$,X,Y
>140 GOSUB 2000
>150 NEXT A
>160 REM READ NUMBERS TO BE
DISPLAYED, ROW,
COLUMN, AND $ (?)
>170 FOR A=1 TO 5
>180 READ B,X,Y,@$
>190 GOSUB 3000
>200 GOSUB 2000
>210 NEXT A
>220 CALL KEY(5,K,S)
>230 IF S<1 THEN 220
>999 END
>2000 REM DISPLAY AT
>2010 FOR D=1 TO LEN(WORD$)

```

->

```

>2020 CALL VCHAR(X,Y+1+D,ASC(
  SEG$(WORD$,D,1)))
>2030 NEXT D
>2040 RETURN
>3000 REM USING

>3010 REM ROUND TO NEAREST
  PENNY
>3020 B=INT(100*B+.5)/100
>3030 REM TRUNCATE NUMBER
  TO 2 DECIMAL
  PLACES, ADD $ (?)
>3040 WORD$=@$&SEG$(STR$(B)&"
  .",1,POS(STR$(B)&".".",1))
  &SEG$(STR$(B)&"000",POS(STR$(
  B)&".".",1)+1,2)
>3050 REM INSERT A COMMA FOR
  #'S TO 999,999.99
>3060 IF LEN(WORD$)<=7 THEN 3
  080
>3070 WORD$=SEG$(WORD$,1,LEN(
  WORD$)-6)&"."&SEG$(WORD$,LEN
  (WORD$)-5,7)
>3080 REM SET TAB FOR
  DECIMAL POINT
>3090 Y=Y-(LEN(WORD$)-3)
>3100 RETURN
>5000 REM PHRASES TO DISPLAY
>5010 DATA "TEST DISPLAY AT U
  SING:",1,1
>5020 DATA "ROUND TO 2 DECIMA
  L PLACES",3,1
>5030 DATA "$ OPTION, ADD COM
  MA'S",5,1
>5040 DATA "SELECT ROW AND CO
  LUMN.",7,1
>5050 DATA "THIS PROGRAM WORK
  S WITH ANY",20,1
>5060 DATA "FORM OF TI BASIC!
  !!",22,1
>6000 REM NUMBER, ROW,
  COLUMN OF DECIMAL,
  AND $ (?)
>6010 DATA 25,9,9,"$"
>6020 DATA 321.7802,10,9," "
>6030 DATA 872449.272,11,9,"
  "
>6040 DATA 1410.135,12,9," "
>6050 DATA 1,13,9," "

```

Beginning BASIC: FOR-NEXT Loops

STANDARD: 1A 9A

Loops are used to condense repetitive steps into a few program

lines. A FOR-NEXT loop can use almost any value to begin, end and increment. For instance, if you want to number the odd rows up to row 9 on the screen by placing a number in the first text column, here is one way to do so:

```

>100 FOR A=49 TO 57 STEP 2
>110 CALL VCHAR(A-48,3,A)
>110 NEXT A

```

Recognizing when a loop can be used is in large part a matter of experience and dedication. Believe us, the patterns are not always easy to recognize! Try anyway!

If someone has convinced you that your TI-99/4A is no better than other computers (or, perish the thought, not as good), try running this program on your TI-99/4A and then another brand:

```

>100 FOR A=1 TO 100 STEP .1
>110 PRINT A
>120 NEXT A

```

Most other computers, even the huge business systems, will make an error before 40 - some before 4! This can be "fixed", but who wants to bother with seemingly illogical extra steps? Documentation on the unique TI number handling system can be found on page III-13 of the BASIC Reference Manual.

USING "Double FORTH"

STANDARD: 1A 2C 4B 5A 6B 7A 9A

TI FORTH is a very flexible language. One common alteration to standard TI FORTH is to change it to recognize screens on disk side 2 and/or disk 2. If you have already done this, we have some tips for you. If you haven't, we'll try to get to how to double FORTH next month.

The editors from -EDITOR and -64SUPPORT cannot be loaded at the same time. If you'd like to be able to switch from one editor to the other quickly, we have a solution. Be sure you are using disks that don't contain

programs on screen 7 or any screen above 90 and with the only changes to screen 3 being for DISK-HI, DISK-LO, and DISK-SIZE.

What we are going to do is set up your own personal rapid load, a binary image load created by BSAVE and loaded with BLOAD, in a manner that will allow you to quickly change editors.

Key -EDITOR and 7 EDIT. On screen 7 list the menu options that you often use. Try to arrange these from the most used to the least used so that you may be able to FORGET the ones at the bottom of the list later. The last two options should be -BSAVE and -EDITOR. Here is one possibility:

-GRAPH -DUMP -VDPMODES -COPY
-PRINT -BSAVE -EDITOR

Do a <FCTN> <9> to return to immediate mode and key the following:

FLUSH
COLD
7 LOAD
' TASK 91 BSAVE

Find the first blank screen above 91. Our example would place the blank screen at 101. Repeat the above steps using -64SUPPORT in place of -EDITOR and 101 or your first blank screen in place of 91. Key 3 EDIT. Toward the bottom of screen 3, you should key 91 BLOAD (101 BLOAD), with the one in parentheses being your second BLOAD screen number.

To change editors, just edit screen 3 and reverse the parentheses to the other BLOAD and do a COLD. Your original disk is left essentially unchanged and you can easily check or update your BLOAD's later by looking at screen 7!

Beginning FORTH: XBASIC To FORTH

STANDARD: 1A 2C 4B 5A 6B 7A 9A

Putting definitions onto disk screens is really not much different

from using immediate mode. We have received a number of requests for information on converting XBASIC to FORTH and simulating XBASIC's numeric ACCEPT. Key -SYNONYMS -FLOAT -EDITOR. Insert your FORTH program diskette and find a blank screen (if you haven't done anything with the disk since last month, all screens except 4 and 5 should be available). We'll use screen 10 as an example. Key 10 CLEAR and 10 EDIT. FORTH is not very picky about your format, but you'll find it best to lay out a screen neatly. Here is the FORTH (XBASIC?) program:

```
0 BASE->R DECIMAL
1 0 VARIABLE AMOUNT ( LET AMOUNT=0 )
2 : CALL_CLEAR CLS ;
3 : ACCEPT_SIZE4_AMOUNT
4           PAD 1+ 4 EXPECT VAL
5           FAC->S AMOUNT ! ;
6 : AT      GOTOXY ;
7 : DISPLAYAMOUNT AMOUNT ? ;
8 : RUN     CALL_CLEAR 0 0 AT
9           ACCEPT_SIZE4_AMOUNT
10          0 5 AT
11          DISPLAYAMOUNT ;
12 R->BASE
```

Key <FCTN> <9>, FLUSH, 10 LOAD, RUN.

Hardware Troubleshooting: The Keyboard

STANDARD: 1A 9A

Keys not working is a fairly common problem. Foreign particles get into the keyboard either through carelessness or even no apparent fault of the user. It is easy to recognize a key that does not respond. Keys that won't quit responding is a more difficult to recognize problem. One episode we were contacted about went like this - turn on the computer and touch no keys, jump through the title screen, fill 4 lines with comma's, give a constant beep, and lock up - the comma key was stuck on! Contact your dealer! To try to fix it, rock the key carefully until it comes off, apply alcohol lightly and replace the key. This often fixes the key.

Using the CorComp 9900
Disk Controller

STANDARD: 1A 2A (o) 2B (o) 4B 5A
6B (no) 7A 9A

Before getting into actual programs using the CorComp Controller, let's get into some of the concepts involved, beginning with the impressive screen movement statements.

It is important to locate an area of memory into which you can store your screen displays. The lower section of Low Memory should be reserved for assembly and related functions. In console BASIC, the solution is simple - High Memory, which ranges from address 40960 to 65535, provides enough space for 32 full screens, which should be plenty for most purposes. Using Mini Memory, the 4K in Mini Memory and the 24K in High Memory are usually the best choices, with a particular selection depending on the requirements of the program. In XBASIC, the solution may not be so easy. The power of XBASIC is tied to XBASIC's use of memory all over the place! Study the memory maps in the Controller Manual. Also, note that the First Free Address in Low Memory is pointed to by address 8194, the Last Free Address in Low Memory is pointed to by 8196, and the Highest Free Address in Memory Expansion is pointed to by -31866. If you'd prefer to load only a few screens and guess where they might fit, addresses from 10000 to 16000 are usually fairly safe bets to be free, but test carefully! Detailed memory maps are beyond the

scope of this article (many pages would be required to cover all of the XBASIC possibilities).

Now let's look at what to store. An excellent idea would be to store an XBASIC input screen that may need different responses later. Here is what we mean - set up your DISPLAY AT's through a subroutine. Then set up corresponding ACCEPT AT's using a negative SIZE (for defaults) in a subroutine. After the last ACCEPT, store the screen. To update the info, skip the DISPLAY subroutine and just bring in the screen you stored. Your default values will already be there! Remember, almost anything can be moved. Try copying your Pattern Descriptor Table (in BASIC, 30 to 159 only, in XBASIC, 30 to 143 only), redefine the characters, copy the new characters, copy the first set back into VDP. WOW! Instant character sets! Try this with the Color Table. Be very careful in trying to manipulate the Sprite Pattern Table, Sprite Motion Table and Sprite Attribute Table as you could throw all sorts of pointers off. Try seeing how quickly you can draw a graph or chart. Such drawings might be good for BASIC or Mini Memory as the extra characters might come in handy. We are also working on our Disk Loader program from last month to speed up the screen displays. And, of course, the game possibilities are endless. Just imagine combining sprites and really fast moving characters! We're working on the above ideas and more. Next month we'll start getting into the actual how's!

We hereby acknowledge any registrations, copyrights, or other legal rights held in association with the company names and/or products listed below:

Texas Instruments, Inc.: TI, TI-99/4A, TI-Writer, TI Extended BASIC, TI BASIC,
TI Disk Manager II, Mini Memory
Star Micronics, Inc.: Gemini 15-X PC
CorComp, Inc.: 9900 Disk Controller Card
TEAC: TEAC 55B Disk Drive

CORRECTIONS TO SEPTEMBER:

PAGE 6: There should be a space after the ":" in the FORTH definition of "SUPER".

PAGE 8: In the DISPLAY AT example, if you change to the text columns, using "Y+1" in line 2020, the string in line 5020 will be too long. Change to "THIS SHOULD BE ROW 5".

PAGE 9: We omitted the fact that the margs are generated from left to right, from top to bottom.

Last month we detailed several steps to take in testing the CorComp 9900 Disk Controller. Most of the tests would be valid for any change in disk configuration. Our CorComp card was not functioning properly - more than likely we had a bad resistor, though we never verified that assumption (our dealer swapped it out for a new one). Inconsistent disk accesses have been reported to occur when adding a second drive to a TI Controller if TI's resistor pack is not installed. At least one user we know of has received the pack at no charge by calling TI. So, check those new disk configurations carefully!

Many readers have written asking for sources of products and info.

If you are interested in quality programs at a reasonable price (\$3 each), send \$1 (refundable with order) for a list of over 130 programs to Tigercub Software, 156 Collinwood Ave., Columbus, OH 43213. Mr. Jim Peterson writes the programs. We call Jim "King of the one-liners", both for his wit and his amazing ability to accomplish tasks in a single line of BASIC. Jim also sends tips to users groups (he requires a small free plug). If you are in a group and your newsletter editor is not printing the tips, fuss at your editor!

If you are interested in TI and HP calculators, TI CC-40 and computer books (including info about a new book on FORTH by Leo Brodie), send \$1 for a catalog to EduCALC Mail Store, 27953 Cabot Road, Laguna Niguel, CA 92677. We know nothing about the firm, but have seen their catalog.

We are having no problems in locating sources of most products. Extended BASIC is a bit difficult to find at a discounted price, but is available from TI's marketing agent, Triton. If you have never received a Triton catalog, their address is P.O. Box 8123, San Francisco, CA 94128.

COMING IN NOVEMBER: Business and Home Finance issue! Plenty of tips and programs! (And the articles we didn't have space for this month - we haven't forgotten!)

AT&T will be testing 56000 baud telecommunications through ordinary phone lines in 12 major cities this year! Such lines presently seldom support over 1200 baud, with the rate for more expensive dedicated lines (for businesses) usually standing at 19200 and below.

When sending in questions, please be as specific as possible, even if your letter runs long. We will make a sincere effort to publish a routine or program that matches your request if we know what language, application, and equipment you have in mind!

TI has now made the assembly source code for FORTH available to users groups. The programs are remarked for easy understanding by experienced assembly programmers.

->

Super 99 Monthly is published monthly by Bytemaster Computer Services, 171 Mustang Street, Sulphur, LA 70663. Subscription rate in U.S. and possessions is \$12.00 per year; all other countries \$16.00 U.S. funds for surface mail. All correspondence received will be considered unconditionally assigned for publication and copyright and subject to editing and comments by the editors of Super 99 Monthly. Each contribution to this issue and the issue as a whole Copyright 1984 by Bytemaster Computer Services. All rights reserved. Copying done for other than personal archival or internal reference use without the permission of Bytemaster Computer Services is prohibited. Bytemaster Computer Services assumes no liability for errors in articles.

STANDARD KEY

1	Computer	A	TI-99/4A
2	Cartridge	A	Extended BASIC
		B	Mini Memory
		C	Editor/Assembler
		E	TI-Writer
3	RS-232	B	TI
4	Disk Drive	A	TI
		B	TEAC 55B
5	Expansion Box	A	TI
6	Disk Controller	A	TI
		B	CorComp
7	32K Card	A	TI
9	Monitor or TV	A	Tv & RF Modulator
10	Printer	A	Gemini 15-X PC

Note: This list will be adjusted monthly to relate to current articles. The reference symbols will always correspond to the same products.

Bytemaster Computer Services
171 Mustang Street
Sulphur, LA 70663

08/85

