VOLUME 1 ISSUE 12 $1.50 AUGUST 1985

SUPER 99 MONTHLY

THE LATEST NEWS FROM MYARC IS ON PAGE 11! WE BRING IT TO YOU FIRST!

This issue represents the conclusion of our first year. Thanks to strong support from our readers, we anticipate being around for quite some time to come. There's no time to reflect on the past year, other than to say it has been a real pleasure, so here's wishing you happy computing in our second year! We hope you are looking forward to it as much as we are!

BASIC

TWO EDUCATIONAL GAMES

STANDARD: 1A 9A

Paul Yorke has granted us permission to print two more of his programs. These were written by Paul and Dave Juckett. The values in a 3 by 3 matrix are unknown, while the totals of the rows, columns and diagonals are known. The user deduces the unknown values. For the second program, change only a few lines and you switch from addition to multiplication! These are great for children of all ages!

```
100 REM ADDITION MAGICIAN
110 REM LOGIC/ADDITION GAME
120 REM FOR ALL AGES
130 CALL CLEAR
140 REM DESIGNED AND WRITTEN
150 REM BY DAVE JUCKETT AND
160 REM PAUL YORKE
170 REM 1200 STARFISH LANE
180 REM STUART FLA. 33494 US
A
190 PRINT "      ADDITION MA
GICAN"
200 PRINT
210 PRINT "    BY DAVE JUCKE
TT AND"
220 PRINT "          PAUL  YO
RKE"
230 FOR I=1 TO 8
240 PRINT
250 NEXT I
260 INPUT "INSTRUCTIONS?(Y/N
)";A$
270 IF A$="N" THEN 290
280 GOSUB 2330
290 CALL CLEAR
300 GUESS=0
310 RANDOMIZE
320 FOR X=1 TO 9
330 C(X)=INT(RND*9)+1
340 FOR Y=0 TO X-1
350 IF C(X)=C(Y)THEN 330
360 NEXT Y
370 NEXT X
380 CALL COLOR(14,9,9)
390 CALL CHAR(128,"181818181
8181818")
400 CALL CHAR(129,"000000FFF
F000000")
410 CALL CHAR(130,"181818FFF
F181818")
420 CALL CHAR(136,"FFFFFFFFF
FFFFFFF")
430 FOR X=9 TO 21 STEP 4
```

-->

```
440 CALL VCHAR(5,X,128,13)
450 NEXT X
460 FOR X=5 TO 17 STEP 4
470 CALL HCHAR(X,9,129,13)
480 NEXT X
490 CHAR=65
500 FOR ROW=6 TO 14 STEP 4
510 FOR COLUMN=10 TO 18 STEP
 4
520 CALL HCHAR(ROW,COLUMN,CH
AR)
530 CHAR=CHAR+1
540 NEXT COLUMN
550 NEXT ROW
560 A(2)=C(1)+C(2)+C(3)
570 A(3)=C(4)+C(5)+C(6)
580 A(4)=C(7)+C(8)+C(9)
590 A(1)=C(3)+C(5)+C(7)
600 A(5)=C(1)+C(5)+C(9)
610 A(6)=C(1)+C(4)+C(7)
620 A(7)=C(2)+C(5)+C(8)
630 A(8)=C(3)+C(6)+C(9)
640 FOR X=1 TO 5
650 COLUMN=24
660 M$=STR$(A(X))
670 ROW=(4*X)-1
680 GOSUB 1930
690 NEXT X
700 FOR X=6 TO 8
710 ROW=19
720 COLUMN=(4*X)-14
730 M$=STR$(A(X))
740 GOSUB 1930
750 NEXT X
760 ROW=3
770 COLUMN=5
780 M$="# OF GUESSES "
790 GOSUB 1930
800 ROW=21
810 COLUMN=7
820 M$="SELECT A LETTER"
830 GOSUB 1930
840 ROW=23
850 COLUMN=7
860 M$="CHOOSE A NUMBER"
870 GOSUB 1930
880 FOR R=7 TO 15 STEP 4
890 FOR COLUMN=11 TO 19 STEP
 4
900 CALL GCHAR(R,COLUMN,XX)
910 YY=YY+XX
920 NEXT COLUMN
930 NEXT R
940 IF YY=477 THEN 2200
950 YY=0
960 CALL KEY(0,BOX,STATUS)
970 IF BOX>73 THEN 960
980 CALL HCHAR(21,24,136)
990 FOR DELAY=1 TO 20
1000 NEXT DELAY
1010 CALL HCHAR(21,24,32)
1020 IF STATUS=0 THEN 960
1030 IF BOX<65 THEN 960
1040 M$=CHR$(BOX)
1050 ROW=21
1060 COLUMN=23
1070 GOSUB 1930
1080 CALL KEY(0,N,STATUS)
1090 IF N>57 THEN 1080
1100 CALL HCHAR(23,24,136)
1110 FOR DELAY=1 TO 20
1120 NEXT DELAY
1130 CALL HCHAR(23,24,32)
1140 IF STATUS=0 THEN 1080
1150 IF N<49 THEN 1080
1160 M$=CHR$(N)
1170 ROW=23
1180 COLUMN=23
1190 GOSUB 1930
1200 N=N-48
1210 X=BOX-64
1220 GUESS=GUESS+1
1230 ROW=3
1240 COLUMN=18
1250 M$=STR$(GUESS)
1260 GOSUB 1930
1270 ON X GOTO 1280,1350,142
0,1490,1560,1630,1710,1780,1
850
1280 IF N<>C(1)THEN 2000
1290 GOSUB 2100
1300 ROW=7
1310 COLUMN=10
1320 M$=STR$(C(1))
1330 GOSUB 1930
1340 GOTO 880
1350 IF N<>C(2)THEN 2000
1360 GOSUB 2100
1370 ROW=7
1380 COLUMN=14
1390 M$=STR$(C(2))
1400 GOSUB 1930
1410 GOTO 880
1420 IF N<>C(3)THEN 2000
1430 GOSUB 2100
1440 ROW=7
1450 COLUMN=18
1460 M$=STR$(C(3))
1470 GOSUB 1930
1480 GOTO 880
1490 IF N<>C(4)THEN 2000
1500 GOSUB 2100
1510 ROW=11
1520 COLUMN=10
1530 M$=STR$(C(4))
1540 GOSUB 1930
```

-->

```
1550 GOTO 880
1560 IF N<>C(5)THEN 2000
1570 GOSUB 2100
1580 ROW=11
1590 COLUMN=14
1600 M$=STR$(C(5))
1610 GOSUB 1930
1620 GOTO 880
1630 IF N<>C(6)THEN 2000
1640 GOSUB 2100
1650 ROW=11
1660 COLUMN=18
1670 M$=STR$(C(6))
1680 GOSUB 1930
1690 GOTO 880
1700 REM
1710 IF N<>C(7)THEN 2000
1720 GOSUB 2100
1730 ROW=15
1740 COLUMN=10
1750 M$=STR$(C(7))
1760 GOSUB 1930
1770 GOTO 880
1780 IF N<>C(8)THEN 2000
1790 GOSUB 2100
1800 ROW=15
1810 COLUMN=14
1820 M$=STR$(C(8))
1830 GOSUB 1930
1840 GOTO 880
1850 IF N<>C(9)THEN 2000
1860 GOSUB 2100
1870 ROW=15
1880 COLUMN=18
1890 M$=STR$(C(9))
1900 GOSUB 1930
1910 GOTO 880
1920 GOTO 1920
1930 REM GOSUB
1940 LN=LEN(M$)
1950 FOR I=1 TO LN
1960 CALL HCHAR(ROW,COLUMN+1
,ASC(SEG$(M$,I,1)))
1970 COLUMN=COLUMN+1
1980 NEXT I
1990 RETURN
2000 ROW=1
2010 COLUMN=10
2020 M$="TRY AGAIN"
2030 GOSUB 1930
2040 FOR DELAY=1 TO 400
2050 NEXT DELAY
2060 COLUMN=10
2070 M$="
2080 GOSUB 1930
2090 GOTO 960
2100 ROW=1
2110 COLUMN=10
2120 M$="GOOD GUESS!"
2130 GOSUB 1930
2140 FOR DELAY=1 TO 400
2150 NEXT DELAY
2160 COLUMN=10
2170 M$="
2180 GOSUB 1930
2190 RETURN
2200 ROW=1
2210 COLUMN=8
2220 M$="NICE GOING!!!"
2230 GOSUB 1930
2240 FOR DELAY=1 TO 1000
2250 NEXT DELAY
2260 COLUMN=6
2270 M$="PLAY AGAIN?? Y/N"
2280 GOSUB 1930
2290 CALL KEY(0,KEY,STATUS)
2300 IF STATUS=0 THEN 2290
2310 IF KEY=89 THEN 290
2320 END
2330 CALL CLEAR
2340 PRINT "INSTRUCTIONS:"
2350 PRINT "A 3 BY 3 GRID WI
LL APPEAR ONTHE SCREEN.EACH
SQUARE WILL HAVE A LETTER TO
 IDENTIFY IT"
2360 PRINT
2370 PRINT "TO THE RIGHT OF
EACH ROW A  NUMBER TELLS YOU
 THE SUM OF THE NUMBERS(0-9)
IN THAT ROW"
2380 PRINT "NO DIGITS ARE US
ED MORE THANONCE"
2390 PRINT
2400 PRINT "A NUMBER BELOW E
ACH COLUMN  AND AT THE END O
F THE       DIAGONALS GIVES
THE SAME    INFORMATION."
2410 PRINT
2420 PRINT "USING THIS INFO
ENTER THE   MISSING NUMBERS,
NO DIGITS   REPEAT."
2430 PRINT
2440 INPUT "PRESS ENTER TO P
LAY GAME":B$
2450 RETURN
```

Here's the changes from the first program that are necessary for the multiplication program:

```
100 REM  MULTIPLICATION MADN
ESS
110 REM  LOGIC/MULTIPLICATIO
N GAME
190 PRINT "   MULTIPLICATION
 MADNESS"
```

```
560 A(2)=C(1)*C(2)*C(3)
570 A(3)=C(4)*C(5)*C(6)
580 A(4)=C(7)*C(8)*C(9)
590 A(1)=C(3)*C(5)*C(7)
600 A(5)=C(1)*C(5)*C(9)
610 A(6)=C(1)*C(4)*C(7)
620 A(7)=C(2)*C(5)*C(8)
630 A(8)=C(3)*C(6)*C(9)
2370 PRINT "TO THE RIGHT OF
EACH ROW A  NUMBER TELLS YOU
 THE PRODUCTOF THE NUMBERS(1
-9)  IN THAT ROW"
2380 PRINT "ALL DIGITS ARE U
SED,ALL ONLYONCE."
```

EXTENDED BASIC

Enlarged Random Numbers

STANDARD: 1A 2XB 9A

Many organizations have use for random numbers for door prizes and such. The program listed below will print enlarged numbers in a selectable range between 1 and 999. While we realize many of you have likely seen such a program before, have you seen it in only 10 program lines?

```
100 ON BREAK NEXT :: DISPLAY
 AT(9,1)ERASE ALL:"SUPER 99
MONTHLY":"ENLARGED RANDOM NU
MBERS":"(1-999)":;:"LOW:":"H
IGH:"
110 ACCEPT AT(13,7)VALIDATE(
DIGIT)BEEP SIZE(3):L :: ACCE
PT AT(14,7)VALIDATE(DIGIT)BE
EP SIZE(3):H :: IF H<L THEN
100
120 DISPLAY AT(23,1):"ONE MO
MENT...LOADING DATA" :: CALL
 PATTERNS(B$(,))
130 DISPLAY AT(23,1)ERASE AL
L:"GETTING NUMBER" :: RANDOM
IZE :: V=INT(RND*(H-L+1))+L
:: V$=STR$(V)
140 FOR I=1 TO 8 :: FOR J=1
TO LEN(V$):: C$=C$&B$(VAL(SE
G$(V$,J,1)),I):: NEXT J :: F
$=F$&C$&RPT$(" ",28-LEN(C$))
:: C$="" :: NEXT I
150 DISPLAY AT(9,1):F$ :: F$
="" :: DISPLAY AT(22,1):"PRE
SS ANY KEY FOR ANOTHER   RAN
DOM NUMBER":L;"TO":H:"PRESS
ENTER TO END"
160 CALL KEY(5,K,S):: IF S<1
 THEN 160 ELSE IF K<>13 THEN
 130 :: CALL CLEAR :: END
20000 SUB PATTERNS(B$(,)):
CALL CHAR(62,RPT$("F",16)):
 H$="0123456789ABCDEF"
20010 C$="        >  >   >> >
   > > >>  >>>>   >  >> > >
>>>>  >> >>>> >>>>" :: FOR I
=48 TO 57 :: CALL CHARPAT(I,
P$):: FOR J=1 TO 8
20020 FOR K=1 TO 2 :: B$(I-4
8,J)=B$(I-48,J)&SEG$(C$,(POS
(H$,SEG$(P$,J*2+K-2,1),1)-1)
*4+1,4):: NEXT K :: NEXT J :
: NEXT I :: SUBEND
```

MULTIPLAN™

SYLK BUILDER - VERSION 1.1

STANDARD: 1A 2XB MP 3B 4B 5A 6B 7A 9A

In May, we began discussing SYLK files and offered a program to convert DIS/FIX 80 files to SYLK format. The major problem with the method was that it involved a number of cumbersome steps. The problem is solved! The modifications listed below will enable doing the sector editing in the same step as writing the SYLK file!

The sector editing is accomplished via Barry Traver's R_A_W program, which many of you may already be familiar with. R_A_W is an Assemly Language program, but before those of you who dread Assembly frown, we have converted it to Extended BASIC by using CALL LOAD's to place the assembled code into memory. We highly recommend that you SAVE the subprogram from lines 21000 to 21480 to disk in MERGE format, for possible use in other endeavors (name it RAW, which, by the way, stands for Read and Write).

And, another surprise! SYLK files can contain control codes, such as printer commands! This will allow you to set up for fancy printing, such as double-strike or condensed print.

-->

- AUGUST 4 -
SUPER 99 MONTHLY

The characters for the commands can even be set up in TI-Writer, using the Special Character Mode, which we covered in March.

Listed below are the changes to the May SYLK program. Lines 170 and 190 are changes and the rest of the lines are additions. Lines 170 and 190 were added as an error trap to make certain that your source file does not exceed the number of rows available on the spreadsheet. Be sure to test your program thoroughly (on a "work disk"), as the RAW subprogram has a lot of numbers that could easily be keyed incorrectly!

WARNING: SYLK Builder 1.1 does not do disk read and write error checking! Use a "work disk", a disk you would not cry over if the data on it were lost or spoiled, when creating your SYLK files. While we have built a number of SYLK files with this program without problems, we do not want to give the impression that it is infallible. We also recommend use of a disk that has fewer than 20 files on it for quick processing.

```
170 IF C>255 THEN DISPLAY AT
(15,1):"SOURCE FILE TOO LONG
,":"TRUNCATED TO 255 LINES"
ELSE IF EOF(1)=0 THEN 150
190 C=MIN(C,255):: DISPLAY A
T(6,1):"ENTER DESIRED SYLK F
ILE NAME":"DSK ."
380 CLOSE #1 :: CLOSE #2 ::
CALL RAW
390 S=2
400 D=VAL(SEG$(F2$,1,1))
410 DISPLAY AT(24,1):"READIN
G SECTOR ";S :: CALL LINK("R
EAD",D,S,A$,B$)
420 IF (SEG$(A$,1,MAX(0,POS(
A$," ",1)-1))<>SEG$(F2$,3,10
))*(SEG$(A$,1,MAX(0,POS(A$,C
HR$(0),1)-1))<>SEG$(F2$,3,10
))THEN S=S+1 :: GOTO 410
430 A$=SEG$(A$,1,12)&CHR$(2)
&SEG$(A$,14,115)
440 CALL LINK("WRITE",D,S,A$
,B$)
21000 SUB RAW
21010 CALL INIT
21020 CALL LOAD(16360,87,82,
73,84,69,32,39,138)
21030 CALL LOAD(16368,82,69,
65,68,32,32,38,192)
21040 CALL LOAD(16376,68,83,
82,76,78,75,37,50)
21050 CALL LOAD(8194,40,108,
63,232)
21060 CALL LOAD(9460,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0)
21070 CALL LOAD(9482,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0)
21080 CALL LOAD(9504,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,100,32,0,
46,170,37,12,37,54)
21090 CALL LOAD(9526,193,126
,83,224,37,46,192,32,131,86,
194,64,2,41,255,248,4,32,32,
40,208,193)
21100 CALL LOAD(9548,9,131,7
,4,2,2,37,2,5,128,5,132,128,
196,19,6,4,32,32,40,220,129)
21110 CALL LOAD(9570,152,1,3
7,48,22,246,193,4,19,82,2,13
2,0,7,21,79,4,224,131,208,20
0,4)
21120 CALL LOAD(9592,131,84,
200,4,36,252,5,132,168,4,131
,86,200,32,131,86,36,254,2,2
24,131,224)
21130 CALL LOAD(9614,4,193,2
,12,15,0,195,12,19,1,30,0,2,
44,1,0,4,224,131,208,2,140)
21140 CALL LOAD(9636,32,0,19
,50,200,12,131,208,29,0,2,2,
64,0,152,18,37,49,22,238,160
,160)
21150 CALL LOAD(9658,37,22,1
6,3,192,160,131,210,29,0,192
,146,19,230,200,2,131,210,5,
194,194,114)
21160 CALL LOAD(9680,209,96,
131,85,19,9,156,133,22,242,9
,133,2,6,37,2,156,182,22,237
,6,5)
21170 CALL LOAD(9702,22,252,
5,129,200,1,37,0,200,9,36,25
0,200,12,36,248,6,153,16,226
,30,0)
21180 CALL LOAD(9724,2,224,3
7,12,192,9,4,32,32,40,9,209,
22,4,3,128,2,224,37,12,4,193
)
21190 CALL LOAD(9746,6,193,2
15,65,243,224,37,46,3,128,1,
16,175,192,192,0,192,192,68,
0,0,0)
21200 CALL LOAD(9768,0,0,0,0
```

-->

```
,88,32,35,100,131,124,2,1,15
8,126,200,1,131,114,4,32,33,
36)
21210 CALL LOAD(9790,208,96,
131,124,32,96,35,100,19,20,1
93,32,32,42,192,68,2,2,38,38
,2,3)
21220 CALL LOAD(9812,0,6,156
,177,22,6,6,3,22,252,193,81,
6,149,16,0,16,5,2,36,0,8)
21230 CALL LOAD(9834,2,132,6
4,0,26,238,4,224,131,124,2,2
24,131,224,3,0,0,2,4,32,0,0)
21240 CALL LOAD(9856,255,255
,255,255,255,255,255,255,255
,255,255,255,255,255,255,255
,255,255,255,255,255,255,255)
21250 CALL LOAD(9878,255,255
,255,255,255,255,255,255,255
,255,255,255,255,255,255,255
,255,255,255,255,255,255,255)
21260 CALL LOAD(9900,255,255
,255,255,255,255,255,255,255
,255,255,255,255,255,255,255
,255,255,255,255,2,224)
21270 CALL LOAD(9922,38,160,
2,0,128,0,216,0,38,30,4,192,
2,1,0,2,4,32,32,12,4,32)
21280 CALL LOAD(9944,32,24,1
8,184,194,32,131,74,2,136,0,
0,17,78,2,136,5,159,21,75,2,
0)
21290 CALL LOAD(9966,3,192,2
00,0,131,86,2,1,38,28,2,2,0,
2,4,32,32,36,4,192,2,1)
21300 CALL LOAD(9988,0,1,4,3
2,32,12,4,32,32,24,18,184,19
2,96,131,74,6,193,2,33,0,1)
21310 CALL LOAD(10010,2,2,60
,239,4,224,131,74,200,8,131,
80,200,1,131,76,200,2,131,78
,4,32)
21320 CALL LOAD(10032,37,50,
0,10,2,0,60,239,2,1,38,31,2,
2,0,128,4,32,32,44,2,32)
21330 CALL LOAD(10054,0,128,
194,0,4,192,2,1,0,3,2,2,38,3
0,4,32,32,16,192,8,2,1)
21340 CALL LOAD(10076,38,31,
2,2,0,128,4,32,32,44,4,192,2
,1,0,4,2,2,38,30,4,32)
21350 CALL LOAD(10098,32,16,
120,32,131,124,131,124,2,224
,131,224,4,96,0,112,2,0,30,0
,4,32)
21360 CALL LOAD(10120,32,52,
2,224,38,160,2,0,128,0,216,0
,38,30,4,192,2,1,0,2,4,32)
21370 CALL LOAD(10142,32,12,
4,32,32,24,18,184,194,32,131
,74,2,136,0,0,17,233,2,136,5
,159)
21380 CALL LOAD(10164,21,230
,4,192,2,1,0,3,2,2,38,30,4,3
2,32,20,4,195,208,224,38,30)
21390 CALL LOAD(10186,6,195,
2,131,0,128,22,73,2,0,60,239
,2,1,38,31,2,2,0,128,4,32)
21400 CALL LOAD(10208,32,36,
2,32,0,128,193,192,4,192,2,1
,0,4,2,2,38,30,4,32,32,20)
21410 CALL LOAD(10230,4,195,
208,224,38,30,6,195,2,131,0,
128,22,48,192,7,2,1,38,31,2,
2)
21420 CALL LOAD(10252,0,128,
4,32,32,36,2,0,3,192,200,0,1
31,86,2,1,38,28,2,2,0,2)
21430 CALL LOAD(10274,4,32,3
2,36,4,192,2,1,0,1,4,32,32,1
2,4,32,32,24,18,184,192,96)
21440 CALL LOAD(10296,131,74
,6,193,2,2,60,239,4,224,131,
74,200,8,131,80,200,1,131,76
,200,2)
21450 CALL LOAD(10318,131,78
,4,32,37,50,0,10,120,32,131,
124,131,124,2,224,131,224,4,
96,0,112)
21460 CALL LOAD(10340,2,0,28
,0,4,32,32,52,0,0)
21470 ! CALL LINK("WRITE",D,
S,A$,B$)
21480 ! CALL LINK("READ",D,S
,A$,B$)
21490 SUBEND
```

Two Spreadsheets for Statistics

by Charles M. Robertson

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

When a sculpter looks at a piece of wood or rock, he sees a thing of beauty trapped within and sets out to free it. I try to see in this way from time to time and it is usually worth the effort.

One of the least used software packages I own, Multiplan™, also happens to be one of the most powerful. My problem has been finding

-->

applications for it once the checkbook has been balanced and the bills are paid. I wish I had a string of restaurants or a couple of multinational corporations or a nice, fat stock portfolio to keep track of, but -- the injustices of life -- I don't. What to do with this very powerful raw material? After wiping the dust from the manual, I began to look through the eyes of creativity for an application that would unlock the power of this program and delegate another of life's tediums to my solid-state slave. I saw two things clearly -- Multiplan (tm) is a very powerful and versatile program; I hate statistics.

These observations may seem a little disjointed, but creativity is like that at times. Besides, I was desperate. I had some data to analyze.

Statistics, usually referred to as "sadistics" in the classes I took, is not a laugh-a-minute pursuit for most. However, few decisions of consequence are reliably made these days without data analysis. Since a spreadsheet is a decision tool, I decided to tool it into a tedium cruncher for the data analysis I had to perform. This article introduces the first two entries of a collection of statistics templates for Multiplan (tm).

The first template calculates Pearson's r, the product-moment correlation coefficient. This statistical procedure is one of the most widely used indicators of degree-of-relationship in social science. When two numerical measures can be obtained for each subject with many subjects, or for each tria with many trials of measurement, the degree of relationship and the direction of relationship between the factors being measured can be developed with this statistical procedure.

I will stop here with the "statistics language" with one caution. I am not a statistician, so beware. Further, if you are not a statistician, you should beware. Red squirrel populations have been found to correlate nicely with rates of airline disasters. Be careful of the conclusions you may be tempted to draw from your data. Correlation indicates relationship, not causality. I am trusting that those of you who will use these templates are familiar with the measures they calculate and the limitations and pitfalls associated with their use and misapplication. Woe unto the rest. Study of the proper application of the procedures is worthwhile, however, since the practical uses of correlational analysis are legion.

In the templates below, where a name occupies a cell, that cell should be "named" accordingly (yes, you can make up your own names if you are consistent in the formulas); where Multiplan (tm) nomenclature is used, you should enter these as the formulas. The templates are designed for expansion to the limits of your data or Multiplan (tm) with the use of Insert Row and Copy Down commands. Once the templates are expanded to fit your data you can Copy From named columns of data on another sheet or enter data directly to the template. I suggest you first save a blank template with a generic file name such as "Pearson" for future use. (ed.: Array notations in COL 1 and 2 represent paired data and should be blank, awaiting input, on your spreadsheet. RM)

	1	2	3	4	5
1	PEARSON'S r				
2					
3					
4	1ST VARIABLE	2ND VARIABLE	X SQUARED	Y SQUARED	X*Y
5	----------	----------	----------	----------	----------
6					
7	DATA (X)	DATA (Y)	X*X	Y*Y	X*Y
8	----------	----------	----	----	----------
9					
10	X(1)	Y(1)	RC[-2]*RC[-2]	RC[-2]*RC[-2]	RC[-4]*RC[-3]
11	X(2)	Y(2)	RC[-2]*RC[-2]	RC[-2]*RC[-2]	RC[-4]*RC[-3]
12	.	.	.	.	.
13	.	.	.	.	.
14	.	.	.	.	.
15	X(N)	Y(N)	RC[-2]*RC[-2]	RC[-2]*RC[-2]	RC[-4]*RC[-3]
16	--------	--------	--------	--------	--------
17	SX	SY	SXSQ	SYSQ	SXY
18					
19					
20	FOR N=	N			
21					
22	CORRELATION=	((N*SXY)-(SX*SY))/SQRT(((N*SXSQ)-(SX*SX))			*((N*SYSQ)-
23					(SY*SY)))
24					
25	----------	----------	----------	----------	----------

-->

```
  WHERE :      SX      IS   SUM(R10:15C1)
               SY      IS   SUM(R10:15C2)
               SXSQ    IS   SUM(R10:15C3)
               SYSQ    IS   SUM(R10:15C4)
               SXY     IS   SUM(R10:15C5)
               N       IS     COUNT(CT)
    AND        CT  NAMES       R10:15C1

CORRELATION= ((N*SXY)-(SX*SY))/SQRT(((N*SXSQ)-(SX*SX)) *((N*SYSQ)-
                                                          (SY*SY)))

 NOTE : USE INSERT ROWS BETWEEN 11 AND 14 TO ENSURE FORMULAS WHEN
        YOU COPY DOWN ON COLUMNS 3,4,AND 5.
```

The next template calculates the Point-Biserial correlation coefficient and is particularly useful when one of the variables under study is a dichotomous variable such as Male-Female, Purchased product-Didn't purchase product, or Voted yes-Voted no, etc.

The above cautions, re: knowing your limits with statistics, apply even more strongly with this proceedure. A test for the significance of the correlation coefficient obtained, particularly with a small sample, is essential to avoid misinterpreting your results. A template for this significance calculation may be forthcoming in this series if interest warrants.

```
        1              2              3              4
 1   POINT-BISERIAL CORRELATION
 2
 3
 4     NUM.VAR      Y SQUARED    DICHOT.VAR   Y/N CAPTURE
 5   ------------ ------------ ------------   -----------
 6
 7        Y           Y*Y       1=YES,0=NO       X*Y
 8     --------     --------     ---------      ------
 9
10       Y(1)     RC[-1]*RC[-1]    1 OR 0    RC[-1]*RC[-3]
11       Y(2)     RC[-1]*RC[-1]    1 OR 0    RC[-1]*RC[-3]
12       Y(3)     RC[-1]*RC[-1]    1 OR 0    RC[-1]*RC[-3]
13        .             .             .             .
14        .             .             .             .
15       Y(N)     RC[-1]*RC[-1]    1 OR 0    RC[-1]*RC[-3]
16     --------     --------     --------      --------
17        SY          SYSQ          YES           SYYES
18
19
20                                   NO            SYNO
21              N=       CT
22   BIAS S STDV=       BSSDY
23    MEAN Y YES=       MYYES
24     MEAN Y NO=       MYNO
25
26   POINT-BISERIAL CORRELATION (Rpb) IS :((MYYES-MYNO)*SQRT(YES*NO
27                                        ))/(BSSDY*(YES+NO))
28
29   ------------------------------------------------------------
```

```
     WHERE :          SY     IS   SUM(R10:15C1)
                      SYSQ   IS   SUM(R10:15C2)
                      YES    IS   SUM(R10:15C3)
                      SYES   IS   SUM(R10:15C4)
                      BSSDY  IS   SQRT((SYSQ-(SY*SY)/CT)/CT)
                      NO     IS   COUNT(N)-YES
                      SYNO   IS   SY-SYYES
                      CT     IS   COUNT(N)
                      MYYES  IS   SYYES/YES
                      MYNO   IS   SYNO/NO
     AND 'N' NAMES R10:15C1

     NOTE: ROWS SHOULD BE INSERTED BETWEEN 10 AND 15 TO  ALLOW THE
           FORMULAS AND NAMES TO REMAIN EFFECTIVE.
           USE 'COPY DOWN' ON COLUMNS 2 AND 4 AFTER ROWS ARE ADDED
```

-->

FORTH

Standards for the TI (or A Tower of Babel)

by Howie Rosenberg

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

When I received my first copy of FORTH DIMENSIONS, The journal of the FORTH Interest Group (FIG), I tried to enter several programs that caught my eye. To my dismay I couldn't get any of them to work. Each had several words which were undefined in any reference that I was aware of at the time. They certainly weren't FIG FORTH, nor were they FORTH 79. In point of fact they were the author's words, so deeply embedded into his mind that he had forgotten that they were not standard FORTH words. This has happened often since that time, but to add to the annoyance I am now in possession of FORTH code, written for the TI, which has the same characteristics. The very properties of FORTH which make it so powerful are the same factors which could cause total confusion among us. The language is extensible. The compiler can be modified so that the actual characteristics of what happens during compile and execute time can be varied. The characteristics of existing words can be easily changed by a knowledgable programmer. Even those with less expertise can easily change the screens on a FORTH disk (for better or worse).

The original TI public domain release had a bug on screen 72 which everybody should have corrected by now. The version sold by TEX-SOFT did not have that bug. On the surface the two versions were indistinguishable. In addition to those factors which could cause total confusion there are pitfalls we can fall into in writing FORTH code for distribution to others which serve to make life difficult for them. Both classes of problems are discussed below. Where there are simple solutions they are presented. The others are going to require a concerted effort by the TI FORTH community if we are to avoid a FORTH Tower of Babel.

BOOTING APPLICATIONS---There are basically two methods of booting a FORTH application. The code may be saved as an image using the BSAVE utility and loaded using the BLOAD word. The user need not then be concerned about words of which he is unfamiliar, nor need he do anything more than load and run the application. I said USER and that's the keyword. The situation is not unlike having assembly object code. Nobody I know enjoys modifying someone elses object code. Nor is it fun to study it for learning purposes. Commented source code is so much nicer. For an application which is not likely to require change or customizing and is not intended to be used for tutorial purposes, BSAVED code is fine. All other applications should be supplied on commented screens. FORTH applications disks can then be transported with the kernel, loader and 90 or 180 screens (single or double sided) or as disks containing just applications screens. Here's a potential problem area. During boot after the FORTH system is setup screen 3 is loaded. Suppose you start with a disk that was initialized double sided, and copy a FORTH disk. If you use a sector copier which copies sector 0 so that it is identical to the original disk the copy is usable. Otherwise the new disk won't boot because the FORTH system is looking for screen 3 in the wrong place. I've seen several of these disks floating around already.

LOADING OPTIONS---Most annoying during load is to get an error message such as HCHAR ? (meaning that the compiler did not recognize HCHAR), and discover that -GRAPH which compiles HCHAR was not loaded by the application screens. Doesn't everyone have the Graphics options loaded in image on their master disk? While a number of users do incorporate options which are often used on their masters, there is no universal combination which everybody uses. It turns out that all options are loaded by what is called a conditional load. If they are already loaded, they will not be loaded again, hence they can be called in an application screen and will be loaded only if not already in the dictionary. This still leaves a problem if one desires to make the application compatible with a single drive system. Suppose you are loading an application from a disk full of screens. With a

-->

one drive system you first boot FORTH and then switch disks and LOAD the application screens. Obviously, if any options are called out, an error occurs. The conditional load of options in the application is probably best taken care of by a remark so that users can load prior to loading the application. We must do what is required to make EVERY FORTH disk usable by anyone without regard for his configuration.

FOREIGN WORDS----I recently received a FORTH disk on which a number of applications used the STARTING FORTH words which appear in appendix C of the FORTH manual. While it was relatively easy to discover the cause of the boot problem, newcomers to the language could be somewhat perplexed by all those question marks during load. Any word that is used in an application, should be defined in a colon definition IN THAT APPLICATION. Particular care has to be taken to avoid the inclusion of your own words. It really is easy to forget that MYWORD is not part of TI FORTH.

REDEFINED DEFINITIONS----Words that appear in screen options are easily changeable. If any of us start assuming that FORTH disks that we acquire with intact option screens are correct versions we're asking for trouble. It's also relatively easy in FORTH to change definitions that appear in the FORTH kernel. It should be noted that so doing can play havoc with the language in that any word that uses the word that you alter will also be altered. I would suggest that any kernel that has been altered should have a note to that effect that clearly is displayed during operation. I really can't say that nobody should ever alter a kernel word because there are some bugs and minor inconveniences that crop up which if controlled properly we can (should) fix.

LOADING SCREENS----WHICH SCREENS ARE WE LOADING?----If a screen contains the words 70 LOAD 85 LOAD, placing the disk in drive 2 or for that matter placing the screen on a double sided version of FORTH would cause the wrong screen to load. As a result of the flexibility in disk formats that we enjoy it's better to stick with --> for a group of screens and if necessary even repeat code from elsewhere on a disk to avoid using the LOAD word.

WHAT DO I RUN?----Generally speaking, most FORTH applications after boot do nothing until a word is executed. Of course, it's possible to have the LOADed screens execute a word which places us in a loop which is a program like environment. I suspect that many applications will take on this format. Generally, however, one must execute a word to get anything to happen. The question usually is which words are executable, what do they do, and what do I have to put on the stack to make it all happen. Another field day for the ingenious user because many times little or no information is given. In addition to the normal FORTH standard of depicting the stack conditions (of PARAMOUNT importance for executable words), the programmer should include sufficient remarks to denote which words can be executed and what they do.

CONCLUSIONS----Unless we come up with a set of standards, and conventions we will indeed have a Tower of Babel. This is particularly true for kernel changes. I expect to be plugging away on my TI-99/4A for a long time, and as FORTH is my language of choice, I would like to see it preserved. Perhaps if those of similar interest would combine in committee or by periodic interchange not unlike our FIG brothers at a higher level, we can keep Babel from occurring. As for the minor annoyances, many of which are described above, I have spoken to several of our FORTH programmers and find similar concerns among others in our FORTH community. In particular, Warren Agee, who helped in formulating the ideas presented in this article, has a suggestion which, if we can implement, will alleviate many of the problems. Suppose all FORTH programmers used the first screen of an application as a document screen. All executable words would be listed as well as required stack inputs and options required. A small price to pay in disk space and effort to make it a bit easier on us all. So simple, we could carry it off!! Any feedback, comments or suggestions in this area would be appreciated. Send to this magazine or if you like you can contact me on CIS (74216,1640).

-->

99 POTPOURRI

News, Corrections, Updates, Editorials, Kudos, and Come-what-may

Look for a great new product to appear around mid-November. Craig Miller has developed "GRAM Cracker", a grom port device that will allow the loading and/or modification of modules from disk or cassette! GRAM Cracker uses GRAM and RAM to act as the GROM and ROM in the cartridges. The device will have 48K of memory, user expandable to 88K, allowing the loading of several modules at once, with the option of software or manual selections. Loads are almost instant, much like a RAMDISK. An independently powered unit, GRAM Cracker will have complete documentation. The device is not cheap to produce, but the price will be very reasonable for what you get (under $200 is what we hear). Utilities on disk will be included! We have talked to persons who have seen GRAM Cracker in action and they say it is fantastic!

Asgard Software has many fine products for the 99/4A. The Asgard Light Pen--> Manufactured by Bob Emmel of Pennsylvania, this high quality pen is an excellent photoreceptor. In other words, it will detect light on the screen and tell your program so. It will not detect certain colors, permitting the mixing of text within the program. It is useful for games, utilities and educational programs. Bundled with it is a 3 program disk, illustrating its uses in BASIC and Extended BASIC. Total price is $18.00. GRAPHX COMPANION-> This package supplements the high quality artwork included with the GRAPHX drawing system by GRAPHX of Australia. It includes 2 disks of new fonts, clipart, animation examples and pictures that are not only useful (and legally usable) in your own works of art, but also illustrate some drawing techniques exclusive to GRAPHX. This package retails for $7.00. For more information on Asgard products, write POB 10306, Rockville, MD 20850 or contact your local dealer (in the Philadelphia area, see Pilgrims' Pride in Hatboro).

Barry Traver is offering diskazine, a magazine on disk. The yet untitled project is available at an introductory price of $30 for a six-issue subscription. It is available in either DS/SD or SS/SD (commercial flippy) format. Several top programmers have contributed for the upcoming first issue. Barry plans to soon have several articles available for this publication and will be offering material in diskazine that will be unique to that publication. The diskazine address is Barry Traver, Editor, Genial Computerware, 835 Green Valley Drive, Philadelphia, PA 19128.

Our COnference on the TI FORUM on Compuserve™ now appears to be likely for early September. Feel free to drop in to discuss Super 99 Monthly, Multiplan™ or whatever you wish. We will stay in COnference until the chatter runs dry or CIS runs us out, whichever comes first. The likely date is either September 14 or 21.

Late news from Myarc (Lou Phillips) is that their 128K card will soon be updated to eliminate problems in handling null strings. Software houses are now adapting their programs to become Myarc compatible. Read news from Myarc first right here!

[]
[] Compuserve is a registered trademark of Compuserve Information Services. []
[] Multiplan is a registered trademark of Microsoft Corp. []
[] []
[]

-->

SUBSCRIPTIONS (PER YEAR):

U.S. AND POSSESSIONS	
FIRST CLASS	$16.00
THIRD CLASS	$12.00
OTHER COUNTRIES	
AIR MAIL	$26.50
SURFACE MAIL	$16.00
INDIVIDUAL COPIES:	
U.S. SUBSCRIBERS	
FIRST CLASS	$ 1.35
THIRD CLASS	$ 1.00
CANADA SUBSCRIBERS	$ 1.35
OTHER	$ 1.50

Check or Money Order in U.S. funds.
(all issues available at press time)

SUPER 99 MONTHLY ORDER FORM

NAME_____

ADDRESS_____

CITY_____STATE_____

ZIP_____COUNTRY_____

For back issues, specify which:

_____

READER FEEDBACK: (Attach comments)

Super 99 Monthly is published monthly by Bytemaster Computer Services, 171 Mustang Street, Sulphur, LA 70663. 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.

EDITOR
Richard M. Mitchell (CIS 70337,1011)

CORRESPONDING STAFF WRITERS

Barry A. Traver
Charles M. Robertson

STANDARD KEY

1	Computer	A	TI-99/4A
2	Cartridge	XB	Extended BASIC
		MP	Multiplan (tm)
		EA	Editor/Assembler
3	RS-232	B	TI
4	Disk Drive	B	TEAC 55B
5	Expansion Box	A	TI
6	Disk Controller	B	CorComp
7	32K Card	A	TI
9	Monitor or TV	A	TV & RF Modulator
10	Printer	A	Gemini 15-X PC

Bytemaster Computer Services
171 Mustang Street
Sulphur, LA 70663

Bulk Rate
U.S. Postage
PAID
Sulphur, LA 70663
Permit No. 141

POSTMASTER: ADDRESS CORRECTION REQUESTED.