TI PPC NOTES

NEWSLETTER OF THE TI PERSONAL PROGRAMMABLE CALCULATOR CLUB
P.O. Box 1421, Largo, FL 34649

Volume 12, Number 4 Fourth Quarter 1987/1988

The emphasis in this issue is on translating TI-59 programs for use with the TI-95. I admit to underestimating the difficulty of translation, particularly for individuals who were not familiar with the TI-59. And, I haven't even begun to address what a translator should do when confronted with HIR commands, fast mode, and the like.

This issue also illustrates another reality--the TI-59 participation is winding down and is being replaced by interest in the TI-74 and the TI-95. In the coming year we will continue to provide coverage for the older machines such as the TI-59, TI-66 and CC-40, but expect to emphasize the newer machines.

Another thing you will notice in this issue is the appearance of a new cadre of participants. Examples include Scott Garver, S. L. Lee and Don Laughery. Old-timers will remember that Don was very active in earlier years. Other new members who submitted material which arrived too late for this issue include Ralph Ennis who has submitted a massive 2600 step discriminant function program, Ross Garside who has submitted a data packing routine, and Steve Shergold who submitted a plotting routine for the TI-74/PC-324. Scott Garver has also submitted a translation of the backgammon program for the TI-95 and has volunteered to assist others in translation. Meanwhile, the "old hands" are staying busy. Hewlett has expanded his list of programs for the TI-95, and Robert Prins has promised a tutorial for the next issue.

With this issue we have completed the eighth "year" of our newsletter under the name TI PPC Notes. I put the word year in quotation marks as an admission that I have difficulty getting four issues out in a calendar year. That is unlikely to change so long as I have to earn a living. Some members have suggested that I should ask for renewals for a half-year so that we can get back on an approximate calendar basis. Unfortunately, the amount of bookkeeping is about the same whether there are two issues or four, and I have a real desire to minimize bookkeeping so that I can concentrate on programming. So, a subscription form for another "year" of four issues is attached. All I can promise is four issues at three to four month intervals with each issue comprised of the usual 24 to 28 pages of material. If you should decide not to continue your membership for the future I would appreciate a note to that effect.

The use of material in this newsletter is entirely at the user's risk. No responsibility as to the accuracy and the consequences due to the lack of it will be borne by either the club or the editor.

ERRATA

Scrambling in BASIC - V10N3P17. Maurice Swinnen writes: I keyed the program into my CC-40 and found that it simply hung up in a loop, line 85 back to 70, ad infinitum. The problem was line 20 which uses the expression Q = ATN(1.E+14) to calculate the value pi/2. That is acceptable if the machine is in radian mode, but if the machine is in degree mode Q = 90, or if the machine is in grad mode Q = 100. With either of those values for Q the value for S will become zero after a few cycles, and the endless loop will be initiated. The problem can be averted by inserting a RAD statement in the program at line 20 as illustrated in the program in the leftmost column below. A printout for a seed of 234 is illustrated in the second column below.

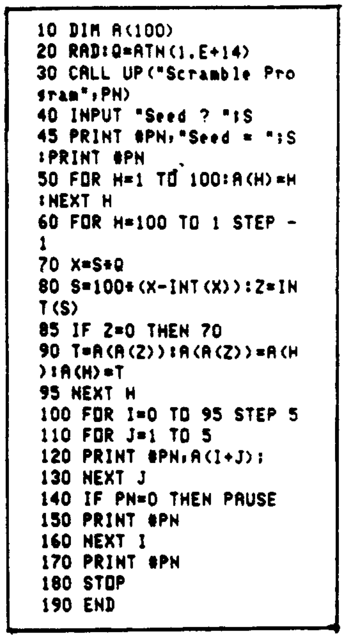
Editor's Note: The omission of control of the angle mode was my error. I simply converted Larry's Model 100 program for the CC-40 and failed to remember that the Model 100 only runs in the radian mode.

Maurice also suggested that we could replace Larry's random number generator in lines 70 through 95 with the RANDOMIZE and RND functions which are available in the CC-40 and TI-74. One such solution appears in the program in the third column below. Comments on the program follow.

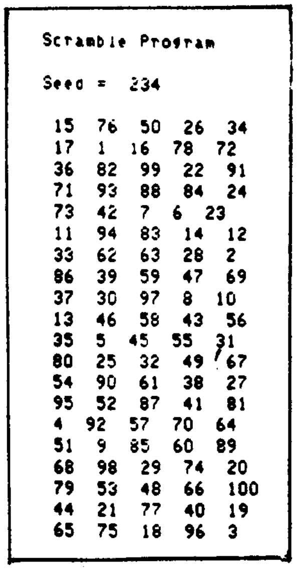
Line 20 - the RANDOMIZE statement without a following numeric expression sets the random number generator to an unpredictable sequence. As a result there is no need for an operator selected seed. The scrambler will yield a sequence which is not under user control, exactly what we really wanted.

Line 70 - the RND function calls a new random number between zero and one. Multiplying by 100 and applying the integer function then generates a series of two digit integers for use by the scramble routine. As noted in earlier issues it would not be necessary to use the integer function since non-integer values are acceptable as subscripts.

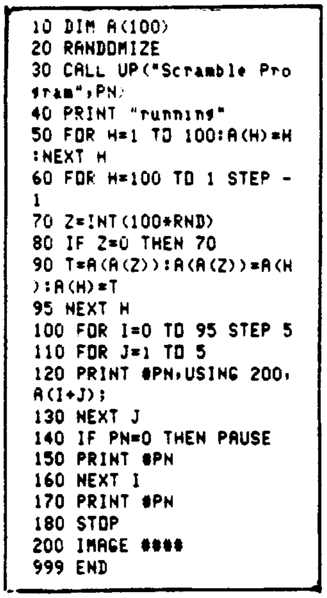
Line 120 - the Print Using function makes the columns of the output line up. See the printout in the righthand column below.



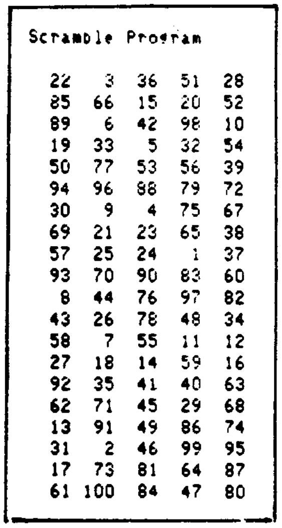
```
10 DIM A(100)
20 RAD:Q=ATN(1.E+14)
30 CALL UP("Scramble Pro
gram",PN)
40 INPUT "Seed ? ";S
45 PRINT #PN,"Seed = ";S
:PRINT #PN
50 FOR H=1 TO 100:A(H)=H
:NEXT H
60 FOR H=100 TO 1 STEP -
1
70 X=S*Q
80 S=100*(X-INT(X)):Z=IN
T(S)
85 IF Z=0 THEN 70
90 T=A(A(Z)):A(A(Z))=A(H
):A(H)=T
95 NEXT H
100 FOR I=0 TO 95 STEP 5
110 FOR J=1 TO 5
120 PRINT #PN,A(I+J);
130 NEXT J
140 IF PN=0 THEN PAUSE
150 PRINT #PN
160 NEXT I
170 PRINT #PN
180 STOP
190 END
```



```
Scramble Program

Seed =  234

 15  76  50  26  34
 17  1  16  78  72
 36  82  99  22  91
 71  93  88  84  24
 73  42  7  6  23
 11  94  83  14  12
 33  62  63  28  2
 86  39  59  47  69
 37  30  97  8  10
 13  46  58  43  56
 35  5  45  55  31
 80  25  32  49  67
 54  90  61  38  27
 95  52  87  41  81
 4  92  57  70  64
 51  9  85  60  89
 68  98  29  74  20
 79  53  48  66  100
 44  21  77  40  19
 65  75  18  96  3
```



```
10 DIM A(100)
20 RANDOMIZE
30 CALL UP("Scramble Pro
gram",PN)
40 PRINT "running"
50 FOR H=1 TO 100:A(H)=H
:NEXT H
60 FOR H=100 TO 1 STEP -
1
70 Z=INT(100*RND)
80 IF Z=0 THEN 70
90 T=A(A(Z)):A(A(Z))=A(H
):A(H)=T
95 NEXT H
100 FOR I=0 TO 95 STEP 5
110 FOR J=1 TO 5
120 PRINT #PN,USING 200,
A(I+J);
130 NEXT J
140 IF PN=0 THEN PAUSE
150 PRINT #PN
160 NEXT I
170 PRINT #PN
180 STOP
200 IMAGE ####
999 END
```



```
Scramble Program

  22   3  36  51  28
  85  66  15  20  52
  89   6  42  98  10
  19  33   5  32  54
  50  77  53  56  39
  94  96  88  79  72
  30   9   4  75  67
  69  21  23  65  38
  57  25  24   1  37
  93  70  90  83  60
   8  44  76  97  82
  43  26  78  48  34
  58   7  55  11  12
  27  18  14  59  16
  92  35  41  40  63
  62  71  45  29  68
  13  91  49  86  74
  31   2  46  99  95
  17  73  81  64  87
  61 100  84  47  80
```

A NEW CLUB FOR THE TI-74 AND TI-95 - Thomas Coppens, who previously was the editor of the TISOFT newsletter for users of the TI-59 and TI-99/4, has announced the beginning of a newsletter and software exchange club for the TI-74 and TI-95. The organization is called SeTIc, which stands for Software exchange for Texas Instruments calculators. The newsletter is available in either French or Dutch. A one year subscription is fifteen dollars ($15.00).

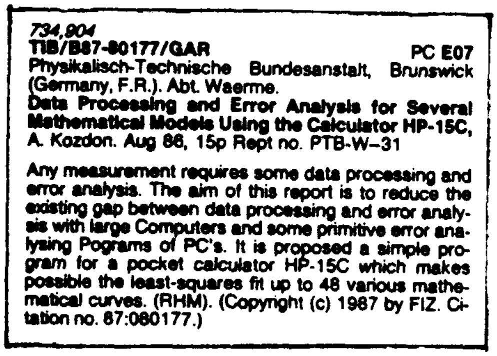
SeTIc has also published a program listing and flow chart for the Mathematics module of the TI-95. There is no explanatory material such as that which was in the so-called "Fish Book" foir the TI-59. The listing comes in a 6 inch by 8 inch loose leaf form. The price is ten dollars ($10.00) including shipping. This wil be a valuable book if you plan to use routines from the module in your programs, or if you want to analyze the routines in the module.

To order these materials send an international postal money order (no checks, please!) to:

Thomas Coppens
P. O. Box 63
2080 Kapellen
Belgium

State whether you want the French or Dutch versions.

FOR BI-LINGUALS - Page 75 of the Volume 87, No. 16 issue of the Government Reports Announcements dated August 15, 1987 contains the listing at the right. The document can be ordered from the NTIS for $20.50 .



734,904
TIB/B87-80177/GAR **PC E07**
Physikalisch-Technische Bundesanstalt, Brunswick (Germany, F.R.). Abt. Waerme.
Data Processing and Error Analysis for Several Mathematical Models Using the Calculator HP-15C,
A. Kozdon. Aug 86, 15p Rept no. PTB-W-31

Any measurement requires some data processing and error analysis. The aim of this report is to reduce the existing gap between data processing and error analysis with large Computers and some primitive error analysing Pograms of PC's. It is proposed a simple program for a pocket calculator HP-15C which makes possible the least-squares fit up to 48 various mathematical curves. (RHM). (Copyright (c) 1987 by FIZ. Citation no. 87:080177.)

An apparently similar capability can be obtained from William Kolb's book Curve Fitting for Programmable Calculators, but not specifically for the HP-15.

INCOMPLETE ADDRESS - The address for Code P on V12N2P15 does not have a city. The complete address should be: James Taylor, P. O. Box 174, Marblehead, MA 01945.

HARDWARE FOR SALE - There continues to be a limited market for used TI-58 and TI-59 hardware. Recent prices are thirty-five dollars for a TI-58 and fifty dollars for a TI-59. The club can continue to provide used TI-59's for fifty dollars plus shipping. Send sixty dollars. We will give fifty dollars to the owner, ship the calculator to you, and return anything remaining. Some of the available units do not have manuals. Let us know if you need the manuals. All used units are provided entirely at the buyers risk.

TI-95 SELF-TEST - Robert Prins. Pages A-4 through A-7 of the TI-95 User's Guide list the system menus. However, one menu is missing, the SELFTESTS: menu which has the following options:

END	ends the test menu and generates a MEMORY CLEARED
TST	a general test of the TI-95, including KBD and LCD
EXT	TST + MOD + PTR
CYC	a loop through EXT
MOD	module
PTR	printer
KBD	keyboard
LCD	display

WARNING

All test functions completely destroy user memory, so before you start using them you must safeguard the entire contents of your machine.

The method which was originally used to access this menu used the following sequence:

Command	Press	Display
1. Switch the machine on	ON	TI-95 PROCALC
2. Select SYSTEM MODE	FUNC SYS YES	SYSTEM FUNCTIONS STB RCB SBA
3. Select UNFORMATTED MODE	CONV BAS UNF	0000000000000000 DEC HEX OCT 2sC UNF
4. Return to the CONV menu	CONV	0000000000000000 MET DMS ANG P-R BAS
5. Enter 0000090003143002	0000090003143002	0000090003143002
6. Store this number in R048	STO 2048	0000090003143002
7. Select the SELFTESTS menu	F5 (BAS)	SELFTESTS: END TST EXT CYC -->
8. To see the other options	F5 (-->)	SELFTESTS: MOD PTR KBD LCD -->

9. Select the mode desired.

You should use this method at least once, particularly if you have never used the systems and unformatted modes; however, in April Robert J. Roeloffzen discovered a much easier way, and probably the "official" way, of calling up the SELFTEST: menu. Press TESTS and HELP and hold both keys down. Then press ALPHA and see "SELFTESTS:" in the display. Release all three keys and see the menu in the windows.

TI-95 Self-test - (cont)

Editor's Note: While these methods provide access to the self-test menu, it is not so clear exactly what is tested by some of the options. Here is what my examinations show so far:

END provides an exit from the SELFTESTS: menu, clears memory and returns the calculator to normal partitioning.

PTR prints the sixteen character line AbCdEfGhIjKlMnO three times and then stops with the message "TEST DONE" in the display.

KBD brings the message "PRESS: F1" to the display. Press F1 and the message changes to "PRESS: F2". Continue pressing keys according to the prompts in the display. After pressing the divide key the message changes to "PRESS: BR". Press the BREAK key and continue with the second through fifth rows in response to the display. Note that the CLEAR key is tested in both the fourth and fifth row. After pressing the = key the display reads "TEST DONE". Some experimentation will show that pressing the HALT key at any time in the sequence will return to the menu. If you press any key other than that asked for by the display or the HALT key the display will not change; therefore, if there is more than one defective key you can only find the first one in the sequence with this routine.

LCD turns on all the status indicators and all the dots in the alphanumeric display and in the function key labels. Press any key and some of the status indicators turn off and the alphanumeric patterns change. Continue pressing any key and see more changes in the display. After the fifth key is pressed the display returns to the "all on" condition. After the sixth key is pressed the status indicators become normal and "TEST DONE" appears in the display.

TST starts with some sort of routine in which progress is indicated with the word "WORKING" moving across the display. Operation stops at the first stage of the LCD mode. After any six keys have been pressed operation is at the first step of the KBD mode. After the user has completed the KBD mode the message "TEST DONE" appears in the display. If HALT is pressed while the message "WORKING" is in the display, then a RCL XXX, where XXX is in the range set by the partitioning will return the value 5AB469D2A54B962D if the calculator is in the unformatted mode.

CYC performs some sort of test similar to the first part of TST except that five additional iterations of the word "WORKING" appear, then performs the printouts of PRT, and repeats the two operations for as long as you let it.

MOD flashes the work "WORKING" at several locations in the display. It stops with the words "8K TEST DONE" in the display if a RAM module is installed. The cartridge name is changed to "NEW" and the cartridge memory is cleared. The mode stops with the words "TEST DONE" if a library module is installed.

TI-66 FOR SALE - Write to Mr. D. Goss, 18004 Fonthill Ave.,
Torrance CA 90504

TI-95 CALENDAR CHALLENGE - Hewlett Ladd. The printing of a calendar has been one of our longstanding benchmark tests of a calculator/printer capability. The earliest calendar printing program for the TI-59/PC-100 appeared in V3N5 of 52 Notes in May 1978. It would print a year in 26 minutes.

A series of program enhancements over the subsequent six years eventually led to Patrick Acosta's program from V9N2P7. It will print a full year on the TI-59/PC-100 in 83 seconds, an average of only seven seconds per month. Individual months can be printed in about nine seconds. The TI-59 program relies on fast mode and intricate use of HIR commands to attain those speeds. It requires 398 program steps and 65 data registers.

A calendar program for the CC-40/HX-1000 appeared in V9N5P8. A printout of a single month required about 21 seconds, much slower than the TI-59/PC-100. The slow speed was associated with the HX-1000 Printer/Plotter which draws each character in order.

V12N2P6 presented a calendar printing program for the TI-74/PC-324. Printout of a single month requires about nine seconds, comparable to the TI-59/PC-100. Note that this program cannot be used with a TI-74/HX-1000 combination and the cable defined on V12N3P13 since the printout is 20 characters wide, while the HX-1000 can only handle 18 full-size characters.

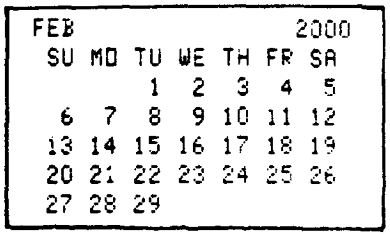
The program which follows provides a calendar printout for the TI-95/PC-324.

```
0000 CMS ADV GTO 0272
0005 NOP NOP NOP NOP NOP
0010 ` JAN` GTO 0223 NOP
0018 NOP NOP SF 02 ` FE`
0025 `B` GTO 0223 NOP
0030 ` MAR` GTO 0223 NOP
0038 NOP NOP SF 01 ` AP`
0045 `R` GTO 0223 NOP
0050 ` MAY` GTO 0223 NOP
0058 NOP NOP SF 01 ` JU`
0065 `N` GTO 0223 NOP
0070 ` JUL` GTO 0223 NOP
0078 NOP NOP ` AUG`
0084 GTO 0223 NOP NOP
0089 NOP SF 01 ` SEP`
0096 GTO 0223 NOP ` OCT`
0104 GTO 0223 NOP NOP
0109 NOP SF 01 ` NOV`
0116 GTO 0223 NOP ` DEC`
0124 GTO 0223 NOP COL 05
0130 MRG B COL 05 ` `
0135 COL 08 MRG C COL 08
0141 ` ` COL 11 MRG D
0146 COL 11 ` ` COL 14
0151 MRG E COL 14 ` `
0156 COL 17 MRG F COL 17
0162 ` ` COL 20 MRG G
0167 COL 20 ` ` COL 23
0172 MRG H COL 23 ` `
0177 PRT RTN COL 20
0181 MRG G COL 20 ` `
0186 COL 17 MRG F COL 17
0192 ` ` COL 14 MRG E
0197 COL 14 ` ` COL 11
0202 MRG D COL 11 ` `
0207 COL 08 MRG C COL 08
0213 ` ` COL 05 MRG B
0218 COL 05 ` ` PRT RTN
0223 COL 24 MRG L COL 24
0229 ` ` PRT RTN CE `  `
0235 `SU MO TU WE TH FR`
0252 ` SA` PRT RTN RCL K
0259 STO IND A INC A
0264 INC K DSZ I
0268 GTO 0257 RTN CE `Y`
0274 `YYY.MM?` BRK STO Q
0284 - INT STO L = ABS *
0291 100= STO M 3 INV
0299 IF> M GTO 0311 INV
0305 INC L 12 ST+ M 1582
0315 .10 IF> Q GTO 0345
0323 RCL L /100= INT
0331 STO R /4= INT -
0338 RCL R +2= STO S 0
0346 INV IF> L GTO 0371
0352 (365.25* RCL L -.75
0366 ) INT GTO 0382 365.
0375 25* RCL L ) INT +((
0385 RCL M +1)*30.6001)
0399 INT +1720994.5+
0411 RCL S +3.5-(/7) INT
0423 *7= INV IF= O
0429 GTO 0433 7 STO N
0435 STO P 12 INV IF< M
0442 GTO 0449 ST- M
0447 INC L 10 ST* M
0453 SBR IND M NOP NOP
0458 NOP SBR 0232 RCL N
0464 STO A -8= ABS STO I
0472 1 STO K SBR 0257 7*
0480 (- RCL N )-170= ABS
0491 STO A CE SBR IND A
0497 4 STO J INV DSZ J
0503 GTO 0521 7 STO I 1
0510 STO A SBR 0257
0515 SBR 0128 GTO 0500
0521 TF 02 GTO 0627
0526 TF 01 GTO 0570 3
0532 ST+ P 7 IF< P
0537 GTO 0580 7 STO I 1
0544 STO A SBR 0257 7*(-
0553 RCL P )+179= STO A
0563 CE SBR IND A
0567 GTO 0622 2 ST+ P 7
0574 INV IF< P GTO 0540
0580 7 STO I 1 STO A
0586 SBR 0257 SBR 0128 8
0593 IF= P GTO 0622 2
0599 STO I 1 STO A
0604 SBR 0257 9 IF= P
0610 GTO 0619 SBR 0207
0616 GTO 0622 SBR 0214
0622 ADV CFG GTO 0000 15
0629 83 IF> L GTO 0622
0636 RCL L /400= FRC
0644 IF= O GTO 0682
0649 RCL L /100= FRC
0657 IF= O GTO 0673
0662 RCL L /4= FRC IF= O
0670 GTO 0682 1 IF= P
0676 GTO 0622 GTO 0540 1
0683 ST+ P 8 IF= P
0688 GTO 0580 GTO 0540
```

TI-95 Calendar Challenge (cont)

User Instructions:

Run the program and see the prompt YYYY.MM in the display and GO above F1. Enter the year and month in the format shown, press F1 and wait about 11 seconds for the printout. A sample printout appears at the right.



```
FEB                2000
 SU MO TU WE TH FR SA
        1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29
```

The program uses 694 program steps and 19 data registers. The previous programs would only provide printout of the Gregorian calendar, that is, after October 1582. This TI-95 program also provides printout for the Julian calendar for the period from 1 AD through September 1582. It does not provide a correct printout for October 1582, the month which has ten days eliminated such that Thursday, October 4 is followed by Friday, October 15. If you call for that month you get a printout as if it were a month in the Gregorian calendar.

One challenge is to reduce the amount of memory required. To compare programs we will assume that one data register is equivalent to eight program steps.

A second challenge is to reduce execution time.

SUPPLEMENTAL BOOK REVIEW - Engineering Statistics with a Programmable Calculator by William Volk (1982, McGraw-Hill). This book was favorably reviewed by W. J. Widmer in V7N7/8P15 and by George Booth in V9N6P3. Walter Bodenmuller reported problems running the Regression-2 program on pages 334-335 and asked for help. The editor found that steps 189/190 should be SBR CE rather than the GTO CLR listed in the book; furthermore, there were misprints in the example accompanying the program. V12N2P23 asked whether readers had encountered other problems with the programs in the book.

W. J. Widmer writes in response: "V12N2P23 noted two errors in Volk's book and which I had not caught in my earlier review (V7N7/8P15). In addition to the two errors cited by P. Hanson (one of which--155.71 in place of 115.71 on page 152--is actually the intercept rather than the slope as stated in V12N2P23), I have found one more: in 2.4.1 on page 16 the number given as 1.0793 should be 1.07193; this is 1 plus interest 0.07193.

The error in steps 189,190 on page 335 is puzzling since this is given as a printer print-out and would not have worked in the first place. I can only guess that twixt program input and printout the incorrect keys were pressed--I've noticed that my own PC-100 is very sensitive to even the gentlest of key 'brushing'! But these are the only errors I've encountered in using many (not all) of the programs in Volk's book, and I do not view the book as 'suspect. "

Editor's Note: Page 24 of EduCALC catalog issue #39 lists the hard cover version of the book for #31.95. The problem of publishing bad program listings is an old one. In V8N1P2 I noted that one way to avoid that is to perform a time-consuming, boring key-in of the program from the final copy which is to go to the printer. Unfortunately, as the time to go to press approaches I tend to take shortcuts. The errors in the Volk book show that others do the same.

SOLUTION FOR A QUARTIC - Peter Messer and Hewlett Ladd.

This program for the solution of a quartic equation demonstrates the use of the QAD and CUB functions in a program. The instructions for the use of those functions in a program appear on pages 2-36 and 2-38 of the TI-95 User's Guide and on page 2-9 of the TI-95 Programming Guide. In a program such as this where it is necessary to know the nature of the roots the important information is that:

* For the quadratic solution the coefficients must be placed in data registers 000 through 002 before the QAD function is called. After the QAD function is complete the roots appear in data registers 000 and 001, and data register contains a 0 if the roots are real or contains a 1 if the roots are complex.

* For the cubic solution the coefficients must be placed in data registers 000 through 003 before the CUB function is called. After the CUB function is complete the roots appear in data registers 000 through 002. One real root appears in data register 002. Either two real roots, or the Re and Im parts of two complex roots appear in data registers 000 and 002. If all three roots are real then data register contains a 0. If two roots are complex then data register 003 contains a 1.

Given a quartic which has been reduced to the form where 1 is the coefficient of the fourth degree term, i.e.

$$f(x) = x^4 + bx^3 + cx^2 + dx + e = 0$$

Let R be a real root of the resolvent cubic:

$$Ay^3 - By^2 + Cy + D = 0$$

where $A = 1$, $B = c$, $C = (bd - 4e)$ and $D = (4ce - d^2 - b^2e)$

Use the CUB function to solve the cubic for a real root R. For 3 real roots, let R = R1; for one real root, let R = R3.

Calculate $p = SQR(b^2 - 4c + 4R)$, $q = SQR(R^2 - 4e)$ and $S = (bR - 2d)$.

If $S <= 0$ then use the QAD function to solve

$$2Z^2 + (b + p)Z + (R - q) = 0 \quad \text{and} \quad 2Z^2 + (b - p)Z + (R + q) = 0$$

If $S > 0$ then use the QAD function to solve

$$2Z^2 + (b + p)Z + (R + q) = 0 \quad \text{and} \quad 2Z^2 + (b - p)Z + (R - q) = 0$$

This solution of a quartic by first solving a resolvent cubic equation is known as Ferrari's solution. A derivation appears in Dickson's First Course in the Theory of Equations. The derived equatons appear in paragraph 3.8.3 of the Handbook of Mathematical Functions (AMS 55).

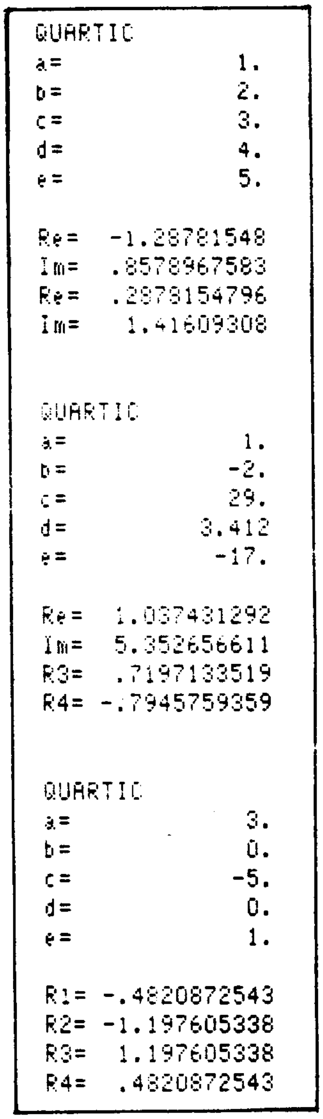
To use the program enter the coefficients of the quartic in accordance with the prompts in the windows. The coefficient of the x^4 term, a, does not have to be one; the program makes the adjustment. The coefficients a through d can be entered in any order. When the coefficient e is entered the solution starts.

Solution for a Quartic - (cont)

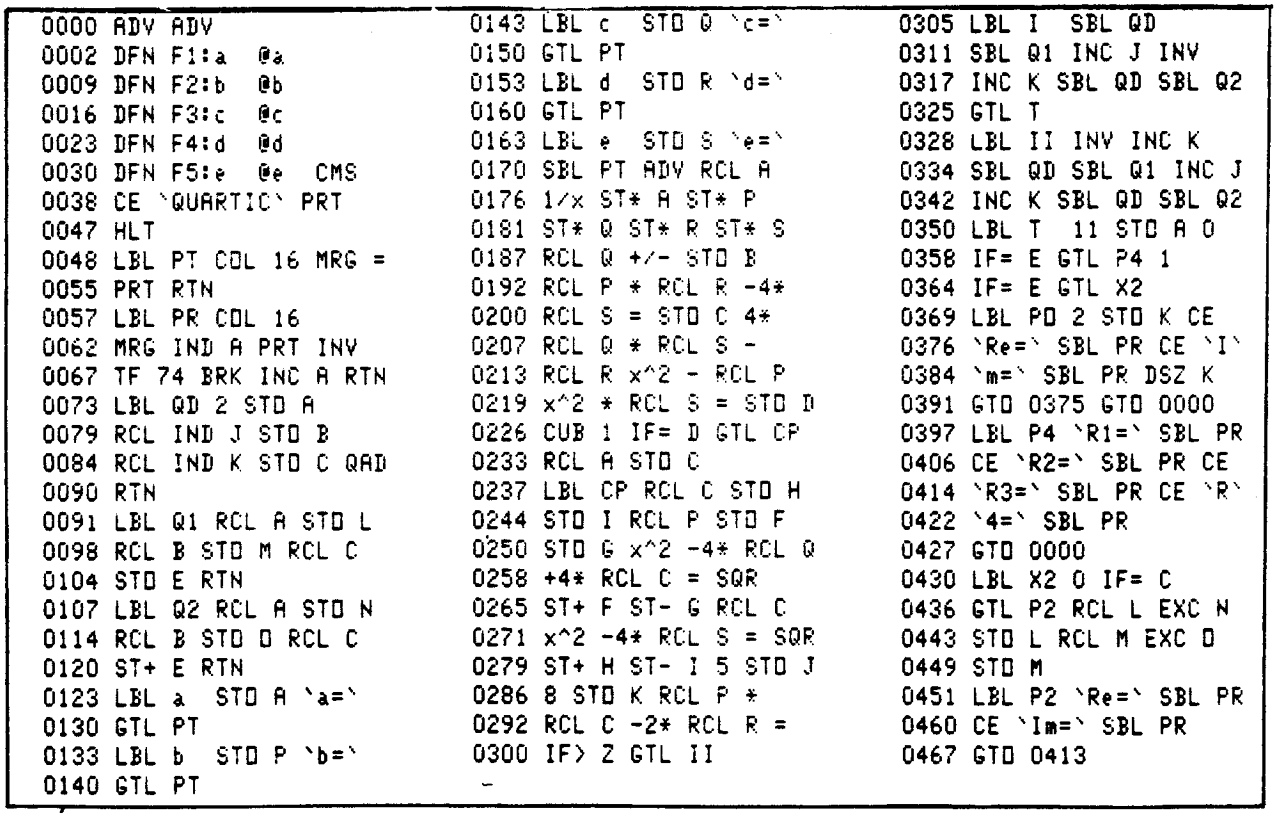
Editor's Note: Printout of the solutions of three different quartics apears at the right. Note that for complex roots the + and - signs for the imaginary parts must be supplied by the user.

This is an example of the kind of cooperation the club has been able to generate in the development of a program. Peter identified the equations to be mechanized and defined the program flow, Hewlett did the major portion of the programming, and the editor assisted with the integration of the QAD and CUB functions in the program.

The use of the CUB function in a program was demonstrated earlier in another quartic solution by William Hawes. That quartic solution demonstrated a call of the math library module as well--see item 2 on V12N1P26. The Hawes program uses 960 bytes while the Messer/Ladd program requires only 472 bytes. For the three problems illustrated at the right the two programs yield the same results to at least the number of places in the display. The Hawes solution requires about twice as long to run, but it yields better solutions in some difficult cases. For example, consider the quartic $3x^4 + 5x^2 + 1 = 0$ which has roots with imaginary parts only. The Messer/Ladd program will show an "ERROR", but will print out a correct (?) solution if the real portions (+/- 1E-12) are considered to be equivalent to zero. The Hawes solution obtains exactly zero for the real parts. The listing for the Messer/Ladd program follows. A listing of the Hawes program appears on page 10.



```
QUARTIC
a=             1.
b=             2.
c=             3.
d=             4.
e=             5.

Re=  -1.28781548
Im=  .8578967583
Re=  .2878154796
Im=   1.41609308


QUARTIC
a=             1.
b=            -2.
c=            29.
d=          3.412
e=           -17.

Re=  1.037431292
Im=  5.352656611
R3=  .7197133519
R4= -.7945759359


QUARTIC
a=             3.
b=             0.
c=            -5.
d=             0.
e=             1.

R1= -.4820872543
R2= -1.197605338
R3=  1.197605338
R4=  .4820872543
```



```
0000 ADV ADV
0002 DFN F1:a  @a
0009 DFN F2:b  @b
0016 DFN F3:c  @c
0023 DFN F4:d  @d
0030 DFN F5:e  @e  CMS
0038 CE `QUARTIC` PRT
0047 HLT
0048 LBL PT COL 16 MRG =
0055 PRT RTN
0057 LBL PR COL 16
0062 MRG IND A PRT INV
0067 TF 74 BRK INC A RTN
0073 LBL QD 2 STO A
0079 RCL IND J STO B
0084 RCL IND K STO C QAD
0090 RTN
0091 LBL Q1 RCL A STO L
0098 RCL B STO M RCL C
0104 STO E RTN
0107 LBL Q2 RCL A STO N
0114 RCL B STO O RCL C
0120 ST+ E RTN
0123 LBL a  STO A `a=`
0130 GTL PT
0133 LBL b  STO P `b=`
0140 GTL PT
0143 LBL c  STO Q `c=`
0150 GTL PT
0153 LBL d  STO R `d=`
0160 GTL PT
0163 LBL e  STO S `e=`
0170 SBL PT ADV RCL A
0176 1/x ST* A ST* P
0181 ST* Q ST* R ST* S
0187 RCL Q +/- STO B
0192 RCL P * RCL R -4*
0200 RCL S = STO C 4*
0207 RCL Q * RCL S -
0213 RCL R x^2 - RCL P
0219 x^2 * RCL S = STO D
0226 CUB 1 IF= D GTL CP
0233 RCL A STO C
0237 LBL CP RCL C STO H
0244 STO I RCL P STO F
0250 STO G x^2 -4* RCL Q
0258 +4* RCL C = SQR
0265 ST+ F ST- G RCL C
0271 x^2 -4* RCL S = SQR
0279 ST+ H ST- I 5 STO J
0286 8 STO K RCL P *
0292 RCL C -2* RCL R =
0300 IF> Z GTL II
0305 LBL I  SBL QD
0311 SBL Q1 INC J INV
0317 INC K SBL QD SBL Q2
0325 GTL T
0328 LBL II INV INC K
0334 SBL QD SBL Q1 INC J
0342 INC K SBL QD SBL Q2
0350 LBL T  11 STO A 0
0358 IF= E GTL P4 1
0364 IF= E GTL X2
0369 LBL PO 2 STO K CE
0376 `Re=` SBL PR CE `I`
0384 `m=` SBL PR DSZ K
0391 GTO 0375 GTO 0000
0397 LBL P4 `R1=` SBL PR
0406 CE `R2=` SBL PR CE
0414 `R3=` SBL PR CE `R`
0422 `4=` SBL PR
0427 GTO 0000
0430 LBL X2 0 IF= C
0436 GTL P2 RCL L EXC N
0443 STO L RCL M EXC O
0449 STO M
0451 LBL P2 `Re=` SBL PR
0460 CE `Im=` SBL PR
0467 GTO 0413
```

Solution for a Quartic - (cont)

Listing for the Hawes Program

```
0000 `QRT V.Q17 18 APR `
0017 `87 W HAWES`
0027 LBL S1 0 STD 100 CE
0035 `QUARTIC`
0042 DFN F1:  a@aa
0049 DFN F2:  b@bb
0056 DFN F3:  c@cc
0063 DFN F4:  d@dd
0070 DFN F5:  e@ee HLT
0078 LBL aa STD S `a =`
0086 COL 16 MRG = HLT
0091 LBL bb STD T `b =`
0099 COL 16 MRG = HLT
0104 LBL cc STD U `c =`
0112 COL 16 MRG = HLT
0117 LBL dd STD V `d =`
0125 COL 16 MRG = HLT
0130 LBL ee STD W `e =`
0138 COL 16 MRG = CE `E`
0144 `DIT?` Y/N GTL S1
0152 DFN CLR CE `WORKIN`
0161 `G` PAU RCL T /
0166 RCL S = STD L RCL U
0173 / RCL S = STD M
0179 RCL V / RCL S =
0185 STD N RCL W / RCL S
0192 = STD O DFN CLR 1
0198 STD 000 RCL M +/-
0204 STD 001 (( RCL L *
0212 RCL N )-(4* RCL O )
0222 ) STD 002 (( RCL L
0230 x^2 * RCL O ) +/- +
0237 (4* RCL M * RCL O )
0246 - RCL N x^2 )
0251 STD 003 CUB RCL 002
0258 STD Q (( RCL L x^2
0265 /4)- RCL M +
0272 RCL 002 ) STD R HEX
0279 7713 STD 027 766C
0290 STD J DEC 1 EE 6
0296 +/- STD Z INV EE
0301 RCL R ABS IF> Z
0306 GTL NZ
0309 LBL RZ 0 STD 028
0316 STD 029 SBL CX
0322 STD X ( RCL Q x^2 -
0329 4* RCL O ) STD Y
0336 STD 000 0 STD 001
0343 SBL SR *2= STD 030
0352 RCL 001 *2= STD 031
0361 SBL CD SBL CE
0367 SBL CR GTL DR
0373 LBL CR RCL L +/- /4
0381 =+ RCL 028 /2=+
0390 RCL 032 /2= STD 036
0399 RCL 029 /2=+
0406 RCL 033 /2= STD 037
0415 RCL L +/- /4=+
0422 RCL 028 /2=-
0429 RCL 032 /2= STD 038
0438 RCL 029 /2=-
0445 RCL 033 /2= STD 039
0454 RCL L +/- /4=-
0461 RCL 028 /2=+
0468 RCL 034 /2= STD 040
0477 RCL 029 +/- /2=+
0485 RCL 035 /2= STD 041
0494 RCL L +/- /4=-
0501 RCL 028 /2=-
0508 RCL 034 /2= STD 042
0517 RCL 029 +/- /2=-
0525 RCL 035 /2= STD 043
0534 RTN
0535 LBL SR CE `MTHPLX `
0546 RUN SBR IND 027
0551 RCL 000 RTN
0555 LBL CX ( RCL L x^2
0562 *3/4)-2* RCL M =
0573 RTN
0574 LBL CD RCL X +
0580 RCL 030 = STD 000
0587 RCL 031 STD 001
0593 SBL SR STD 032
0599 RCL 001 STD 033 RTN
0606 LBL CE RCL X -
0612 RCL 030 = STD 000
0619 RCL 031 +/- STD 001
0626 SBL SR STD 034
0632 RCL 001 STD 035 RTN
0639 LBL NZ RCL R
0644 STD 000 0 STD 001
0651 SBL SR STD 028
0657 RCL 001 STD 029
0663 SBL CX - RCL R =
0670 STD X (4* RCL L *
0678 RCL M -8* RCL N -(
0687 RCL L y^x 3))
0693 STD 000 0 STD 001 4
0701 * RCL 028 = STD 002
0709 4* RCL 029 =
0715 STD 003 CE `MTHPLX`
0725 RUN SBR IND J
0729 RCL 000 STD 030
0735 RCL 001 STD 031
0741 SBL CD SBL CE
0747 SBL CR GTL DR
0753 LBL DR CE DFN CLR
0759 DFN F1:Re1@R1
0766 DFN F2:Re2@R2
0773 DFN F3:Re3@R3
0780 DFN F4:Re4@R4
0787 DFN F5:EDD@ED
0794 LBL R1 CE `Re1`
0801 COL 16 MRG 036
0806 DFN F1:Im1@I1 HLT
0814 LBL I1 CE `Im1`
0821 COL 16 MRG 037
0826 DFN F1:Re1@R1 HLT
0834 LBL R2 CE `Re2`
0841 COL 16 MRG 038
0846 DFN F2:Im2@I2 HLT
0854 LBL I2 CE `Im2`
0861 COL 16 MRG 039
0866 DFN F2:Re2@R2 HLT
0874 LBL R3 CE `Re3`
0881 COL 16 MRG 040
0886 DFN F3:Im3@I3 HLT
0894 LBL I3 CE `Im3`
0901 COL 16 MRG 041
0906 DFN F3:Re3@R3 HLT
0914 LBL R4 CE `Re4`
0921 COL 16 MRG 042
0926 DFN F4:Im4@I4 HLT
0934 LBL I4 CE `Im4`
0941 COL 16 MRG 043
0946 DFN F4:Re4@R4 HLT
0954 LBL ED GTL S1
```

To use the Hawes program enter the coefficients in response to the prompts in the windows. You may enter coefficients a through d in any order. After you enter coefficient e you are given an opportunity to change ("edit") any of the input coefficients. A "No" response to the edit question starts the solution. The program does not provide printout of either the input values or of the solution. The prompts in the windows allow you to view the real or imaginary parts as many times as you wish. There is no direct indication of real roots. You must determine that the roots are real by the absence of an imaginary part.

MORE ON TI-59 DIAGNOSTICS - Scott Garver writes: On V12N1P4/5 you gave us a good extended diagnostic program for the TI-59; however, for those of us without a printer, it would entail staring at the display for the entire sixteen minutes. A better option is to halt instead of print when an error is found. The following program provides that capability. It also changes the test values from a n/9 sequence to a n/11 sequence which generates a more diversified pattern: 1/11 = .0909..., 2/11 = .1818..., etc.

```
000  76 LBL    012  42 STO    024  32 X:T    036  32 X:T    048  91 R/S    060  01   1
001  11   A    013  00  00    025  94 +/-    037  42 STO    049  97 DSZ    061  01   1
002  01   1    014  32 X:T    026  63 EX*    038  00  00    050  00  00    062  95   =
003  00   0    015  72 ST*    027  00  00    039  73 RC*    051  00  00    063  22 INV
004  69 OP     016  00  00    028  63 EX*    040  00  00    052  39  39    064  77  GE
005  17  17    017  97 DSZ    029  00  00    041  29 CP     053  01   1    065  00  00
006  29 CP     018  00  00    030  74 SM*    042  67  EQ    054  82 HIR    066  08  08
007  37 P/R    019  00  00    031  00  00    043  00  00    055  37  37    067  06   6
008  47 CMS    020  15  15    032  97 DSZ    044  49  49    056  32 X:T    068  69 OP
009  32 X:T    021  32 X:T    033  00  00    045  32 X:T    057  82 HIR    069  17  17
010  09   9    022  42 STO    034  00  00    046  43 RCL    058  17  17    070  91 R/S
011  09   9    023  00  00    035  26  26    047  00  00    059  55   ÷    071  00   0
```

To run the program press A. If an error occurs the test stops. The display shows the failed register. The contents of the failed register will be in the t register. Press R/S to continue the test. When the test is complete the default partitioning is in the display. The running time is about 21 minutes.

A.O.S. CALCULATION FROM H.P. - W. J. Widmer. A comment by the editor in V12N3P23 that "old-timers" ... have to be surprised at an advertisement for an HP machine which states "It calculates with Algebraic Logic, dealing with equations just as you write them." rather misses the point. This is no surprise and simply involves the difference between "how you write an equation" and "how you perform the calculation"--which is the basic difference between A.O.S. and R.P.N. That is, you may write or express a multiplication as 625 x 321 = 200625; but you perform this mentally (or were you to do it long-hand) as

```
                                                 625
   625 enter 321 --> 200625          or          321
                                              200625
```

Either system works well and preference centers on how you want to show the problem or how you want to do the work. Both are good systems and the old argument is simply an internal mental problem (even TI does RPN with internal hidden storage!). People are trained from grade school on up through professional life to write the problem first, then do the work! Surprise? No; in business, HP--like any company--caters to the market mind.

ELEKTEK STILL CARRIES TI-59 SUPPLIES - Elek-Tek has been one of the few continuing sources of magnetic cards, printer paper and batteries for the TI-59, PC-100 and other older TI programmable calculators. However, their latest catalog (Volume 15) no longer lists these supplies. I called Elek-Tek at the toll-free number (1-800-621-1269) and was told that they often continue to carry items not listed in the catalog. Call them for the latest availability before you order.

<u>ANOTHER</u> <u>SIMULTANEOUS</u> <u>EQUATION</u> <u>SOLUTION</u> - In V12N3P4-7 I reviewed three books of BASIC programs which were co-authored by former TI PPC Notes editor Maurice Swinnen. The programs are written in Sharp BASIC which contains some unique commands which are not available with the TI-74. An example conversion of a program from the <u>Statistics</u> <u>Library</u> was provided.

Although the contents of the <u>Mathematics</u> <u>Library</u> book were listed on V12N3P5, the book did not become available until after V12N3 was printed. When it arrived I converted the simultaneous equations program for use with the TI-74 so I could compare its capability with similar programs published in earlier issues. The listing for the converted program on page 13 was made using the HX-1000 and a cable like that described in V12N3P13. Comments on the conversion follow:

Lines 10 through 210 set the dimensioning, call the subroutine which selects the printer options, and provide for data input. Appropriate changes have been made to accommodate the differences between machines.

All of the programs in the <u>Mathematics</u> <u>Library</u> use the statement GOSUB "PRINTER?" in line 20 to call the printer option subroutine, and the first line of the subroutine is the statement "PRINTER?". This indicates that Sharp BASIC has a label capability that is not available on the TI-74 or CC-40. The GOSUB 800 statement at line 30 of the conversion provides the equivalent result. The programs in the <u>Statistics</u> <u>Library</u> and the <u>Electrical</u> <u>Engineering</u> <u>Library</u> did not use the label capability.

Lines 220 through 720 which mechanize the solution equations are nearly identical to the program in the book. The only changes are the replacement of GOTO with THEN in lines 230, 240, 350, 390, 420, 590, 710 and 720.

Lines 730 through 790 provide for output of the solution. Again, appropriate changes were made to accommodate the differences between machines.

Line 800 through 890 provide for selection of the printer option. The subroutine provides prompting for use of either the PC-324 or HX-1000. Line 880 selects the compressed print (36 characters per line) option of the HX-1000 to avoid the wraparound which would occur with the 18 character per line normal mode.

As with the conversion in V12N3P6-7 the BEEP 2 statements which appear in the book were deleted since the TI-74 does not have a BEEP capability. CC-40 users can replace the BEEP 2 statements with DISPLAY BEEP.

Lines 790 through 810 and 910 through 930 in the program in the book provide an option to solve another problem without going through the printer selection process again. That capability was not provided in the translation. To solve another problem with the translation simply run the program again. The RUN command zeroes all the variables.

A full set of prompts are provided with the program. Each equation is entered in order. The matrix of coefficients of the variables is stored in the two dimensional A array. The vector of constants is stored in the one dimensional B array. The solutions are derived in the X array.

Another Simultaneous Equation Solution - (cont)

Program Listing:

```
10 A$="Simultaneou
s Equations":PRINT
 A$:PAUSE 2
20 DIM A(22,22),B(
22),X(22)
30 GOSUB 800
40 IF PF=0 THEN 70
50 PRINT #1,A$
60 PRINT #1
70 INPUT "Number o
f equations = ? ";
N
100 FOR I=1 TO N
110 FOR J=1 TO N
120 II$=STR$(I):JJ
$=STR$(J):AA$="A("
&II$&","&JJ$&")= "
140 INPUT AA$;A(I,
J)
150 PRINT #PF,AA$;
A(I,J)
160 NEXT J
170 BB$="B("&II$&"
)= "
190 INPUT BB$;B(I)
200 PRINT #PF,BB$;
B(I)
210 PRINT #PF:NEXT
 I
220 Z=0
230 IF N<>0 THEN 2
90
240 IF A(1,1)=0 TH
EN 270
250 X(1)=B(1)/A(1,
1)
260 GOTO 720
270 Z=1
280 GOTO 720
290 M=N-1
300 FOR I=1 TO M
310 BC=ABS(A(I,I))
320 L=I
330 IJ=I+1
340 FOR J=IJ TO N
350 IF ABS(A(J,I))
<BC THEN 380
360 BC=ABS(A(J,I))
370 L=J
380 NEXT J
390 IF BC<>0 THEN
420
400 Z=1
410 GOTO 720
420 IF L=I THEN 51
0
430 FOR J=I TO N
440 G=A(L,J)
450 A(L,J)=A(I,J)
460 A(I,J)=G
470 NEXT J
480 G=B(L)
490 B(L)=B(I)
500 B(I)=G
510 FOR J=IJ TO N
520 T=A(J,I)/A(I,I
)
530 FOR K=IJ TO N
540 A(J,K)=A(J,K)-
T*A(I,K)
550 NEXT K
560 B(J)=B(J)-T*B(
I)
570 NEXT J
580 NEXT I
590 IF A(N,N)<>0 T
HEN 620
600 Z=1
610 GOTO 720
620 X(N)=B(N)/A(N,
N)
630 I=N-1
640 S=0
650 IJ=I+1
660 FOR J=IJ TO N
670 S=S+A(I,J)*X(J
)
680 NEXT J
690 X(I)=(B(I)-S)/
A(I,I)
700 I=I-1
710 IF I>0 THEN 64
0
720 IF Z<>1 THEN 7
50
730 PRINT #PF,"No
Solution Found":IF
 PF=0 THEN PAUSE
740 GOTO 785
750 PAUSE ALL:FOR
I=1 TO N
760 II$=STR$(I):XX
$="X("&II$&")= "
770 PRINT #PF,XX$;
X(I)
780 NEXT I
785 IF PF=1 THEN P
RINT #1
790 STOP
800 IF PF=1 THEN C
LOSE #1:PF=0
810 INPUT "Use pri
nter? Y/N ";N$
820 IF N$="Y"OR N$
="y"THEN PF=1 ELSE
 890
830 PRINT "Device
Numbers:":PAUSE 2
840 PRINT "For the
 HX-1000 enter 10"
:PAUSE 2
850 PRINT "For the
 PC-324 enter 12":
PAUSE 2
860 INPUT "Enter d
evice number ";D$
870 OPEN #1,D$,OUT
PUT
880 IF D$="10"THEN
 PRINT #1,CHR$(18)
890 PRINT #PF:RETU
RN
999 END
```

Another Simultaneous Equation Solution - (cont)

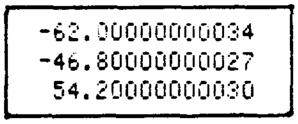
One of the important features of the Swinnen books is the provision of example problems for each program. The two examples for the simultaneous equations program are:

$$12a + 22b + 33c = 15 \qquad \text{and} \qquad b - 2c = -8$$

$$23a + 34b + 56c = 18 \qquad\qquad b + c = 7$$

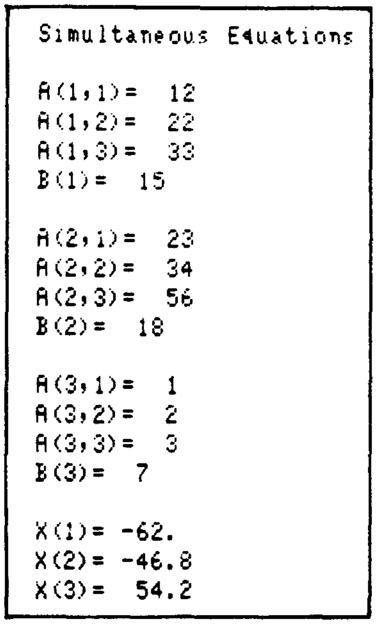
$$a + 2b + 3c = 7 \qquad\qquad 2b - c = 10$$

The exact solution for the first example is a = -62, b = -46.8 and C = 54.2 . Sample printouts for the first problem with both the PC-324 and the HX-1000 are illustrated at the right. The apparently exact solution from the program is a result of the ten digit output format of the print statement. If PRINT USING with the format ###.########### were used the printout would be:

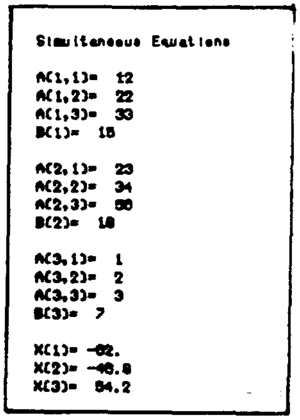


```
-62.00000000034
-46.80000000027
 54.20000000030
```

where those values are about 500 times more accurate than the values shown in the book for one of the Sharp machines.



```
Simultaneous Equations

A(1,1)=  12
A(1,2)=  22
A(1,3)=  33
B(1)=  15

A(2,1)=  23
A(2,2)=  34
A(2,3)=  56
B(2)=  18

A(3,1)=  1
A(3,2)=  2
A(3,3)=  3
B(3)=  7

X(1)= -62.
X(2)= -46.8
X(3)=  54.2
```



```
Simultaneous Equations

A(1,1)=  12
A(1,2)=  22
A(1,3)=  33
B(1)=  15

A(2,1)=  23
A(2,2)=  34
A(2,3)=  56
B(2)=  18

A(3,1)=  1
A(3,2)=  2
A(3,3)=  3
B(3)=  7

X(1)= -62.
X(2)= -46.8
X(3)=  54.2
```

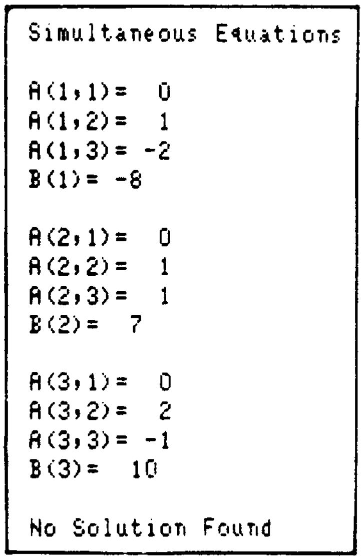
A more demanding test of a simultaneous equation solver is the 7x7 sub-Hilbert proposed by George Thomson in V8N6P18 together with the test of the solution to the sub-Hilbert proposed by James Walters in V9N2P18. The results for this program were:

Exact	Solution	Walters Test
56	56.00068891	1.0000000000
-1512	-1512.016156	0.9999999970
12600	12600.119	1.0000000020
-46200	-46200.391	0.9999999973
83160	83160.63758	0.9999999980
-72072	-72072.50508	0.9999999973
24024	24024.15504	0.9999999983
Max Error	1.23E-05	3.0E-09
RMS Error	9.07E-06	2.2E-09

The shorter row reduction program from V8N6P20 yields an RMS error for the solution of 2.57E-06 and an RMS error for the Walters test of 1.43E-08. The errors from the program from the Mathematica Library are five times larger for the solution and six times smaller for the Walters test. Either program yields results which compare favorably with results from other programs and machines. The row reduction program presented in V8N6P20 was incorporated into the the least square programs on V12N1P14 and V12N2P21.

Another Simultaneous Equation Solution - (cont)

The second example in the book is a test of the ability of the program to recognize indeterminate sets of equations. The printout at the upper right illustrates the message "No Solution Found" for this problem.

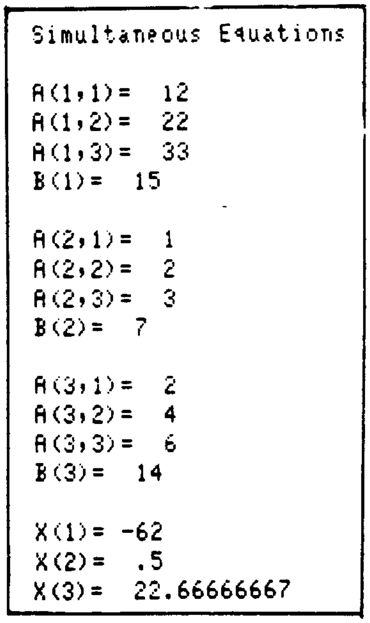


```
Simultaneous Equations

A(1,1)=  0
A(1,2)=  1
A(1,3)= -2
B(1)= -8

A(2,1)=  0
A(2,2)=  1
A(2,3)=  1
B(2)=  7

A(3,1)=  0
A(3,2)=  2
A(3,3)= -1
B(3)=  10

No Solution Found
```

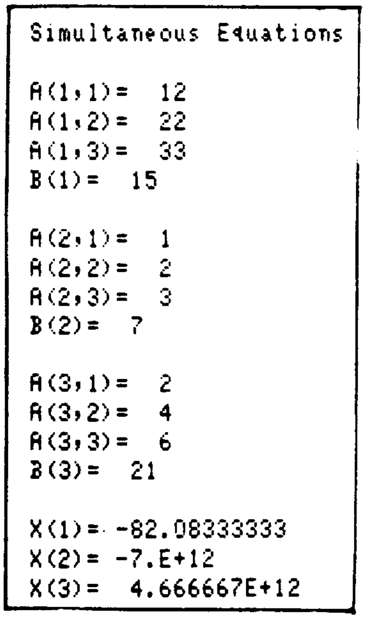
Note that you must enter the zero coefficients in the example. Clearly, the determinant of the matrix of coefficients is zero. What about other cases where the determinant is zero such as

$$\begin{aligned} 12a + 22b + 33c &= 15 \\ 1a + 2b + 3c &= 7 \\ 2a + 4b + 6c &= 14 \end{aligned} \quad \text{and} \quad \begin{aligned} 12a + 22b + 33c &= 15 \\ 1a + 2b + 3c &= 7 \\ 2a + 4b + 6c &= 21 \end{aligned}$$

where we recognize that the system at the left has many solutions and the system at the right has no solution. The middle and lower printouts at the left show the results from the program. How do other linear equation solution programs respond to these indeterminate problems?



```
Simultaneous Equations

A(1,1)=  12
A(1,2)=  22
A(1,3)=  33
B(1)=  15

A(2,1)=  1
A(2,2)=  2
A(2,3)=  3
B(2)=  7

A(3,1)=  2
A(3,2)=  4
A(3,3)=  6
B(3)=  14

X(1)= -62
X(2)=  .5
X(3)=  22.66666667
```



```
Simultaneous Equations

A(1,1)=  12
A(1,2)=  22
A(1,3)=  33
B(1)=  15

A(2,1)=  1
A(2,2)=  2
A(2,3)=  3
B(2)=  7

A(3,1)=  2
A(3,2)=  4
A(3,3)=  6
B(3)=  21

X(1)= -82.08333333
X(2)= -7.E+12
X(3)=  4.666667E+12
```

The Matrices (MAT) program in the TI-74 Mathematics Library gives the message "THE SYSTEM IS SINGULAR" for both problems.

The row reduction program from V8N6P20 yields the message "... Division by Zero" for all three problems where the determinant should be zero.

The Inversion/Linear Systems program in the TI-95 Mathematics Library yields the message "SINGULAR" for the second problem from the book, and the determinant is zero. The determinant for the matrix of coefficients for the two problems above is 1.2e-12, not zero. The problem at the left yields the solution -62, -50.5 and 56.66666667 which is different from that from the program from the book, but is also a solution. The problem at the right yields -132.0833333, -1.75e13 and 1.166667e13 which is not a solution.

The ML-02 program in the TI-59 Master Library yields zero for the determinant for the second problem from the book. The determinant for the matrix of coefficients for the two problems above is -1.2e-12, not zero. The problem at the left yields the solution -62, -60.5 and 63.33333333 which is different from that from the program from the book or from the TI-95 Mathematics Library, but is also a solution. The problem at the right yields -123.75, -1.75e13 and 1.166667e13 which is not a solution.

TRANSLATING PROGRAMS FROM THE TI-59 TO THE TI-95 - Palmer Hanson. Many of our newer members have reported problems in translating TI-59 programs for use on the TI-95 and have asked for a tutorial treatment on that subject. The approach I have taken is to present what I call "brute force" translation. That means that the conversion for the TI-95 emulates the operation of the TI-59 as closely as possible. Once a user has achieved the "brute force" translation he can make modifications as he chooses to use the more powerful features of the TI-95.

Replacing the User Defined Keys

The first step is to emulate the User Defined Keys of the TI-59 (A through E') with a Function-Key menu on the TI-95. The sequence at the right will provide the required function key definitions. The arrow pointing to the right for F5 allows to user to change the definitions in a manner similar to many of the built-in menus.

```
0000 LBL XX
0003 DFN F1: A @AA
0010 DFN F2: B @BB
0017 DFN F3: C @CC
0024 DFN F4: D @DD
0031 DFN F5:-->@YY HLT
0039 LBL YY
0042 DFN F1: E @EE
0049 DFN F2: A'@A'
0056 DFN F3: B'@B'
0063 DFN F4: C'@C'
0070 DFN F5:-->@ZZ HLT
0078 LBL ZZ DFN CLR
0083 DFN F1: D'@D'
0090 DFN F2: E'@E'
0097 DFN F5:-->@XX HLT
```

What remains is to replace any user defined labels in the TI-59 program with the two-character labels defined by the DFN functions. I did not use any single character labels (one character is a space) such as in the conversion of L. Leeds Odds Against program in V11N4P6/7. I have found that I have problems in reading the listing with single character labels.

Truncated versions of this routine can be used where not all of the User Defined Keys are used in a particular program. Examples of truncated versions appear in the Odds Against translation on V11N4P7, in the cubic translation on V11N4P18, and in the ML-20 conversion on page 20 of this issue.

Getting Rid of Absolute Addressing

One of the first things you will discover as you try to translate programs is that absolute addresses simply won't convert easily from the TI-59 to the TI-95. Therefore, in the discussion that follows I will assume that the user has located all of the absolute address calls in the TI-59 program, has replaced the addresses with labels of his choice, and has inserted the labels at the approriate addresses. If the user wants to convert to absolute addresses once the translation is complete he can do so easily with the ASM function.

Tests

Tests for decision making with the TI-59 include Dsz tests, comparision tests, and flag tests. The TI-95 adds Yes/No tests. The only responses of the TI-59 to a test instruction are to exercise an implied GOTO with an address or label immediately following the test, or to skip the address of the implied GOTO and exercise the next instruction. The tests with the TI-95 are more versatile. Any valid instruction can follow the test; however, a GOTO instruction will probably the most frequently used instruction immediately after a test.

Old-timers will remember that machines such as the TI-57 had test functions with a structure like the TI-95. Page 7-6 of Making Tracks into Programming, the manual for the TI-57 said it well: "The step that immediately follows the 2nd Dsz key sequence can be anything, but a GTO n instruction is often handy for setting up repetitive calculations". Page 7-9 of the TI-57 manual contained a similar comment concerning the instruction immediately following a t register test.

Translating Programs from the TI-59 to the TI-95 -- (cont)

An example of the use of an instruction other than a GTO or GTL immediately following a test in a TI-95 program appears as a TF 01 HLT sequence at steps 0272-0274, 0297-0299 and 0351-0353 of the program on page 21. If flag 1 has been set by the error routine at LBL xx, then the TF 01 HLT stops the calculator with the message to "RE-ENTER DATE" message in the display. If an error has not been found flag 1 will be reset and program execution skips the HLT and proceeds to complete the mode. Similar examples appear on pages 5-6 and 5-14 of the TI-95 Programming Guide. Of course, these capabilities are not important for "brute force" translations since the exact equivalent can not occur in TI-59 programs.

Translating Dsz Tests

The TI-59 Dsz instruction is in the format Dsz XX N where XX is a data register and N is any valid label used by the implied GTO. The label can be replaced by an nnn address. For the TI-95 the Dsz register can be defined by three digits or by a letter, and a GTL or GTO follows to mechanize the transfer if the contents of the Dsz register is not zero. Thus,

Dsz XX N in a TI-59 converts to Dsz XXX GTL NN in a TI-95

where NN is a two-character label for the TI-95. Of course, the three digit register notation for the TI-95 can be replaced by a one character letter for data registers 000 through 025.

Translating t Register Tests

The only comparison tests available with the TI-59 involve the display register and the t register. Comparison tests are available in the TI-95 between the display register and any data register, but NOT with the t register. The ability of the TI-95 to perform a comparison test between the display register and any data register is certainly a more powerful capability; however, the designers of the TI-95 would have done translation from the TI-59 to the TI-95 a real favor by retaining the t register test capability.

Since the t register is not available for comparison tests with the TI-95 the translator must designate a data register to take its place. In the following table of translations it is assumed that data register A, also addressable as register 000, has been selected. Of course, any other data register may be used provided it is compatible with the remainder of the translation. Again, the TI-59 instructions are at the left, and the equivalent TI-95 instructions are at the right:

x⇄t	EXC A
CP	EXC A 0 EXC A
x=t N	IF= A GTL NN
x≥t N	INV IF< A GTL NN
INV x=t N	INV IF= A GTL NN
INV x≥t N	IF< A GTL NN

where N is any valid TI-59 label, and NN is any valid two-character TI-95 label. Of course the labels can be replaced by the appropriate absolute addressing.

Translating Programs from the TI-59 to the TI-95 - (cont)

Translating Flag Operations

The RST function in the TI-59 resets all flags, but it also clears the subroutine return register and moves the program pointer to the origin. There is no single function which will yield the same results with the TI-95. All TI-95 user flags can be reset by the CFG function. Also, Scott Garver observes that STB 205A will clear the user subroutine counter (see page C-18 of the TI-95 Programming Guide).

The Stflg X function with the TI-59 can be replaced with SF XX for the TI-95. The only change is the two digit flag designation of the TI-95 which allows fourteen user flags versus the ten of the TI-59.

The Ifflg X N function with the TI-59 tests flag X and goes to label N if the flag is set. TF XX GTL NN provides the same capability with the TI-95.

Clearing the Statistics Registers

In the TI-59 the sequence Pgm 01 SBR CLR could be used to initialize the calculator for the statistics functions. The subroutine that was called was provided in every Solid State Software module except the RPN module. The routine places zeroes in data registers 1 through 6, in the t register and in the display. The TI-95 also provides for clearing of the statistics registers with the CS1 or CS2 functions.

There is an important difference between the Pgm 01 SBR CLR sequence in the TI-59 and the CLR CS2 sequence in the TI-95--in the TI-95 the statistics registers are not a part of user memory. That difference becomes important in those instances where TI-59 programmers used statistics initialize sequence for some other reason. An example appeared when a member was trying to translate PPX program 628008 (Beams in Flexure). Program steps 352-355 contain the sequence Pgm 01 SBR CLR. The CLR CS2 sequence didn't provide an equivalent since what was needed was to place zeroes in data registers 1 through 6. The sequence 0 STO B STO C STO D STO E STO F STO G will provide the desired effect. A Dsz routine could be written to save two steps.

Some Pitfalls

Previous issues have discussed some functions where TI-59 and TI-95 responses are different when one would have expected the responses to be the same:

* With the TI-95 the random number generator R# delivers a different sequence of values each time the routine is used. With the TI-59 the random number generator in the Master Library module delivers the same sequence if the seed is the same. See V11N3P20.

* There are some subtle differences in the way the TI-59 and the TI-95 handle the EE function. See V11N4P4.

* INV $\Sigma+$ doesn't decrement the t register.

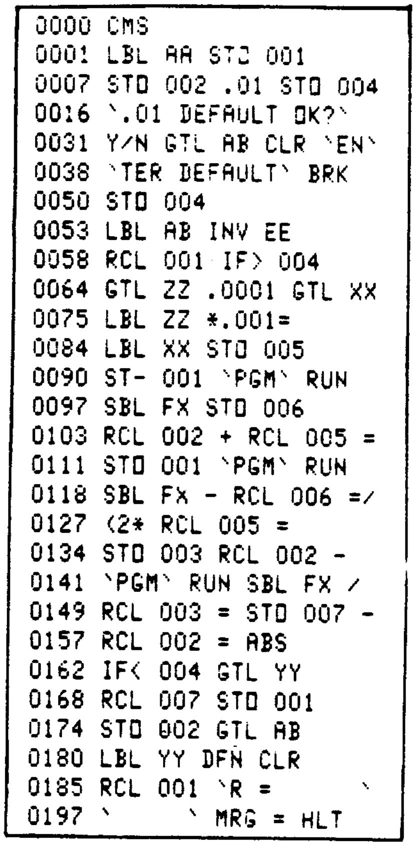
* The Signum function of the TI-95 (SGN) and of the TI-59 (OP 10) are not equivalent. See V12N3P24 for routines which will provide equivalent results.

* The TI-59 and TI-95 react in different ways to a negative arguments for y^x and INV y^x. See V11N4P18.

In the next issue we will discuss some TI-59 features which will not submit to an easy "brute force" translation such as use of the hierarchy registers.

ROOT-FINDER FOR THE TI-95 - D. Laughery

This program is a combination and improvement on two programs found in the Sourcebook for Programmable Calculators published by TI for use with the TI-58 and TI-59. The program uses the Newton-Raphson method to find a real root of a function f(x), and uses the difference between the value of f(x) in successive iterations as the basis for termination. The program may be stored in either user memory (MEM) or in a Constant Memory Cartridge. The program calls the function as a subroutine from system memory (PGM).



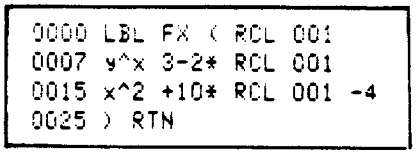
```
0000 CMS
0001 LBL AA STC 001
0007 STO 002 .01 STO 004
0016 `.01 DEFAULT OK?`
0031 Y/N GTL AB CLR `EN`
0038 `TER DEFAULT` BRK
0050 STO 004
0053 LBL AB INV EE
0058 RCL 001 IF> 004
0064 GTL ZZ .0001 GTL XX
0075 LBL ZZ *.001=
0084 LBL XX STO 005
0090 ST- 001 `PGM` RUN
0097 SBL FX STO 006
0103 RCL 002 + RCL 005 =
0111 STO 001 `PGM` RUN
0118 SBL Fx - RCL 006 =/
0127 (2* RCL 005 =
0134 STO 003 RCL 002 -
0141 `PGM` RUN SBL FX /
0149 RCL 003 = STO 007 -
0157 RCL 002 = ABS
0162 IF< 004 GTL YY
0168 RCL 007 STO 001
0174 STO 002 GTL AB
0180 LBL YY DFN CLR
0185 RCL 001 `R =          `
0197 `       ` MRG = HLT
```

The subroutine must begin with the label FX, be enclosed in parentheses, and end with RTN. It should not use = or CLR, and should not write to memory registers 001 through 007. The current value of x can be recalled from memory register 001. Sample subroutines for three functions are at the right. The functions mechanized are:

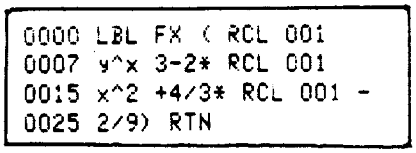
$$x^3 - 2x^2 + 10x - 4$$

$$x^3 - 2x^2 + (4/3)x - 2/9$$

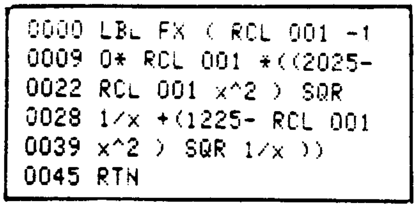
$$x^3 - 10x(1/\sqrt{(2025 - x^2)} + 1/\sqrt{(1225 - x^2)})$$



```
0000 LBL FX ( RCL 001
0007 y^x 3-2* RCL 001
0015 x^2 +10* RCL 001 -4
0025 ) RTN
```



```
0000 LBL FX ( RCL 001
0007 y^x 3-2* RCL 001
0015 x^2 +4/3* RCL 001 -
0025 2/9) RTN
```



```
0000 LBL FX ( RCL 001 -1
0009 0* RCL 001 *((2025-
0022 RCL 001 x^2 ) SQR
0028 1/x +(1225- RCL 001
0039 x^2 ) SQR 1/x ))
0045 RTN
```

The second function is Peter Messer's cubic test from V11N4P16. The third function is the ladder problem from V12N1P11. To use the program:

1. Enter the function in system memory (PGM).

2. Enter an estimate of the root into the display register.

3. Press RUN and select the program from user memory or a cartridge. The program stops with the message ".01 DEFAULT OK?" in the display. Press F1 to use the default error of 0.01. Press F2 to select the error and see "ENTER DEFAULT" in the display. Enter the desired error and press F1.

For the first function, an initial x of zero, and the default error the solution of 0.4288568507 will be displayed after about three seconds.

For the second function, an initial x of zero, and an error of 1EE-12 the real root is displayed as 0.2466929834 after seven seconds, and the 13d readout shows the value to be 0.2466929833685, which is within 2 in the least significant place of the exact solution. For an initial x of one the solution requires 34 seconds.

For the third function, an initial x of 30 and an error of 1EE-12 the solution of 31.8174591 is displayed in 15 seconds.

Translating ML-20 FOR THE TI-95 - Scott Garver and Palmer Hanson.

The ML-20 program from the Master Library module for the TI-59 calculates days between dates and day of the week. The user enters dates into the display using mmdd.yyyy format. Pressing User Defined Key A enters the first date. Pressing User Defined Key B enters a second date. Given prior entries with keys A and B, pressing User Defined Key C displays the days between the dates. With a date in the display, User Defined Key displays the day of the week, with 0 = Saturday, 1 = Sunday, ... 6 = Friday. The TI-59 listing is:

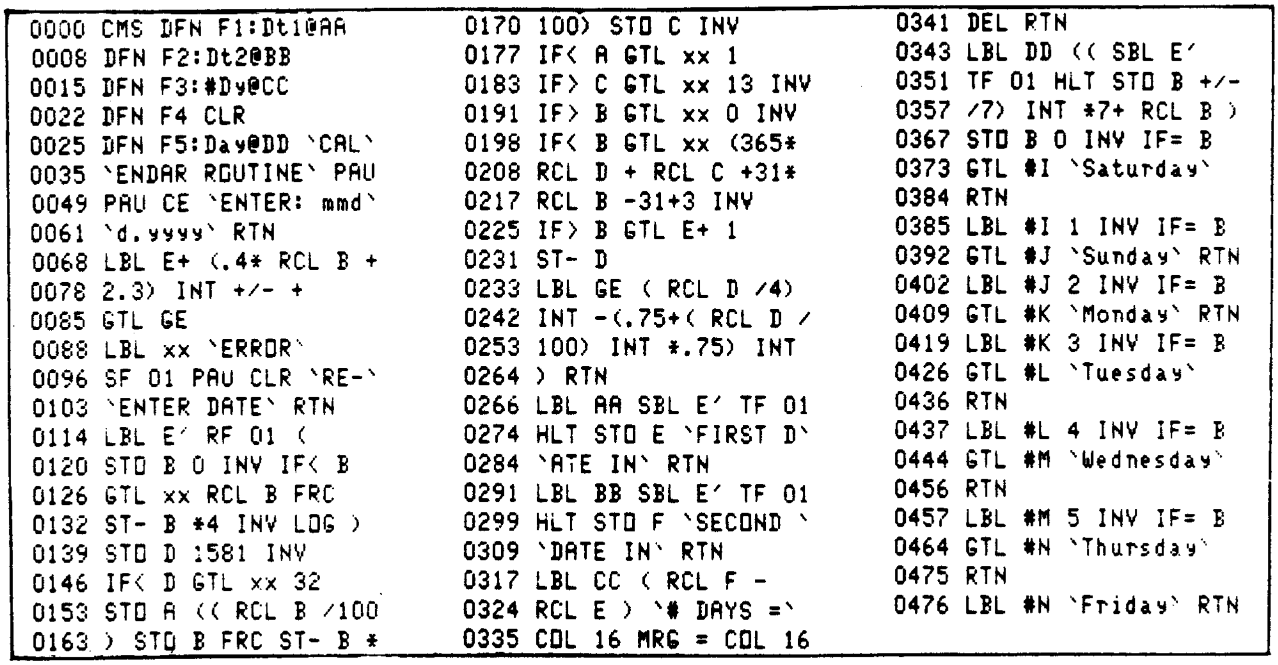
```
000  76 LBL    032  22 INV    064  01  01    096  85  +     128  07  7     160  00  0
001  78 Σ+     033  59 INT    065  22 INV    097  03  3     129  05  5     161  92 RTN
002  53 (      034  22 INV    066  59 INT    098  01  1     130  85  +     162  76 LBL
003  93  .     035  44 SUM    067  22 INV    099  65  ×     131  53  (     163  13  C
004  04  4     036  01  01    068  44 SUM    100  43 RCL    132  43 RCL    164  53  (
005  65  ×     037  65  ×     069  01  01    101  01  01    133  03  03    165  43 RCL
006  43 RCL    038  04  4     070  65  ×     102  75  -     134  55  ÷     166  05  05
007  01  01    039  22 INV    071  01  1     103  03  3     135  01  1     167  75  -
008  85  +     040  28 LOG    072  00  0     104  01  1     136  00  0     168  43 RCL
009  02  2     041  54  )     073  00  0     105  85  +     137  00  0     169  04  04
010  93  .     042  42 STD    074  54  )     106  03  3     138  54  )     170  54  )
011  03  3     043  03  03    075  42 STD    107  32 X:T    139  59 INT    171  92 RTN
012  54  )     044  32 X:T    076  02  02    108  43 RCL    140  65  ×     172  76 LBL
013  59 INT    045  01  1     077  77  GE    109  01  01    141  93  .     173  14  D
014  94 +/-    046  05  5     078  79  x̄     110  77  GE    142  07  7     174  53  (
015  85  +     047  08  8     079  01  1     111  78 Σ+     143  05  5     175  53  (
016  61 GTO    048  01  1     080  03  3     112  01  1     144  54  )     176  10 E'
017  77  GE    049  77  GE    081  32 X:T    113  22 INV    145  59 INT    177  42 STD
018  76 LBL    050  79  x̄     082  43 RCL    114  44 SUM    146  54  )     178  01  01
019  79  x̄     051  03  3     083  01  01    115  03  03    147  92 RTN    179  94 +/-
020  00  0     052  02  2     084  77  GE    116  76 LBL    148  76 LBL    180  55  ÷
021  35 1/X    053  32 X:T    085  79  x̄     117  77  GE    149  11  A     181  07  7
022  92 RTN    054  53  (     086  53  (     118  53  (     150  10 E'     182  54  )
023  76 LBL    055  53  (     087  03  3     119  43 RCL    151  42 STD    183  59 INT
024  10 E'     056  43 RCL    088  06  6     120  03  03    152  04  04    184  65  ×
025  53  (     057  01  01    089  05  5     121  55  ÷     153  00  0     185  07  7
026  42 STD    058  55  ÷     090  65  ×     122  04  4     154  92 RTN    186  85  +
027  01  01    059  01  1     091  43 RCL    123  54  )     155  76 LBL    187  43 RCL
028  29 CP     060  00  0     092  03  03    124  59 INT    156  12  B     188  01  01
029  22 INV    061  00  0     093  85  +     125  75  -     157  10 E'     189  54  )
030  77  GE    062  54  )     094  43 RCL    126  53  (     158  42 STD    190  92 RTN
031  79  x̄     063  42 STD    095  02  02    127  93  .     159  05  05    191  00  0
```

A "brute force" conversion can be obtained by (1) using DFN to define four function keys as A through D, and (2) replacing the t register comparisons of the TI-59 with equivalent register 000 comparisons on the TI-95. The listing follows:

```
0000 CLR DFN F1: A @AA           0093 581 INV IF< 000         0196 LBL GE ( RCL 003 /4
0008 DFN F2: B @BB               0100 GTL xx 32 EXC 000 (     0205 ) INT -(.75+(
0015 DFN F3: C @CC               0109 ( RCL 001 /100)         0214 RCL 003 /100) INT *
0022 DFN F4: D @DD HLT           0118 STD 001 FRC ST- 001     0224 .75) INT ) RTN
0030 LBL E+ (.4* RCL 001         0125 *100) STD 002 INV       0231 LBL AA SBL E'
0040 +2.3) INT +/- +             0134 IF< 000 SBL xx 13       0237 STD 004 0 RTN
0048 GTL GE                      0142 EXC 000 RCL 001 INV     0242 LBL BB SBL E'
0051 LBL xx 0 1/x RTN            0149 IF< 000 GTL xx (365     0248 STD 005 0 RTN
0057 LBL E' ( STD 001            0159 * RCL 003 + RCL 002     0253 LBL CC ( RCL 005 -
0064 EXC 000 0 EXC 000           0167 +31* RCL 001 -31+3      0261 RCL 004 ) RTN
0071 IF< 001 GTL xx FRC          0179 EXC 000 RCL 001 INV     0266 LBL DD (( SBL E'
0078 ST- 001 *4 INV LOG          0186 IF< 000 GTL E+ 1        0274 STD 001 +/- /7) INT
0085 ) STD 003 EXC 000 1         0193 ST- 003                 0282 *7+ RCL 001 ) RTN
```

A user will find that the instructions for the converted program are nearly the same as for the source TI-59 program. The major exception is that an erroneous entry causes a flashing display with the TI-59 but only sets the ERROR status indicator on the TI-95, and that indication is relatively easy to miss. The following translation of ML-20 by Scott Garver makes much better use of the TI-95 capability.

Translating ML-20 FOR THE TI-95 - (cont)



```
0000 CMS DFN F1:Dt1@AA
0008 DFN F2:Dt2@BB
0015 DFN F3:#Dy@CC
0022 DFN F4 CLR
0025 DFN F5:Day@DD `CAL`
0035 `ENDAR ROUTINE` PAU
0049 PAU CE `ENTER: mmd`
0061 `d.yyyy` RTN
0068 LBL E+ (.4* RCL B +
0078 2.3) INT +/- +
0085 GTL GE
0088 LBL xx `ERROR`
0096 SF 01 PAU CLR `RE-`
0103 `ENTER DATE` RTN
0114 LBL E′ RF 01 (
0120 STO B 0 INV IF< B
0126 GTL xx RCL B FRC
0132 ST- B *4 INV LOG )
0139 STO D 1581 INV
0146 IF< D GTL xx 32
0153 STO A (( RCL B /100
0163 ) STO B FRC ST- B *
0170 100) STO C INV
0177 IF< A GTL xx 1
0183 IF> C GTL xx 13 INV
0191 IF> B GTL xx 0 INV
0198 IF< B GTL xx (365*
0208 RCL D + RCL C +31*
0217 RCL B -31+3 INV
0225 IF> B GTL E+ 1
0231 ST- D
0233 LBL GE ( RCL D /4)
0242 INT -(.75+( RCL D /
0253 100) INT *.75) INT
0264 ) RTN
0266 LBL AA SBL E′ TF 01
0274 HLT STO E `FIRST D`
0284 `ATE IN` RTN
0291 LBL BB SBL E′ TF 01
0299 HLT STO F `SECOND `
0309 `DATE IN` RTN
0317 LBL CC ( RCL F -
0324 RCL E ) `# DAYS =`
0335 COL 16 MRG = COL 16
0341 DEL RTN
0343 LBL DD (( SBL E′
0351 TF 01 HLT STO B +/-
0357 /7) INT *7+ RCL B )
0367 STO B 0 INV IF= B
0373 GTL #I `Saturday`
0384 RTN
0385 LBL #I 1 INV IF= B
0392 GTL #J `Sunday` RTN
0402 LBL #J 2 INV IF= B
0409 GTL #K `Monday` RTN
0419 LBL #K 3 INV IF= B
0426 GTL #L `Tuesday`
0436 RTN
0437 LBL #L 4 INV IF= B
0444 GTL #M `Wednesday`
0456 RTN
0457 LBL #M 5 INV IF= B
0464 GTL #N `Thursday`
0475 RTN
0476 LBL #N `Friday` RTN
```

Important improvements include:

* The function key menu has more descriptive notation.

* Error indication is changed from setting the status indicator to the word "ERROR" flashed in the display followed by a prompt to "Re-ENTER DATE".

* The day of the week solution output is changed from a numeric code to an alphanumeric display of the day.

* Data register notation is changed from numbers to letters to save program steps.

* Input error identification has been extended to include the case where the entry for the day or month is zero. There are still some invalid entries which will be accepted. Examples include such entries as June 31, February 31, February 30, and February 29 in a non leap year.

There is a cost in memory useage. The "brute force" translation requires 296 bytes. Scott's translation uses 488 bytes.

REDUCED TI-74 AND TI-95 PRICES - V12N2P4 listed prices from various suppliers. A recent visit to the local Service Merchandise outlets shows that they have reduced prices for TI-74 and TI-95 hardware by some ten to eighteen percent. The current prices are:

TI-74	$ 99.97	ROM Modules	$29.80
TI-95	129.90		

MORE PPX PROGRAM AVAILABILITY - Earlier issues set up an informal program exchange to provide access to programs which were formerly available from the PPX Exchange. The following list shows 42 structures analysis programs which have been made available by member Shiu Lum Lee:

R	628008	- Beams in Flexure
R	628016	- Bearing Plate Design
R	628017	- Beams in Flexure with PC-100A Labels
R	628020	- Moment Distribution: Variable Number of Spans
R	628021	- Base Plate Design
R	628025	- Wall Footing Design
R	628063	- Rectangular Concrete Rooting Design
MR	628066	- Elastic Properties of Non-prismatic Beams
MR	628067	- Steel Beam Column Analysis
R	628068	- AISC Allowable Column Loads
R	628073	- Simple Beam Moment Diagram
R	628076	- Bending Stress Program
R	628078	- AISC Allowable Bending Stresses
R	628079	- Pipe Network Calculator
R	628082	- Double Box Frame Moment Distribution
R	628087	- Analysis of Two Hinged Rigid Rectangular Frames
R	628090	- Mechanical Testing of Spans (Composite Materials)
MR	628091	- Truss Design
R	628095	- Interaction Equation
R	628100	- Cantilever Retaining Wall with Backfill
R	628101	- Footing Soil Bearing
MR	628102	- Masonry Column Design
R	628114	- Foundation Design for Pile Supported Abutment
MR	628123	- Frame Analysis - Moment Distribution
R	628124	- Properties of Builtup Sections
R	628129	- Flat Plate Design II
MR	628133	- Rectangular Biaxial Column
R	628137	- Flat Plate Design I
R	628147	- Effective Moment of Inertia
R	628148	- Reactions for program 628020
R	628151	- Loads or Columns at Each Floor
R	628156	- Concrete Mix Design (Absolute Volume)
R	628162	- Finite Beam on an Elastic Foundation
R	628165	- Steel Column Baseplates
R	628169	- Geometric Section Properties
MR	628171	- Retaining Wall Design
R	628174	- Wide Flange Shapes
R	628177	- Simple Beam Analysis
R	628187	- AISC Composite Beams
R	628192	- Composite Beam Design
MR	628195	- Column Web Stiffeners

Code M means the programs are available from Thomas E. Ceteraki, 10010 Alderson St., Schofield WI 54476. Send a stamped and self-addressed envelope for details.

Code R means the programs are available from Shiu Lum Lee, 1029 E. 102nd Street, Brooklyn, NY 11236. Send a stamped and self-addressed envelope for information on specific programs.

<u>36 DIGIT DMS TO RADIANS</u> - L. Leeds. V9N3P11 presented Peter Messer's program for calculating the sine or cosine to 36 digits. The program required that the input angle be in radians. The program below converts a degree-minute-second (DMS) input to radians for use by Peter's program:

```
000  91 R/S    035  55  ÷     070  42 STO    105  69 OP     140  42 STO    175  01  01
001  25 CLR    036  03  3     071  53  53    106  22  22    141  05  05    176  01  1
002  61 GTO    037  06  6     072  42 STO    107  69 OP     142  73 RC*    177  00  0
003  00  00    038  00  0     073  54  54    108  23  23    143  01  01    178  42 STO
004  58  58    039  00  0     074  42 STO    109  97 DSZ    144  55  ÷     179  03  03
005  76 LBL    040  95  =     075  55  55    110  05  05    145  43 RCL    180  73 RC*
006  11  A     041  65  ×     076  42 STO    111  00  00    146  06  06    181  03  03
007  06  6     042  43 RCL    077  56  56    112  97  97    147  95  =     182  91 R/S
008  69 OP     043  06  06    078  42 STO    113  07  7     148  75  -     183  69 OP
009  17  17    044  95  =     079  57  57    114  42 STO    149  59 INT    184  23  23
010  47 CMS    045  44 SUM    080  05  5     115  05  05    150  74 SM*    185  97 DSZ
011  08  8     046  58  58    081  01  1     116  06  6     151  02  02    186  01  01
012  69 OP     047  91 R/S    082  42 STO    117  01  1     152  95  =     187  01  01
013  17  17    048  76 LBL    083  01  01    118  42 STO    153  65  ×     188  80  80
014  01  1     049  13  C     084  06  6     119  02  02    154  43 RCL    189  03  3
015  52 EE     050  61 GTO    085  01  1     120  06  6     155  06  06    190  35 1/X
016  06  6     051  03  03    086  42 STO    121  22 INV    156  95  =     191  91 R/S
017  42 STO    052  15  15    087  02  02    122  44 SUM    157  72 ST*
018  06  06    053  76 LBL    088  01  1     123  03  03    158  01  01
019  25 CLR    054  14  D     089  01  1     124  69 OP     159  69 OP
020  91 R/S    055  61 GTO    090  42 STO    125  21  21    160  31  31
021  42 STO    056  01  01    091  03  03    126  97 DSZ    161  69 OP
022  51  51    057  72  72    092  07  7     127  04  04    162  32  32
023  91 R/S    058  43 RCL    093  42 STO    128  00  00    163  97 DSZ
024  55  ÷     059  58  58    094  04  04    129  97  97    164  05  05
025  06  6     060  75  -     095  42 STO    130  02  2     165  01  01
026  00  0     061  59 INT    096  05  05    131  03  3     166  42  42
027  95  =     062  42 STO    097  73 RC*    132  42 STO    167  25 CLR
028  65  ×     063  52  52    098  01  01    133  01  01    168  69 OP
029  43 RCL    064  95  =     099  65  ×     134  02  2     169  99  99
030  06  06    065  65  ×     100  73 RC*    135  02  2     170  66 PAU    315  04  4
031  95  =     066  43 RCL    101  02  02    136  42 STO    171  81 RST    316  05  5
032  42 STO    067  06  06    102  95  =     137  02  02    172  25 CLR    317  30 TAN
033  58  52    068  95  =     103  74 SM*    138  01  1     173  07  7     318  33 X²
034  91 R/S    069  59 INT    104  03  03    139  03  3     174  42 STO    319  86 STF
```

The program also requires seven constants in data registers 61 through 67 as listed at the right. Store the program and the constants in banks 1 and 2 using the startup partitioning.

17453.	61
292519.	62
943295.	63
769236.	64
907684.	65
886127.	66
134428.	67

User Instructions:

1. Press A to initialize.

2. Enter the degrees and press R/S.

3. Enter the minutes and press R/S.

4. Enter the seconds and press R/S.

5. To run in fast mode press C and see a flashing "1.". Press 7 and then EE. After about 35 seconds a flashing zero indicates the conversion is complete.

6. Press D and see the integer part of the result. Press R/S six times to see the fractional part. Add leading zeroes to any return of less than six digits. An extra R/S will yield all threes in the display. To print the result change the R/S (code 91) at step 182 to PRT (code 99).

The result is located in data registers 10 through 16. These are not the positions required by Peter's program; furthermore, the integer part and the first six digits of the fractional part must be combined as the first entry for Peter's program. Will someone write a combination program which accepts the DMS format and delivers the sine or cosine without intermediate readout and reentry?

__FIVE FUNCTION CURVE FIT__ - This program was written in response to requests for a TI-74 curve fit program which would be versatile, but easier to use than some of the recently published programs. I decided to model the program after the Forecasting - Auto Curve Choice program in the Real Estate/Investment Solid State module for the TI-59. That program tests the capability of four functions--linear, power, exponential and logarithmic--to fit the input data, selects the function which yields the largest coefficient of determination (r^2), and provides the ability to calculate values for y as a function of x using the selected best fit function. This program provides the same capability, adds the hyperbolic function, selects the best fit based on the largest magnitude of the correlation coefficient (r), and saves the input data pairs so that the residual errors can be examined. That capability is particulary helpful in identifying wild data points. The program also provides for operator intervention to select one of the functions for fitting, a capability which was also available in the Real Estate/Investment module.

The program includes a fairly complete set of prompts. Users are cautioned that the use of zero or negative input values will cause the "Find Best Fit" option to abort. If non-positive input values are required for the fit then individual functions may be used according to the following table:

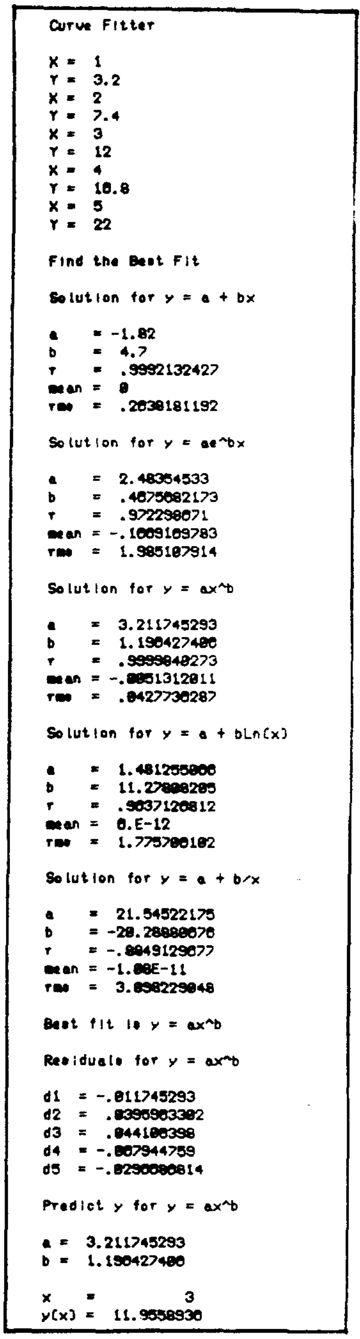
Linear	No limits on input values.
Exponential	Y input must be positive.
Power	X and Y input must be positive.
Logarithmic	X input must be positive.
Hyperbolic	X input may not be zero.

In some cases the user may find that a rule against zero values can be circumvented by replacing the zero by a very small positive number.

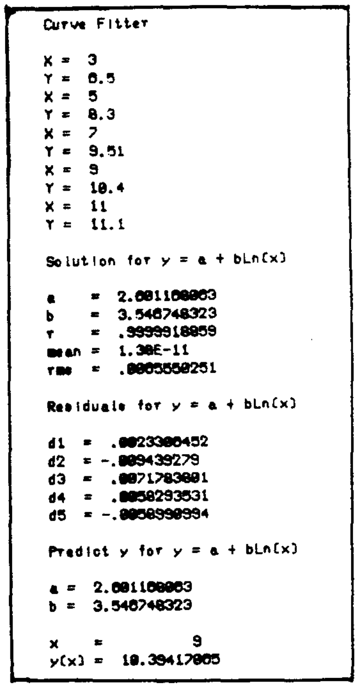
The sample printout at the right was made with the HX-1000. You may recognize the test problems as being from Maurice Swinnen's <u>Statistics Library</u> book. When run without a printer the program pauses to permit the user to view the output.

The listing on page 25 was printed with the PC-324.

In the next issue I will try to present a TI-95 equivalent.



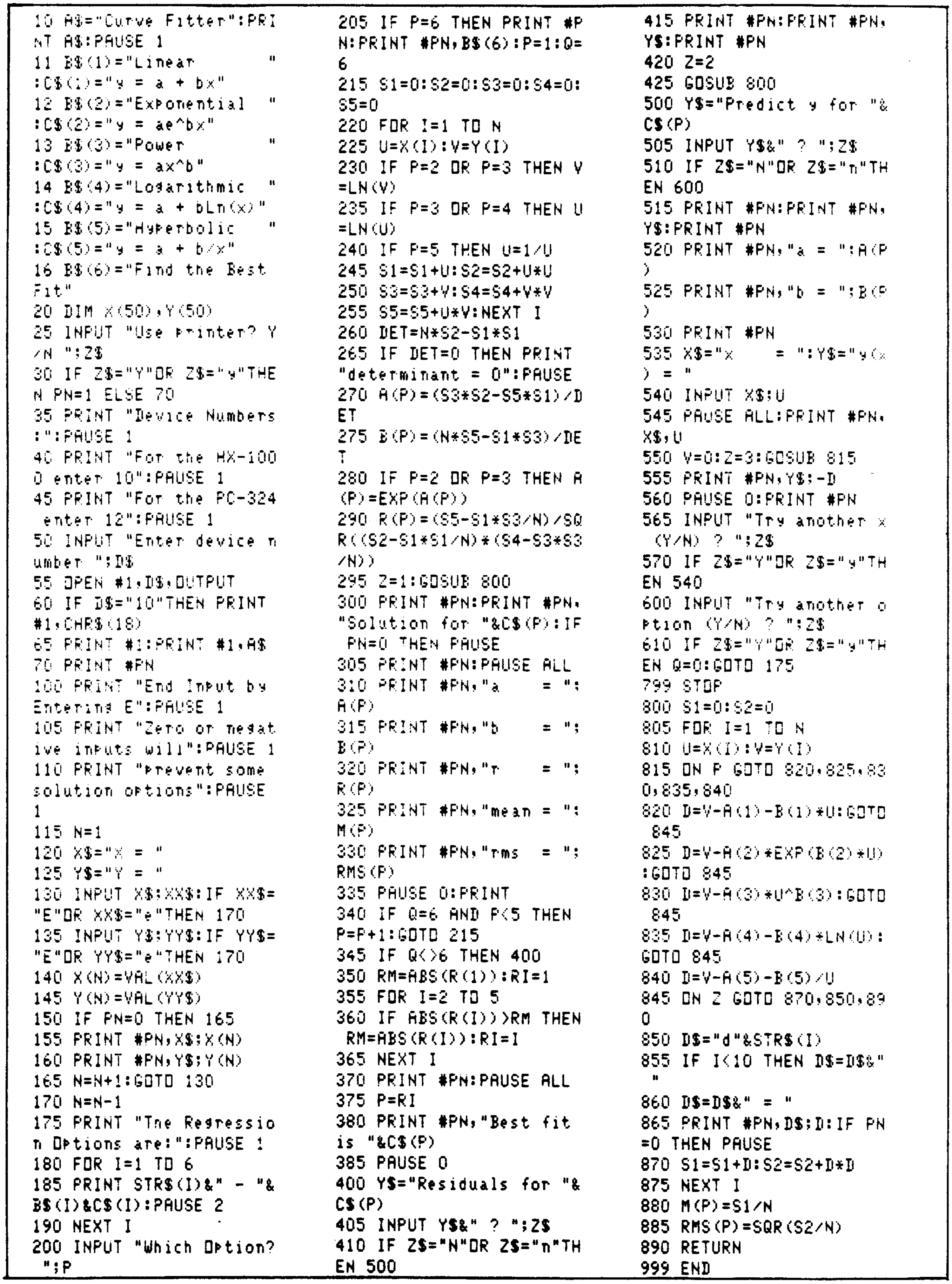
```
Curve Fitter

X =  1
Y =  3.2
X =  2
Y =  7.4
X =  3
Y =  12
X =  4
Y =  16.8
X =  5
Y =  22

Find the Best Fit

Solution for y = a + bx

a    = -1.82
b    =  4.7
r    =  .9992132427
mean =  0
rms  =  .2638181192

Solution for y = ae^bx

a    =  2.48364533
b    =  .4675082173
r    =  .972298071
mean = -.1009109783
rms  =  1.985107914

Solution for y = ax^b

a    =  3.211745293
b    =  1.196427406
r    =  .9999840273
mean = -.0051312011
rms  =  .0427736287

Solution for y = a + bLn(x)

a    =  1.481255006
b    =  11.27808285
r    =  .9037120812
mean =  0.E-12
rms  =  1.775700102

Solution for y = a + b/x

a    =  21.54522175
b    = -20.28880676
r    = -.8849129677
mean = -1.08E-11
rms  =  3.898229048

Best fit is y = ax^b

Residuals for y = ax^b

d1  = -.011745293
d2  =  .0395963302
d3  =  .044106398
d4  = -.067944759
d5  = -.0296080814

Predict y for y = ax^b

a =  3.211745293
b =  1.196427406

x    =           3
y(x) =  11.9558936
```



```
Curve Fitter

X =  3
Y =  6.5
X =  5
Y =  8.3
X =  7
Y =  9.51
X =  9
Y =  10.4
X =  11
Y =  11.1

Solution for y = a + bLn(x)

a    =  2.601108063
b    =  3.546748323
r    =  .9999918859
mean =  1.38E-11
rms  =  .0065550251

Residuals for y = a + bLn(x)

d1  =  .0023306452
d2  = -.009439279
d3  =  .0071783001
d4  =  .0058293531
d5  = -.0058990994

Predict y for y = a + bLn(x)

a =  2.601108063
b =  3.546748323

x    =           9
y(x) =  10.39417065
```

Five Function Curve Fit - (cont)

Program Listing:



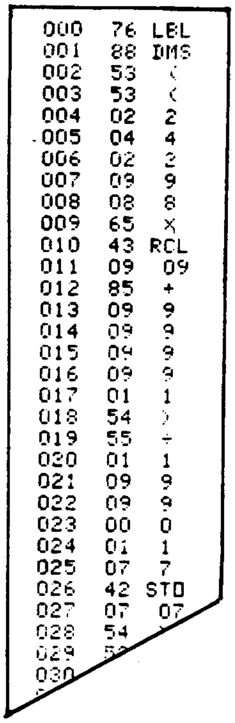
```
10 A$="Curve Fitter":PRI
NT A$:PAUSE 1
11 B$(1)="Linear       "
:C$(1)="y = a + bx"
12 B$(2)="Exponential  "
:C$(2)="y = ae^bx"
13 B$(3)="Power        "
:C$(3)="y = ax^b"
14 B$(4)="Logarithmic  "
:C$(4)="y = a + bLn(x)"
15 B$(5)="Hyperbolic   "
:C$(5)="y = a + b/x"
16 B$(6)="Find the Best
Fit"
20 DIM X(50),Y(50)
25 INPUT "Use printer? Y
/N ";Z$
30 IF Z$="Y"OR Z$="y"THE
N PN=1 ELSE 70
35 PRINT "Device Numbers
:":PAUSE 1
40 PRINT "For the HX-100
0 enter 10":PAUSE 1
45 PRINT "For the PC-324
 enter 12":PAUSE 1
50 INPUT "Enter device n
umber ";D$
55 OPEN #1,D$,OUTPUT
60 IF D$="10"THEN PRINT
#1,CHR$(18)
65 PRINT #1:PRINT #1,A$
70 PRINT #PN
100 PRINT "End Input by
Entering E":PAUSE 1
105 PRINT "Zero or negat
ive inputs will":PAUSE 1
110 PRINT "prevent some
solution options":PAUSE
1
115 N=1
120 X$="X = "
125 Y$="Y = "
130 INPUT X$:XX$:IF XX$=
"E"OR XX$="e"THEN 170
135 INPUT Y$:YY$:IF YY$=
"E"OR YY$="e"THEN 170
140 X(N)=VAL(XX$)
145 Y(N)=VAL(YY$)
150 IF PN=0 THEN 165
155 PRINT #PN,X$;X(N)
160 PRINT #PN,Y$;Y(N)
165 N=N+1:GOTO 130
170 N=N-1
175 PRINT "Tne Regressio
n Options are:":PAUSE 1
180 FOR I=1 TO 6
185 PRINT STR$(I)&" - "&
B$(I)&C$(I):PAUSE 2
190 NEXT I
200 INPUT "Which Option?
 ";P
```

```
205 IF P=6 THEN PRINT #P
N:PRINT #PN,B$(6):P=1:Q=
6
215 S1=0:S2=0:S3=0:S4=0:
S5=0
220 FOR I=1 TO N
225 U=X(I):V=Y(I)
230 IF P=2 OR P=3 THEN V
=LN(V)
235 IF P=3 OR P=4 THEN U
=LN(U)
240 IF P=5 THEN U=1/U
245 S1=S1+U:S2=S2+U*U
250 S3=S3+V:S4=S4+V*V
255 S5=S5+U*V:NEXT I
260 DET=N*S2-S1*S1
265 IF DET=0 THEN PRINT
"determinant = 0":PAUSE
270 A(P)=(S3*S2-S5*S1)/D
ET
275 B(P)=(N*S5-S1*S3)/DE
T
280 IF P=2 OR P=3 THEN A
(P)=EXP(A(P))
290 R(P)=(S5-S1*S3/N)/SQ
R((S2-S1*S1/N)*(S4-S3*S3
/N))
295 Z=1:GOSUB 800
300 PRINT #PN:PRINT #PN,
"Solution for "&C$(P):IF
 PN=0 THEN PAUSE
305 PRINT #PN:PAUSE ALL
310 PRINT #PN,"a    = ";
A(P)
315 PRINT #PN,"b    = ";
B(P)
320 PRINT #PN,"r    = ";
R(P)
325 PRINT #PN,"mean = ";
M(P)
330 PRINT #PN,"rms  = ";
RMS(P)
335 PAUSE 0:PRINT
340 IF Q=6 AND P<5 THEN
P=P+1:GOTO 215
345 IF Q<>6 THEN 400
350 RM=ABS(R(1)):RI=1
355 FOR I=2 TO 5
360 IF ABS(R(I))>RM THEN
 RM=ABS(R(I)):RI=I
365 NEXT I
370 PRINT #PN:PAUSE ALL
375 P=RI
380 PRINT #PN,"Best fit
is "&C$(P)
385 PAUSE 0
400 Y$="Residuals for "&
C$(P)
405 INPUT Y$&" ? ";Z$
410 IF Z$="N"OR Z$="n"TH
EN 500
```

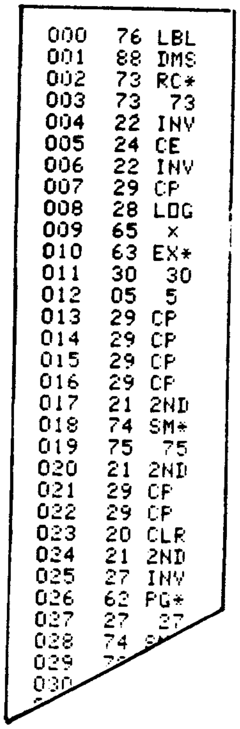
```
415 PRINT #PN:PRINT #PN,
Y$:PRINT #PN
420 Z=2
425 GOSUB 800
500 Y$="Predict y for "&
C$(P)
505 INPUT Y$&" ? ";Z$
510 IF Z$="N"OR Z$="n"TH
EN 600
515 PRINT #PN:PRINT #PN,
Y$:PRINT #PN
520 PRINT #PN,"a = ";A(P
)
525 PRINT #PN,"b = ";B(P
)
530 PRINT #PN
535 X$="x     = ":Y$="y(x
) = "
540 INPUT X$:U
545 PAUSE ALL:PRINT #PN,
X$;U
550 V=0:Z=3:GOSUB 815
555 PRINT #PN,Y$;-D
560 PAUSE 0:PRINT #PN
565 INPUT "Try another x
 (Y/N) ? ";Z$
570 IF Z$="Y"OR Z$="y"TH
EN 540
600 INPUT "Try another o
ption (Y/N) ? ";Z$
610 IF Z$="Y"OR Z$="y"TH
EN Q=0:GOTO 175
799 STOP
800 S1=0:S2=0
805 FOR I=1 TO N
810 U=X(I):V=Y(I)
815 ON P GOTO 820,825,83
0,835,840
820 D=V-A(1)-B(1)*U:GOTO
 845
825 D=V-A(2)*EXP(B(2)*U)
:GOTO 845
830 D=V-A(3)*U^B(3):GOTO
 845
835 D=V-A(4)-B(4)*LN(U):
GOTO 845
840 D=V-A(5)-B(5)/U
845 ON Z GOTO 870,850,89
0
850 D$="d"&STR$(I)
855 IF I<10 THEN D$=D$&"
 "
860 D$=D$&" = "
865 PRINT #PN,D$;D:IF PN
=0 THEN PAUSE
870 S1=S1+D:S2=S2+D*D
875 NEXT I
880 M(P)=S1/N
885 RMS(P)=SQR(S2/N)
890 RETURN
999 END
```

AN UNUSUAL TI-59 DIAGNOSTIC MALFUNCTION

I continue to be intrigued by the ways in which computers and calculators can fail. The latest example involves the diagnostic routine in the Master Library Solid State Software module for the TI-59 (ML-01). In this particular TI-59 the Pgm 01 SBR = routine will operate properly immediately after turnon, but after a short warmup the routine will stop with a flashing "1." in the disp;ay. I downloaded the routine to user memory in order to pinpoint the problem, and isolated the problem to the call to Pgm 15 SBR DMS at steps 050-053 of the ML-01 diagnostic. I downloaded ML-15 and found the curious listing which is in the right hand column. The correct listing is in the left hand column. Starting at step 002 the incorrect listing has key codes which are 20 higher than they should be. Curious!



```
000  76 LBL
001  88 DMS
002  53  (
003  53  (
004  02  2
005  04  4
006  02  2
007  09  9
008  08  8
009  65  X
010  43 RCL
011  09  09
012  85  +
013  09  9
014  09  9
015  09  9
016  09  9
017  01  1
018  54  )
019  55  ÷
020  01  1
021  09  9
022  09  9
023  00  0
024  01  1
025  07  7
026  42 STO
027  07  07
028  54
029
030
```



```
000  76 LBL
001  88 DMS
002  73 RC*
003  73  73
004  22 INV
005  24 CE
006  22 INV
007  29 CP
008  28 LOG
009  65  X
010  63 EX*
011  30  30
012  05  5
013  29 CP
014  29 CP
015  29 CP
016  29 CP
017  21 2ND
018  74 SM*
019  75  75
020  21 2ND
021  29 CP
022  29 CP
023  20 CLR
024  21 2ND
025  27 INV
026  62 PG*
027  27  27
028  74
029
030
```

A PROGRAMMING CHALLENGE - Don Laughery.

This problem comes from the application area of positional tolerancing. Details of the complete problem appear in the article "True Position" in the September 1987 issue of Quality magazine. However, in the barest terms, the problem reduces to the need for an algorithm and program that will find the smallest diameter circle which will just enclose n random points. The article describes graphical methods for accomplishing the solution, but a program which would solve analytically would be helpful.

Editor's Note: Clearly, n is greater than three since programs already exist which will find the circle through three points, e.g., PPX 398008 for the TI-59 and V7N1/2P27 for the TI-57. The algebraic solution to the three point problem is used as an illustration of how the equation of a circle is defined by three conditions in both of my analytic geometry textbooks. A cursory check of my textbooks failed to find any discussion of finding the smallest circle which will encompass four or more points.

TABLE OF CONTENTS

ERRATA 2
SeTIc 3
TI-95 SELF-TEST, R. Prins 4
TI-95 CALENDAR CHALLENGE, H. Ladd 6
SOLUTION FOR A QUARTIC, P. Messer, H. Ladd, W. Hawes 8
TI-59 DIAGNOSTICS, S. Garver 11
AOS CALCULATION FROM HP, W. Widmer 11
ANOTHER SIMULTANEOUS EQUATION SOLUTION 12
TRANSLATING FROM THE TI-59 TO THE TI-95 16
ROOT-FINDER FOR THE TI-95, D. Laughery 19
TRANSLATING ML-20 FOR THE TI-95, S. Garver 20
MORE PPX PROGRAM AVAILABILITY, S. L. Lee 22
36 DIGIT DMS TO RADIANS, L. Leeds 23
FIVE FUNCTION CURVE FIT FOR THE TI-74 24
A PROGRAMMING CHALLENGE, D. Laughery 26
AN UNUSUAL TI-59 DIAGNOSTIC MALFUNCTION 26
HARDWARE AVAILABILITY - 3, 5, 11, 21

Magnetic card service continues to be available for TI-59 programs which appear in TI PPC Notes. One dollar per card and a SASE, please.

That's all for this "year".