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 trans-
lation, 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 Ernis 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 sub-
mitted 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 newsletter is not copyrighted and may be reproduced for personal
use. When material is used elsewhere we ask as a matter of courtesy
that TI PPC Notes be mentioned. 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.
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 \( Q = \text{ATN}(1.0 \times 14) \) 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.

<table>
<thead>
<tr>
<th>10 DIM R(100);</th>
<th>10 DIM R(100);</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 RANDOMIZE</td>
<td>20 RANDOMIZE</td>
</tr>
<tr>
<td>30 CALL UP(&quot;Scramble Program&quot;</td>
<td>30 CALL UP(&quot;Scramble Program&quot;</td>
</tr>
<tr>
<td>FROM PH)</td>
<td>FROM PH)</td>
</tr>
<tr>
<td>40 PRINT &quot;Seed?&quot;; S = 234</td>
<td>40 PRINT &quot;Seed:&quot;; S = 234</td>
</tr>
<tr>
<td>45 PRINT #PH; *Seed = &quot;IS</td>
<td>45 PRINT #PH; &quot;Seed = &quot;IS</td>
</tr>
<tr>
<td>50 PRINT #H; 50 PRINT #H;</td>
<td></td>
</tr>
<tr>
<td>50 FOR I = 1 TO 100; I = H</td>
<td>50 FOR I = 1 TO 100; I = H</td>
</tr>
<tr>
<td>50 NEXT I</td>
<td>50 NEXT I</td>
</tr>
<tr>
<td>60 FOR H = 100 TO 1 STEP + 1</td>
<td>70 FOR H = 100 TO 1 STEP + 1</td>
</tr>
<tr>
<td>70 PRINT #H; 50 IF H = 100 THEN 70</td>
<td>80 IF H = 100 THEN 70</td>
</tr>
<tr>
<td>80 H = H + 1; 80 H = H + 1;</td>
<td></td>
</tr>
<tr>
<td>80 NEXT H</td>
<td>90 FOR J = 1 TO 95 STEP 5</td>
</tr>
<tr>
<td>90 H = H + 1; 90 H = H + 1;</td>
<td></td>
</tr>
<tr>
<td>90 NEXT J</td>
<td>110 FOR J = 1 TO 5</td>
</tr>
<tr>
<td>100 PRINT #PH; &quot;R(J) + 1; 120 PRINT #PH; USING 200;</td>
<td></td>
</tr>
<tr>
<td>100 NEXT J</td>
<td>120 R(J) + 1;</td>
</tr>
<tr>
<td>100 IF PW = 0 THEN PAUSE</td>
<td>130 IF PW = 0 THEN PAUSE</td>
</tr>
<tr>
<td>100 PRINT #PH;</td>
<td>150 PRINT #PH;</td>
</tr>
<tr>
<td>180 STOP</td>
<td>160 NEXT I</td>
</tr>
<tr>
<td>180 STOP</td>
<td>170 PRINT #PH;</td>
</tr>
<tr>
<td>190 END</td>
<td>180 STOP</td>
</tr>
<tr>
<td></td>
<td>200 IMAGE ???</td>
</tr>
<tr>
<td></td>
<td>999 END</td>
</tr>
</tbody>
</table>

**Scramble Program**

| Seed = 234 |
| 15 76 50 26 34 |
| 17 1 16 78 72 |
| 36 82 99 22 91 |
| 71 93 86 84 24 |
| 73 42 7 6 23 |
| 11 94 63 14 12 |
| 92 62 63 28 2 |
| 84 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 |
| 92 52 87 41 81 |
| 4 92 57 70 64 |
| 51 9 65 60 89 |
| 68 98 29 74 20 |
| 79 53 46 66 100 |
| 44 21 77 40 19 |
| 65 75 16 96 3 |
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" for 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 will 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.

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 buyer's 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:

<table>
<thead>
<tr>
<th>Command</th>
<th>Press</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Switch the machine on</td>
<td>ON</td>
<td>TI-95 PROCALC</td>
</tr>
<tr>
<td>2. Select SYSTEM MODE</td>
<td>FUNC SYS YES</td>
<td>SYSTEM FUNCTIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STB RCB SBA</td>
</tr>
<tr>
<td>3. Select UNFORMATTED MODE</td>
<td>CONV BAS UNF</td>
<td>00000000000000000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEC HEX OCT 2AC UNF</td>
</tr>
<tr>
<td>4. Return to the CONV menu</td>
<td>CONV</td>
<td>00000000000000000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MET DMS ANG P-R BAS</td>
</tr>
<tr>
<td>5. Enter 0000090003143002</td>
<td>0000090003143002</td>
<td>0000090003143002</td>
</tr>
<tr>
<td>6. Store this number in R048</td>
<td>STO 2048</td>
<td>0000090003143002</td>
</tr>
<tr>
<td>7. Select the SELFTESTS menu</td>
<td>F5 (BAS)</td>
<td>SELFTESTS:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>END TST EXT CYC --&gt;</td>
</tr>
<tr>
<td>8. To see the other options</td>
<td>F5 (---&gt;)</td>
<td>SELFTESTS:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MOD PTR KBD LCD --&gt;</td>
</tr>
</tbody>
</table>

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 window.
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. Goza, 18004 Fonthill Ave.,
Torrance CA 90504
TI-95 CALENDAR CHALLENGE - Hewlett Ladd. The printing of a calendar has been one of our longstanding benchmarks for a calculator printer capability. The first 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 code 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 | 0202 MRR D CDL 11 | 0472 1 STD K SBR 0257 7* |
| 0005 NDF NDF NDF NDF NDF | 0207 CDL 08 MRR C CDL 08 | 0480 (< RCL N )|-170= ABS |
| 0010 'JAN' GTO 0223 NDF | 0213 " CDL 05 MRR B | 0491 STD A CE SBR IND A |
| 0018 NDF NDF SF 02 'FE' | 0218 CDL 05 MRT RTN | 0497 4 STD J INV DS2 J |
| 0025 'B' GTO 0223 NDF | 0223 CDL 24 MRR L CDL 24 | 0503 GTO 0521 7 STD 1 1 |
| 0030 'MAR' GTO 0223 NDF | 0229 RPT RTN CE | 0510 STD R A SBR 0257 |
| 0038 NDF NDF SF 01 'AP' | 0235 'SU MO TU WE TH FR' | 0515 SBR 0128 GTO 0500 |
| 0045 'N' GTO 0223 NDF | 0252 'SA' RPT RTN RCL K | 0521 TF 02 GTO 0627 |
| 0050 'MAY' GTO 0223 NDF | 0259 STD IND A INC A | 0526 TF 01 GTO 0570 3 |
| 0058 NDF NDF SF 01 'JU' | 0264 INC K DS2 I | 0532 ST+ P 7 IF< P |
| 0065 'JUN' GTO 0223 NDF | 0268 GTO 0257 RTN CE 'Y' | 0537 GTO 0580 7 STD 1 1 |
| 0070 'JUL' GTO 0223 NDF | 0274 "YYYY.MM'' BRK STD Q | 0544 STD R A SBR 0257 7*(-\\|
| 0078 NDF NDF 'AUG' | 0284 INT STD L = ABS * | 0553 RCL P )+179= STD A |
| 0084 GTO 0223 NDF | 0291 100= STD M = INV | 0563 CE SBR IND A |
| 0089 NDF SF 01 'SEP' | 0299 IF> M GTO 0311 INV | 0567 GTO 0622 2 ST+ P 7 |
| 0096 GTO 0223 NDF 'OCT' | 0305 INC L 12 ST+ M 1582 | 0574 INV IF< P GTO 0540 |
| 0104 GTO 0223 NDF | 0315 .10 IF> Q GTO 0345 | 0580 7 STD 1 1 STD A |
| 0109 NDF SF 01 'NOV' | 0323 RCL L /100= INT | 0586 SBR 0257 SBR 0128 8 |
| 0116 GTO 0223 NDF 'DEC' | 0331 STD R /4= INT | 0593 IF= P GTO 0622 2 |
| 0124 GTO 0223 NDF CDL 05 | 0338 RCL R /2= STD S 0 | 0599 STD 1 1 STD A |
| 0130 MRR B CDL 05 "" | 0346 INV IF> L GTO 0371 | 0604 SBR 0257 9 IF= P |
| 0135 CDL 08 MRR C CDL 08 | 0352 "365.25= RCL L -.75 | 0610 GTO 0619 SBR 0207 |
| 0141 'CDL 11 MRR D | 0366 INT GTO 0382 365. | 0616 GTO 0622 SBR 0214 |
| 0146 CDL 11 'CDL 14 | 0375 25= RCL L } INT +(< | 0622 ADV CFG GTO 0000 15 |
| 0151 MRR E CDL 14 "" | 0385 RCL M +1)#.40.6001) | 0629 83 IF) L GTO 0622 |
| 0156 CDL 17 MRR F CDL 17 | 0399 INT /1720994.5+ | 0636 RCL L /400= FRC |
| 0162 '' CDL 20 MRR G | 0411 RCL S +3.5-<7) INT | 0644 IF= 0 GTO 0682 |
| 0167 CDL 20 \" CDL 23 | 0423 *2= INV IF= 0 | 0649 RCL L /100= FRC |
| 0172 MRR H CDL 23 | 0429 GTO 0433 7 STD N | 0657 IF= 0 GTO 0673 |
| 0177 PRT RTN CDL 20 | 0435 STD P 12 INV IF< M | 0662 RCL L /4= FRC IF= 0 |
| 0181 MRR G CDL 20 | 0442 GTO 0449 ST = M | 0670 GTO 0682 1 IF= P |
| 0186 CDL 17 MRR F CDL 17 | 0447 INC L 10 ST+ M | 0676 GTO 0622 GTO 0540 1 |
| 0192 "" CDL 14 MRR E | 0453 SBR IND M NDF NDF | 0683 ST+ P 8 IF= P |
| 0197 CDL 14 "" CDL 11 | 0456 NDF SBR 0232 RCL K | 0688 GTO 0580 GTO 0540 |
TI-95 Calendar Challenge (cont)

User Instructions:

Run the program and see the prompt YYYMM 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.

<table>
<thead>
<tr>
<th>FEB 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SU MD TU WE TH FR SA</td>
</tr>
<tr>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>6 7 8 9 10 11 12</td>
</tr>
<tr>
<td>13 14 15 16 17 18 19</td>
</tr>
<tr>
<td>20 21 22 23 24 25 26</td>
</tr>
<tr>
<td>27 28 29</td>
</tr>
</tbody>
</table>

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 Bodensauler 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 slightest of key 'brushing!' But these are the only errors I've encountered in using any (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 \( p = \text{SQR}(b^2 - 4c + 4R), q = \text{SQR}(R^2 - 4e) \) and \( S = (bR - 2d) \).

If \( S \leq 0 \) then use the QAD function to solve

\[ 2z^4 + (b + p)z + (R - q) = 0 \] and \[ 2z^4 + (b - p)z + (R + q) = 0 \]

If \( S > 0 \) then use the QAD function to solve

\[ 2z^4 + (b + p)z + (R + q) = 0 \] and \[ 2z^4 + (b - p)z + (R - q) = 0 \]

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 equations 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 appears 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 Meuser/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 Meuser/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 Meuser/Ladd program follows. A listing of the Hawes program appears on page 10.
Solution for a Quartic — (cont)

Listing for the Hawes Program

0000 `OR T V.017 18 APR` 0329 4* RCL D ) STD Y ) 0657 RCL 001 STD 029
0017 `87 w HAWES` 0336 STD 000 0 STD 001 0663 SBL CX - RCL R =
0027 LBL 51 0 STD 100 CE 0343 SBL SR 2= STD 030 0670 STD X (4* RCL L *
0035 `QUARTIC` 0352 RCL 001 2= STD 031 0678 RCL M -8* RCL N =<
0042 DFN F1: a@aa 0361 SBL CD SBL CE 0687 RCL L y^x 3>)
0049 DFN F2: b@bb 0367 SBL CR GTL DR 0693 STD 000 0 STD 001 4
0056 DFN F3: c@cc 0373 LBL CR RCL L +/- /4 0701 * RCL 028 = STD 002
0063 DFN F4: d@d@ 0381 +/- RCL 028 /2=+ 0709 4* RCL 029 =
0070 DFN F5: e@ee HLT 0390 RCL 032 /2= STD 036 0715 STD 003 CE `MPTHPLX`.
0078 LBL aa STD S `a` = 0399 RCL 029 /2=+ 0725 RUN SBR IND J
0086 COL 16 MGR = HLT 0406 RCL 033 /2= STD 037 0729 RCL 000 STD 030
0091 LBL bb STD T `b` = 0415 RCL L +/- /4=+ 0735 RCL 001 STD 031
0099 COL 16 MGR = HLT 0421 RCL 026 /2= 0741 SBL CD SBL CE
0104 COL cc STD U `c` = 0429 RCL 032 /2= STD 038 0747 SBL CR GTL DR
0112 COL 16 MGR = HLT 0430 RCL 029 /2= 0753 LBL DR CE DFN CLR
0117 LBL dd STD W `d` = 0445 RCL 033 /2= STD 039 0759 DFN F1:R=1R1
0125 COL 16 MGR = HLT 0454 RCL L +/- /4= 0766 DFN F2:R=2R2
0130 LBL ee STD W `e` = 0461 RCL 026 /2= 0773 DFN F3:R=3R3
0138 COL 16 MGR = CE `E` 0468 RCL 034 /2= STD 040 0780 DFN F4:R=4R4
0144 `DITTO` Y/N GTL S1 0477 RCL 029 +/- /2=+ 0787 DFN F5:EDED
0152 DFN CLR CE `WORKIN` 0485 RCL 035 /2= STD 041 0794 LBL R1 CE `R=1`
0161 `G` PRA RCL T `G` 0494 RCL L +/- /4= 0801 COL 16 MGR 036
0166 RCL S = STD L RCL U 0501 RCL 026 /2= 0806 DFN F1:ITAL HLT
0173 / RCL S = STD M 0508 RCL 034 /2= STD 042 0814 LBL I1 CE `I=1`
0179 RCL V / RCL S = 0517 RCL 029 +/- /2= 0821 COL 16 MGR 037
0185 STD N RCL W / RCL S 0525 RCL 035 /2= STD 043 0826 DFN F1:R=1R1 HLT
0192 = STD O DFN CLR 1 0534 RTN 0834 LBL R2 CE `R=2`
0198 STD 000 RCL M +/- 0535 LBL SR CE `MPTHPLX` 0841 COL 16 MGR 038
0204 STD 001 (4 RCL L ) 0546 RUN SBR IND 027 0846 DFN F2:IM=2R2 HLT
0212 RCL N ) (4 RCL D ) 0551 RCL 000 RTN 0854 LBL I2 CE `I=2`
0222 ) STD 002 (4 RCL L ) 0555 LBL CX ( RCL L x^2 0861 COL 16 MGR 039
0230 x^2 + RCL D ) +/- /+ 0562 3/4)-2* RCL M = 0866 DFN F2:R=2R2 HLT
0237 (4* RCL M + RCL D ) 0573 RTN 0874 LBL R3 CE `R=3`
0246 - RCL N x^2 0574 LBL CD RCL X + 0881 COL 16 MGR 040
0251 STD 003 CUB RCL 002 0580 RCL 030 = STD 000 0886 DFN F3:IM=3R3 HLT
0258 STD Q (4 RCL L x^2 0587 RCL 031 STD 001 0894 LBL I3 CE `I=3`
0265 /4- RCL M + 0593 SBL SR STD 032 0901 COL 16 MGR 041
0272 RCL 002 ) STD R HEX 0599 RCL 001 STD 033 RTN 0906 DFN F3:RE3R3 HLT
0279 7713 STD 027 766C 0606 LBL CE RCL X - 0914 LBL R4 CE `R=4`
0290 STD J DEC I EE 6 0612 RCL 030 = STD 000 0921 COL 16 MGR 042
0296 +/- STD 2 INV EE 0619 RCL 031 +/- STD 001 0926 DFN F4:IM=4R4 HLT
0301 RCL L ABS 1F> Z 0626 SBL SR STD 034 0934 LBL I4 CE `I=4`
0306 GTL NZ 0632 RCL 001 STD 035 RTN 0941 COL 16 MGR 043
0309 LBL RZ O STD 028 0639 LBL NZ RCL R 0946 DFN F4:RE4R4 HLT
0316 STD 029 SBL CX 0644 STD 000 O STD 001 0954 LBL ED GTL S1
0322 STD X ( RCL Q x^2 - 0651 SBL SR STD 028

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
diversity pattern: 1/11 = .0909..., 2/11 = .1818..., etc.

| 000 | 76 LBL | 012 | 42 STD | 024 | 32 X:IT | 036 | 32 X:IT | 048 | 91 R:S | 060 | 01 1 |
| 001 | 11 R | 013 | 00 00 | 026 | 32 44 +/- | 038 | 42 STD | 050 | 32 16Z | 062 | 01 1 |
| 002 | 01 1 | 014 | 32 X:IT | 026 | 63 EX* | 038 | 00 00 | 050 | 00 00 | 062 | 95 = |
| 003 | 00 0 | 015 | 72 ST+ | 027 | 00 00 | 039 | 73 PC* | 051 | 00 00 | 063 | 22 INV |
| 004 | 69 DP | 016 | 00 00 | 028 | 63 EX* | 040 | 00 00 | 052 | 39 39 | 064 | 77 GE |
| 005 | 17 17 | 017 | 97 DES | 029 | 00 00 | 041 | 29 CP | 053 | 01 1 | 065 | 00 00 |
| 006 | 29 CP | 018 | 00 00 | 030 | 74 SH* | 042 | 67 EQ | 054 | 62 HIF | 066 | 08 08 |
| 007 | 37 FR | 019 | 00 00 | 031 | 00 00 | 043 | 00 00 | 055 | 37 37 | 067 | 06 6 |
| 008 | 47 CMS | 020 | 15 15 | 032 | 97 DES | 044 | 49 49 | 056 | 32 X:IT | 068 | 69 DF |
| 009 | 32 X:IT | 021 | 32 X:IT | 033 | 00 00 | 045 | 32 X:IT | 057 | 62 HIF | 069 | 17 17 |
| 010 | 09 9 | 022 | 42 STD | 034 | 00 00 | 046 | 43 PCL | 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

\[
625 \text{ enter } 321 \rightarrow 200625 \text{ or } 321 \text{ enter } 625 \rightarrow 200625
\]

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.
ANOTHER SIMULTANEOUS EQUATION SOLUTION - 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 Statistics Library was provided.

Although the contents of the Mathematica Library 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 Mathematica Library 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 Statistics Library and the Electrical Engineering Library 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$="Simultaneous Equations":PRINT
20 DIM A(22,22),B(22),C(22)
30 GOSUB 800
40 IF PF=0 THEN 70
50 PRINT #1,A$
60 PRINT #1
70 INPUT "Number of equations = ? "; N
80 FOR I=1 TO N
90 FOR J=1 TO N
100 II$=STR$(I):JJ$=STR$(J)
110 $="A$="=A$(" &II$&“, &JJ$&")=
120 INPUT AA$;A(I,J)
130 PRINT #PF,AA$;
140 NEXT J
150 PRINT #PF:AA$;
160 NEXT I
170 BB$="B[" &II$&"
180 INPUT BB$;B(I)
190 PRINT #PF,BB$;
200 PRINT #PF:NEXT I
210 PRINT #PF:NEXT I
220 Z=0
230 IF N<0 THEN 290
240 IF A(1,1)=0 THEN
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 IF I=1+1
340 FOR J=1 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
400 Z=1
410 GOTO 720
420 IF L=I THEN 51
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=1 TO N
520 T=A(J,I)/A(I,I)
530 FOR K=I 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 THEN 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 IF X(I)=(B(I)-S)/A(I,I)
700 I=I-1
710 IF I>0 THEN 64
720 IF Z<1 THEN 7
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$;
780 NEXT I
790 NEXT I
800 IF PF=1 THEN CLOSE #1;PF=0
810 INPUT "Use printer? Y/N ";N$
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 device number ";D$
870 OPEN #1,D$,OUT PUT
880 IF D$="10"THEN PRINT #1,CHR$(18)
890 PRINT #PF:RETN
900 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:

\[
\begin{align*}
12a + 22b + 33c &= 15 \\
23a + 34b + 56c &= 18 \\
a + 2b + 3c &= 7
\end{align*}
\]

and

\[
\begin{align*}
b - 2c &= -8 \\
b + c &= 7 \\
2b - c &= 10
\end{align*}
\]

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.

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:

<table>
<thead>
<tr>
<th>Exact</th>
<th>Solution</th>
<th>Walters Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>56.00066891</td>
<td>1.0000000000</td>
</tr>
<tr>
<td>-1512</td>
<td>-1512.016156</td>
<td>0.9999999970</td>
</tr>
<tr>
<td>12600</td>
<td>12600.119</td>
<td>1.0000000020</td>
</tr>
<tr>
<td>-46200</td>
<td>-46200.391</td>
<td>0.9999999973</td>
</tr>
<tr>
<td>83160</td>
<td>83160.63758</td>
<td>0.9999999980</td>
</tr>
<tr>
<td>-72072</td>
<td>-72072.50508</td>
<td>0.9999999973</td>
</tr>
<tr>
<td>24024</td>
<td>24024.15504</td>
<td>0.9999999983</td>
</tr>
</tbody>
</table>

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.

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

\[
12a + 22b + 33c = 15 \quad \text{and} \quad 12a + 22b + 33c = 15
\]
\[
1a + 2b + 3c = 7 \quad \text{and} \quad 1a + 2b + 3c = 7
\]
\[
2a + 4b + 6c = 14 \quad \text{and} \quad 2a + 4b + 6c = 21
\]

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?

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 1.1666667e13 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 1.1666667e13 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 the user to change the definitions in a manner similar to many of the built-in menus.

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 Odda 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 Odda 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 appropriate 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 Daz tests, comparison tests, and flag tests. The TI-95 adds Yes/No tests. The only responses of the TI-59 to a test instruction are to execute 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 Daz 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" translation since the exact equivalent can not occur in TI-59 programs.

Translating Daz Tests

The TI-59 Daz instruction is in the format Daz 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 Daz 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 Daz register is not zero. Thus,

\[
\text{Daz XX N} \quad \text{in a TI-59 converts to} \quad \text{Daz XXX GTL NN} \quad \text{in a TI-95}
\]

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 255.

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:

- \( x \neq t \)  \hspace{1cm} EXC A
- \( CP \)  \hspace{1cm} EXC A O EXC A
- \( x = t \ N \)  \hspace{1cm} IF = A GTL NN
- \( x \geq t \ N \)  \hspace{1cm} INV IF < A GTL NN
- \( \text{INV} \ x = t \ N \)  \hspace{1cm} INV IF = A GTL NN
- \( \text{INV} \ x \geq t \ N \)  \hspace{1cm} 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 Daz 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 ∑+ 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 argument for y^n and INV y^n. 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. Laughey

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).

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:

\[
\begin{align*}
x^3 - 2x^2 + 10x - 4 \\
x^3 - 2x^2 + \left(\frac{4}{3}\right)x - 2/9 \\
x^3 - 10x\left(1/\sqrt{(x^2)}\right) + 1/\sqrt{(1225 - x^4)}
\end{align*}
\]

The second function is Peter Messer’s cubic test from V11N4PL6. 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 "0.01 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 0.2466929839685, 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 mdd,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 O = Saturday, 1 = Sunday, ... 6 = Friday. The TI-59 listing is:

```
000 02 LBL 035 32 INV 064 01 01 026 35 + 128 07 7 160 05 0 014 01 01 013 02 22 INV 065 22 INV 097 03 2 128 05 5 161 92 RTN 002 01 01 024 22 INV 036 25 INV 032 01 1 030 03 3 130 35 + 162 76 LEL 003 01 01 035 44 SUM 067 22 INV 039 65 0 033 13 0 131 53 X 034 13 0 004 01 01 036 01 01 068 44 SUM 100 43 RCL 152 43 RCL 164 53 005 01 01 037 65 0 069 01 01 101 01 01 133 03 03 152 43 RCL 164 53 006 43 RCL 033 04 4 070 45 + 102 76 - 104 55 - 166 05 05 007 01 01 039 22 INV 071 01 1 103 07 3 135 01 1 137 76 0 008 05 0 040 22 LDG 072 00 0 104 01 1 136 00 0 148 43 RCL 009 02 2 041 54 073 00 0 010 05 5 042 42 STD 074 50 0 170 03 3 136 45 0 172 45 0 010 01 01 043 03 03 075 42 STD 107 32 INT 139 59 INT 171 92 RTN 011 03 ? 044 32 INT 076 02 02 108 43 RCL 140 65 X 172 76 LEL 012 05 5 045 01 1 077 77 GE 109 01 01 141 91 X 073 14 0 013 05 7 046 05 5 078 77 GE 110 55 + 142 70 X 014 05 7 047 42 STD 079 01 1 107 76 2 143 05 5 075 59 X 015 05 4 048 01 1 080 03 3 112 01 1 144 54 X 176 10 E 016 05 5 049 77 GE 081 32 INT 113 22 INV 145 59 INT 177 42 STD 017 05 7 050 01 1 082 42 STD 114 44 SUM 146 54 0 178 96 0 018 05 6 051 03 3 083 01 01 115 03 03 147 92 RTN 179 94 0 019 05 8 052 02 2 084 77 GE 116 76 LEL 148 76 LEL 180 55 1 020 05 6 053 32 INT 085 77 GE 117 77 GE 149 11 A 181 07 0 021 05 8 054 53 0 086 53 0 118 53 0 150 10 E 182 54 0 022 05 8 055 3 0 087 03 3 119 43 RCL 151 42 STD 183 59 INT 023 05 8 056 43 RCL 088 06 6 120 03 03 152 04 04 184 65 1 024 05 8 057 01 01 089 05 5 121 55 2 153 05 00 185 07 0 025 05 8 058 01 1 090 65 2 122 04 04 154 92 RTN 186 95 0 026 05 8 059 01 1 091 43 RCL 123 54 0 155 76 LEL 187 43 RCL 027 05 8 060 00 0 092 03 03 124 59 INT 156 12 B 188 01 01 028 05 8 061 00 0 093 55 + 125 75 = 157 10 E 189 54 0 029 05 8 062 00 0 094 43 RCL 126 59 = 158 42 STD 190 52 RTN 030 05 8 063 42 STD 095 02 02 127 93 1 159 05 05 191 00 1
```

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 8AA 0093 581 INV IFK 000 0196 LBL GE ( RCL 003 /4 0005 DFN F2: S 3BE 0100 GTL xx 32 EXC 00 ( 0205 ) INT ( .75X( 0015 DFN F3: C 0CC 0109 ( RCL 001 /100) 0214 RCL 003 /100) INT * 0022 DFN F4: D 000 HLT 0118 STD 001 FRC ST- 001 0224 .75X) INT ) RTN 0030 LBL E+ (.4+ RCL 001 0125 +100) STD 002 INV 0231 LBL AR SBL E' 0040 +2.3) INT /+ + 0134 IFK 000 SBL xx 13 0237 STD 004 0 RTN 0048 GTL GE 0142 EXC RCL 001 INV 0242 LBL BB SBL E' 0051 LBL xx 0 1/x RTN 0149 IFK 000 GTL xx (365 0248 LBL 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 IFK 001 GTL xx FRC 0179 EXC 000 RCL 001 INV 0266 LBL DD ( SBL E' 0078 ST- 001 4 INV LDG 0186 IFK 000 GTL E+ 0274 STD 001 +/- /7) INT 0085 ) STD 003 EXC 000 1 0193 ST- 003 0282 87+ 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 such better use of the TI-95 capability.
Translating ML-20 FOR THE TI-95 - (cont)

<table>
<thead>
<tr>
<th>Line</th>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>CMS DFN F1:D1#AA</td>
<td>0170 100 STD C INV</td>
</tr>
<tr>
<td>0008</td>
<td>DFN F2:D2#BB</td>
<td>0177 IF&lt; R GTL xx 1</td>
</tr>
<tr>
<td>0015</td>
<td>DFN F3:D3#CC</td>
<td>0183 IF&gt; C GTL xx 13 INV</td>
</tr>
<tr>
<td>0022</td>
<td>DFN F4 CLR</td>
<td>0191 IF&gt; B GTL xx 0 INV</td>
</tr>
<tr>
<td>0025</td>
<td>DFN F5:D5#DD 'CAL'</td>
<td>0198 IF&lt; B GTL xx (365+</td>
</tr>
<tr>
<td>0035</td>
<td>'ENDAR RGUTINE' PRU</td>
<td>0208 RCL D + RCL C <em>3+1</em></td>
</tr>
<tr>
<td>0049</td>
<td>PRU CE 'ENTER: mad'</td>
<td>0217 RCL B -31+3 INV</td>
</tr>
<tr>
<td>0061</td>
<td>'d. yyyy' RTN</td>
<td>0225 IF&gt; B GTL E+ 1</td>
</tr>
<tr>
<td>0068</td>
<td>LBL E+ (.4* RCL B +</td>
<td>0231 ST- D</td>
</tr>
<tr>
<td>0078</td>
<td>2.3) INT +/ / +</td>
<td>0233 LBL GE ( RCL D /4</td>
</tr>
<tr>
<td>0085</td>
<td>RTL GE</td>
<td>0242 INT - (.75+ ( RCL D /</td>
</tr>
<tr>
<td>0088</td>
<td>LBL xx 'ERROR'</td>
<td>0253 100)) INT *75) INT</td>
</tr>
<tr>
<td>0096</td>
<td>SF 01 PRU CLR 'RE-'</td>
<td>0264 ) RTN</td>
</tr>
<tr>
<td>0103</td>
<td>'ENTER DATE' RTN</td>
<td>0266 LBL AA SBL E' TF 01</td>
</tr>
<tr>
<td>0114</td>
<td>LBL E' RF O1 (</td>
<td>0274 HLT STD E 'FIRST D'</td>
</tr>
<tr>
<td>0120</td>
<td>STD B 0 INV IF&lt; B</td>
<td>0284 'ATE IN' RTN</td>
</tr>
<tr>
<td>0126</td>
<td>RTL xx RCL B FRC</td>
<td>0291 LBL BB SBL E' TF 01</td>
</tr>
<tr>
<td>0132</td>
<td>ST- B #4 INV LOG )</td>
<td>0299 HLT STD F 'SECOND'</td>
</tr>
<tr>
<td>0139</td>
<td>STD D 1581 INV</td>
<td>0309 'DATE IN' RTN</td>
</tr>
<tr>
<td>0146</td>
<td>IF&lt; D GTL xx 32</td>
<td>0317 LBL CC ( RCL F -</td>
</tr>
<tr>
<td>0153</td>
<td>STD A ( RCL B /100</td>
<td>0324 RCL E ) 'DAYS='</td>
</tr>
<tr>
<td>0163</td>
<td>) STD B FRC ST- B #</td>
<td>0335 COL 16 MRG = COL 16</td>
</tr>
</tbody>
</table>

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 usage. 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:

<table>
<thead>
<tr>
<th>Model</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI-74</td>
<td>$99.97</td>
</tr>
<tr>
<td>ROM Modules</td>
<td>$29.80</td>
</tr>
<tr>
<td>TI-95</td>
<td>129.90</td>
</tr>
</tbody>
</table>
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 628132 - 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.
36 DIGIT DMS TO RADIANS - L. Leeds. V9N3P11 presented Peter Neaser'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:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>035</td>
<td>55</td>
<td>070 42.00</td>
</tr>
<tr>
<td>036</td>
<td>.00</td>
<td>071 53.55</td>
</tr>
<tr>
<td>037</td>
<td>00</td>
<td>072 42.00</td>
</tr>
<tr>
<td>038</td>
<td>.00</td>
<td>073 54.54</td>
</tr>
<tr>
<td>039</td>
<td>00</td>
<td>074 42.00</td>
</tr>
<tr>
<td>040</td>
<td>95</td>
<td>075 55.55</td>
</tr>
<tr>
<td>041</td>
<td>65</td>
<td>076 42.00</td>
</tr>
<tr>
<td>042</td>
<td>45</td>
<td>077 56.56</td>
</tr>
<tr>
<td>043</td>
<td>06</td>
<td>078 42.00</td>
</tr>
<tr>
<td>044</td>
<td>95</td>
<td>079 57.57</td>
</tr>
<tr>
<td>045</td>
<td>44</td>
<td>080 05.05</td>
</tr>
<tr>
<td>046</td>
<td>8</td>
<td>081 01.01</td>
</tr>
<tr>
<td>047</td>
<td>9</td>
<td>082 42.00</td>
</tr>
<tr>
<td>048</td>
<td>76</td>
<td>083 01.01</td>
</tr>
<tr>
<td>049</td>
<td>13</td>
<td>084 06.06</td>
</tr>
<tr>
<td>050</td>
<td>61</td>
<td>085 01.01</td>
</tr>
<tr>
<td>051</td>
<td>03</td>
<td>086 42.00</td>
</tr>
<tr>
<td>052</td>
<td>15</td>
<td>087 02.02</td>
</tr>
<tr>
<td>053</td>
<td>76</td>
<td>088 01.01</td>
</tr>
<tr>
<td>054</td>
<td>14</td>
<td>089 01.01</td>
</tr>
<tr>
<td>055</td>
<td>61</td>
<td>090 42.00</td>
</tr>
<tr>
<td>056</td>
<td>01</td>
<td>091 03.03</td>
</tr>
<tr>
<td>057</td>
<td>72</td>
<td>092 07.07</td>
</tr>
<tr>
<td>058</td>
<td>43</td>
<td>093 42.00</td>
</tr>
<tr>
<td>059</td>
<td>58</td>
<td>094 04.04</td>
</tr>
<tr>
<td>060</td>
<td>75</td>
<td>095 02.02</td>
</tr>
<tr>
<td>061</td>
<td>53</td>
<td>096 05.05</td>
</tr>
<tr>
<td>062</td>
<td>45</td>
<td>097 72.07</td>
</tr>
<tr>
<td>063</td>
<td>52</td>
<td>098 01.01</td>
</tr>
<tr>
<td>064</td>
<td>95</td>
<td>099 02.02</td>
</tr>
<tr>
<td>065</td>
<td>65</td>
<td>100 06.06</td>
</tr>
<tr>
<td>066</td>
<td>42</td>
<td>101 01.01</td>
</tr>
<tr>
<td>067</td>
<td>06</td>
<td>102 95</td>
</tr>
<tr>
<td>068</td>
<td>95</td>
<td>103 27</td>
</tr>
<tr>
<td>069</td>
<td>42</td>
<td>104 03.03</td>
</tr>
</tbody>
</table>

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.

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 particularly 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 *Statistica Library* 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.
### Program Listing:

<table>
<thead>
<tr>
<th>Lines</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>\texttt{AS=&quot;Curve Fitter&quot;;PRINT 1}</td>
</tr>
<tr>
<td>11</td>
<td>\texttt{B(1)=&quot;Linear&quot;}</td>
</tr>
<tr>
<td>12</td>
<td>\texttt{B(2)=&quot;Exponential&quot;}</td>
</tr>
<tr>
<td>13</td>
<td>\texttt{B(3)=&quot;Power&quot;}</td>
</tr>
<tr>
<td>14</td>
<td>\texttt{B(4)=&quot;Logarithmic&quot;}</td>
</tr>
<tr>
<td>15</td>
<td>\texttt{B(5)=&quot;Hyperbolic&quot;}</td>
</tr>
<tr>
<td>16</td>
<td>\texttt{B(6)=&quot;Find the best Fit&quot;}</td>
</tr>
<tr>
<td>20</td>
<td>\texttt{DIM X(50);Y(50)}</td>
</tr>
<tr>
<td>25</td>
<td>\texttt{INPUT &quot;Use Printer? Y/N&quot;:I$}</td>
</tr>
<tr>
<td>30</td>
<td>\texttt{IF I$=&quot;Y&quot; THEN PRINT NPRINT=1 ELSE 70}</td>
</tr>
<tr>
<td>35</td>
<td>\texttt{PRINT &quot;Device Numbers&quot;}</td>
</tr>
<tr>
<td>40</td>
<td>\texttt{PAUSE 1}</td>
</tr>
<tr>
<td>45</td>
<td>\texttt{PRINT &quot;For the HX-100&quot;}</td>
</tr>
<tr>
<td>50</td>
<td>\texttt{PAUSE 1}</td>
</tr>
<tr>
<td>55</td>
<td>\texttt{OPEN #1:INPUT 1}</td>
</tr>
<tr>
<td>60</td>
<td>\texttt{IF D$=&quot;10&quot; THEN PRINT #1:CHR$(18)}</td>
</tr>
<tr>
<td>65</td>
<td>\texttt{PRINT #1:PRINT #1:A$}</td>
</tr>
<tr>
<td>80</td>
<td>\texttt{PRINT #N:PAUSE ALL}</td>
</tr>
<tr>
<td>100</td>
<td>\texttt{PRINT &quot;End Input by Entering E:PAUSE 1&quot;}</td>
</tr>
<tr>
<td>105</td>
<td>\texttt{PRINT &quot;Zero or negative inputs will:&quot;}</td>
</tr>
<tr>
<td>110</td>
<td>\texttt{PRINT &quot;prevent some solution options:&quot;}</td>
</tr>
<tr>
<td>115</td>
<td>\texttt{N=1}</td>
</tr>
<tr>
<td>120</td>
<td>\texttt{X$=&quot;=&quot;}</td>
</tr>
<tr>
<td>125</td>
<td>\texttt{Y$=&quot;=&quot;}</td>
</tr>
<tr>
<td>130</td>
<td>\texttt{INPUT X$:Y$:IF X$:IF X$=&quot;END DR X$:&quot;THEN 170}</td>
</tr>
<tr>
<td>135</td>
<td>\texttt{INPUT Y$:Y$:IF Y$=&quot;END DR Y$:&quot;THEN 170}</td>
</tr>
<tr>
<td>140</td>
<td>\texttt{X(N)=VAL(X$)}</td>
</tr>
<tr>
<td>145</td>
<td>\texttt{Y(N)=VAL(Y$)}</td>
</tr>
<tr>
<td>150</td>
<td>\texttt{IF PN=0 THEN 165}</td>
</tr>
<tr>
<td>155</td>
<td>\texttt{PRINT #N:X$:Y$:PAUSE ALL}</td>
</tr>
<tr>
<td>160</td>
<td>\texttt{PAUSE #N:Y$:PAUSE ALL}</td>
</tr>
<tr>
<td>165</td>
<td>\texttt{N=4:GOTO 130}</td>
</tr>
<tr>
<td>170</td>
<td>\texttt{N=1}</td>
</tr>
<tr>
<td>175</td>
<td>\texttt{PRINT &quot;The regression options are:&quot;}</td>
</tr>
<tr>
<td>180</td>
<td>\texttt{FDR 1=1 TO 6}</td>
</tr>
<tr>
<td>185</td>
<td>\texttt{PRINT STR$(1)&amp;&quot; &amp; B$(1)&amp;&quot;:PAUSE 2}</td>
</tr>
<tr>
<td>190</td>
<td>\texttt{NEXT 1}</td>
</tr>
<tr>
<td>200</td>
<td>\texttt{INPUT &quot;Which Option?&quot;;I$}</td>
</tr>
</tbody>
</table>

### Code Snippet 1:

```plaintext
10 AS="Curve Fitter";PRINT 1
11 B(1)="Linear"
12 B(2)="Exponential"
13 B(3)="Power"
14 B(4)="Logarithmic"
15 B(5)="Hyperbolic"
16 B(6)="Find the best Fit"
20 DIM X(50);Y(50)
25 INPUT "Use Printer? Y/N":I$
30 IF I$="Y" THEN PRINT NPRINT=1 ELSE 70
35 PRINT "Device Numbers"
40 PAUSE 1
45 PRINT "For the HX-100"
50 PAUSE 1
55 OPEN #1:INPUT 1
60 IF D$="10" THEN PRINT #1:CHR$(18)
65 PRINT #1:PRINT #1:A$
80 PRINT #N:PAUSE ALL
100 PRINT "End Input by Entering E:PAUSE 1"
105 PRINT "Zero or negative inputs will:"
110 PRINT "prevent some solution options:" PAUSE 1
115 N=1
120 X$="="
125 Y$="="
130 INPUT X$:Y$:IF X$:IF X$="END DR X$":THEN 170
135 INPUT Y$:Y$:IF Y$="END DR Y$":THEN 170
140 X(N)=VAL(X$)
145 Y(N)=VAL(Y$)
150 IF PN=0 THEN 165
155 PRINT #N:X$:Y$:PAUSE ALL
160 PRINT #N:Y$:PAUSE ALL
165 N=4:GOTO 130
170 N=1
175 PRINT "The regression options are:" PAUSE 1
180 FDR 1=1 TO 6
185 PRINT STR$(1)&" B$(1)&":PAUSE 2
190 NEXT 1
200 INPUT "Which Option?";I$
```

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 display. 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!

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, F. Messer, H. Ladd, W. Hawes .. 8
TI-59 DIAGNOSTICS, S. Gerver ................................ 11
AOS CALCULATION FROM HP, W. Witsmer .................... 11
ANOTHER SIMULTANEOUS EQUATION SOLUTION ................ 12
TRANSFERS FROM THE TI-95 TO THE TI-95 ................. 16
ROOT-FINDER FOR THE TI-95, D. Laughery ................. 19
TRANSFILING ML-20 FOR THE TI-95, S. Gerver ............ 20
MORE PFX PROGRAM AVAILABILITY, S. L. Lee .............. 22
36 DIGITS 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".