TI PPC NOTES

NEWSLETTER OF THE TI PROGRAMMABLE CALCULATOR CLUB

P.O. Box 1421, Largo, FL 34294

Volume 8, Number 6 November/December 1983

With this issue we have completed the fourth year of our newsletter, and the first under a new editor. It has been an eventful year for owners of programmable calculators manufactured by Texas Instruments; the withdrawal of the TI-88, the termination of production of the TI-58/59, the termination of PPX Exchange, and the announcement of the TI -66. It was no less hectic for the owners of home computers with the arrival of the CC-40 and the demise of the TI-99/4A.

The dominant theme for this issue is the solution of systems of linear equations, with particular emphasis on the use of the ML-02 program. This was one more part of a continuing effort to answer questions left over from earlier years. The big news is that TI-66's are finally becoming available. The sad news is a recently received letter from Elek-Tek refunding my deposit on Hex-bus components for the CC-40, and suggesting that I reorder in the second quarter of 1984.

For the future the newsletter will continue to support the TI-58/59 by publishing programs, programming tricks, and the like. The backlog from 1983 includes extension of the solutions for linear equations to least squares curve fitting, business related programs, additional high resolutions graphics, etc. Information for the TI-57, the TI-57LCD and even for older programmables will be published if reader interest is apparent. The newsletter willl also provide expanding coverage for the CC-40 and the TI-66. Dave Leising is already searching for the kind of extended modes which were so helpful with the TI-59.

In the last issue I asked whether members would be willing to share programs as a means to fill in for the loss of PPX. I received a wide variety of responses ranging from suggestions that the club purchase a set of microfiche to one volunteer to set up an exchange with prices similar to PPX. I received offers to share some 300 programs. I plan to sort all of that out and make an announcement in the first issue for 1984. In the meantime, if you have a need for a particular program, please write and I will try to put you in contact with a source. Send a SASE for a reply.

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 the source 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.

ERRATA:

BA-55 Professional Business Analyst - V8N5P9 - Square root and square notation was inadvertently omitted in three places, twice in the test, and once in the program listing. In the program, step 05 should be $\sqrt{x}$ and step 06 should be x^2 . Similarly, in the next to the last line of the fifth paragraph and in the second line of the sixth paragraph, the sequence x x should have been $\sqrt{x}$ x^2 .

Periodic Table - V7N4/5P21 - Robert Prins writes that data card A contains two errors. The integer part of register 18 should be 1335, not 13, since the symbol for Argon is Ar, not A. Register 43 should be negative since Technetium is a human made element.

Editor's Note: V7N4/5P21 did not identify the author of the program so we cannot determine his age. I checked my college textbooks, circa 1946, and they all show the symbol for Argon as A. Somehow, I am feeling my age.

Still More on Numerical Precision - V8N5P26 listed exact solutions to the test of numerical precision proposed by Bob Fruit in V8N4P4. Carl Rabe writes to report that the last five digits (the 45th through the 49th) in his list of the first 49 should be 58270, not 53760. That means that the solution to 49 digits is now:

2260.48792 47960 86067 64793 83933 44540 34437 92694 58270

Carl also reports that he has obtained the exact solution out through 75 digits.

HP-41 vs HP-67 - In V8N4P2 Laurance Leeds reported that using the EE INV EE routine with the TI-59 does not always give the same answer as would be obtained for the similar calculations with a HP-67. He gave the example of the y^x function with x = 2.5 and y = 2.543210631 . The exact solution rounded to ten digits is 10.314 68159 which can be obtained from the TI-59. The HP-67 yields 10.314 68158 , a difference of one in the last digit. Laurance asked whether the HP-41 makes the same mistake. Gene Friel reports that it does not make that mistake, but obtains the correct solution rounded to ten digits. That is in agreement with the editor's experience with the HP-11.

---

13 Digit Modulo 30 SFF - On V8N4P15 in the seventeenth line the value should be 0285 not 0235 as in the text. The following example is correct.

Ulam's Conjecture is Collatz's Algorithm - V6N9/10P13 described this conjecture on the characteristics of numbers, and reported that the attribution to Ulam was questionable. Professor Widmer has identified the correct source. See page 13 of this issue for details.

0^b QUIRKS - Robert Prins. Soon after the TI-59 came on the market G. E. Wilkins reported in the July 1977 issue of 52 Notes that

0 y^x 0 = 1

0 y^x n = 0 if n is greater than zero

0 y^x n = 9.9999999 99 flashing = error

But Robert reports that some very strange quirks were reported in the 20-79 issue of the German Funkschau, pages 74-76:

I. $0^b - a$ where $0 < a < 1$ and $0 < b < 1$

Examples: 0 y^x .3 - .3 = -.03 instead of -0.3

0 y^x .5 - .01 = -0.0001 instead of -0.01

Putting parenthesis around the 0 y^x b cures the problem.

II. $a \cdot 0^b$ where $0 < a < 1$ and $b < 0$

Examples: .5 x 0 y^x 2 +/- = -5. EE 99 without flashing.

.2 +/- x 0 y^x 3 +/- = 2. EE 99 without flashing.

III. $a \pm 0^b$ where $-1 < b < 0$

Examples: 10 + 0 y^x .5 +/- = 10 no flashing.

20 - 0 y^x .9 +/- = 20 no flashing.

IV. $a/0^b$ where $a < 10$ and $-1 < b < 0$

Example: 5 DIV 0 y^x .5 +/- = -5. EE 98 no flashing.

BRAIN TEASER - Using a single digit integer three, and only three, times, use the functions of the calculator to arrive at 24. For example:

22 + 2 = 24

4 x (4 + 4 $\sqrt{x}$ = 24

Hint: Although you may only use the given single digit integer three times, and may not use any other number, you may need the factorial function. If so, it is permissible to use the sequence Pgm 16 from the Master Library to find a factorial.

TI-59 AVAILABILITY - I have received many inquiries as to sources of TI-59's and related material. Maurice Swinnen has found several firms in the Washington, D.C. area which still have TI-59's. The firms and their telephone numbers are:

Washington Calculators in Silver Springs, MD	301-384-2010
Shavitz Calculators in Rockville, MD.	301-340-0200
Bell & Co. Several stores in the area	301-881-2000
Best Products, Greenbelt, MD	301-474-5500

If you are interested, call direct. Don't call Maurice. The latest Elek-Tek catalog (Volume VII, page 15) continues to list printer paper, magnetic cards and battery packs. You can call 800-621-1269 toll-free.

SOLVING SIMULTANEOUS EQUATIONS - P. Hanson. V7N3P10 of TI PPC Notes asked:

> "Does anyone have any idea how to use the ML-02 routines as subroutines in a program to enable one to solve simultaneous equations? TI is not able to come up with a complete answer either. Any good solution would be accepted with thanks and published. Several members have been asking me about this problem."

Page 10 of the manual for the Master Library module gives detailed instructions for the use of the program from the keyboard; but, the module programs react differently to a call from a user program than to a call from the keyboard. PPX Exchange described the difference in the January 1978 issue:

> "Some people are confused about how to use Solid State Module programs as subroutines in their own programs. The difficulty arises because they think that it is sufficient to call a subroutine by using a sequence like Pgm-15-A to access the whole program. Each particular part of the program that the person wants to use must be preceded by Pgm-mm-U where U is the label associated with that part."

Unfortunately, even that information is not sufficient to enable a user to successfully access module routines from his program. The first key to a successful implementation was the discovery that the Pgm-mm-SBR-nnn sequence could be used to call a code sequence in module program mm beginning at step nnn and ending with the first RTN encountered. In V2N7P2 of 52 Notes (July 1977) the editor, Richard Vanderburgh, demonstrated the technique in a program titled "Enhanced ML-02 Program". The program provides annotation for the entries and for the solutions.

The Pgm-mm-SBR-nnn feature is not described in Personal Programming and was not described in PPX Exchange until almost two years later. The May 1979 issue stated:

> "Mr. Pat Eaton, Scarborough, Canada, has discovered another way of calling subroutines from TI Solid State Modules. In addition to calling label addresses, direct addresses may be used. For example, the key sequence Pgm-mm-SBR-nnn can be used, where mm is the library program number and nnn is the library program location (address) from which you wish to begin execution. The advantage of using direct addresses is that any group of steps may be directly accessed without beginning at a label. There is one golden rule to remember: your selected group of steps must end with an 'INV SBR'. This method provides added flexibility when using Solid State Software."

The Pgm-mm-SBR-nnn capability was not reflected in the reprinting of Personal Programming in 1979 to include TI-58C information. But neither did the reprinting include a discussion of the ability to Dsz on memory registers other than 0 to 9. The Pgm-mm-SBR-nnn concept was illustrated in Exercise 3-1 in the Advanced Programming section (page III-10) of the TI Programmable 59 Workbook. In that exercise the program sequence N-Pgm-01-SBR-012 was used to clear data memories 01 through N. The workbook was described in V8N1P10. Workbooks are available from Educalc for $2.95 . Ask for Stock # P-192.

Solving Simultaneous Equations - (cont)

The use of Pgm-02-SBR-nnn techniques are well established for integrating subroutines of ML-02 into polynomial regression programs. Examples include:

* William Skillman's "Polynomial Least Squares Fit" program in V4N2P5 of 52 Notes.

* Thomas Wysmuller's "Polynomial Regression" program in the November/December 1980 issue of PPX Exchange.

* My program "Polynomial Curve Fit with Errors (PPX 208059).

* A recently submitted optimization of the Wysmuller program by Gene Friel.

Fred Fish's book "User Survival Guide for TI-58/59 Master Library" (see V6N6/7P1) includes information on many of the options in using the Pgm-mm-SBR-mmm technique, and also includes useful examples of the sequence Pgm-mm -SBR-N, where N is a common label. Fred's book is currently available from The Educalc Mail Store, 27953 Cabot Road, Laguna Niguel, CA 92677 for $12.95 plus $1.00 for shipping and handling. Ask for Stock # P-150.

But there is a better way--the Pgm-mm-R/S method. I discovered it while scanning old issues for references to the 0^b quirks discussed elsewhere in this issue. That technique allows the user to implement the instructions from the module manuals directly, without the downloading and analysis required for the Pgm-mm-SBR-nnn method. V3N1P3 of 52 Notes reported:

"Roger Gentry decided to try out sequences of the form Pgm-MM-R/S (suggested on page V-62 of the Owner's Manual, but acknowledged by TI as a mistake). Results appear to be program-dependent as well as machine-state-dependent. ... "

V3N3P5 of 52 Notes expanded on the use of the Pgm-mm-R/S sequence:

"John Hirsch has found that there are indeed applications for sequences of the form Pgm-mm-R/S in user programs. After a CROM routine in Pgm mm has been called bya user program, the effect of Pgm-mm-R/S is to resume CROM execution at the step following the last RTN encountered, just as a plain R/S does during keyboard interaction with a CROM program.

A CROM pointer maintains this restart address through intermediate user-code execution, but not following a manual or programmed call to another specified Pgm label or step number. For example, write:

Lbl A Pgm 2 A 1 Pgm 2 B R/S Lbl B Pgm 2 R/S R/S GTO B.

Press A, see 1; key in a datum, press R/S, repeat for say three data. Now go elsewhere in user memory, write a small routine (that doesn't clobber the ML-02 registers, but may use all six subroutine levels), run it, then key the next datum for input to ML-02, and press B; key a few more data with R/S's and find that all data were stored sequentially as expected in memory registers 8, 9, etc."

Solving Simultaneous Equations - (cont)

Bill Skillman incorporated the Pgm-02-R/S sequence in his Polynomial Least Squares Fit in V4N2P5 of 52 Notes to recall the solution but not to control the input of the vector values. Use of the technique seems to have been largely lost thereafter. A year later Thomas Wysmuller's Polynomial Regression program in the November/December 1980 issue of PPX Exchange used the Pgm-02-SBR-097, Pgm-02-SBR-355, and Pgm-02-SBR- 529 techniques. A recent submission by Robert Prins doesn't use the Pgm-02-R/S technique; but Gene Friel's submission uses a mix of Pgm-mm-SBR-nnn and Pgm-mm-R/S. Programs which illustrate the various techniques are presented on the following pages.

ENHANCED ML-02 PROGRAM - Richard Vanderburgh. This program is copied directly from page V2N7P2 of the July 1977 issue of 52 Notes. Page V2N7P1 of the same issue included the earliest reference to the Pgm-mm-SBR-nnn technique.

Program Listing:

```
000  76 LBL
001  14  D
002  32 X:T
003  01  1
004  44 SUM
005  89  89
006  43 RCL
007  89  89
008  22 INV
009  67  EQ
010  78 Σ+
011  85  +
012  02  2
013  95  =
014  42 STO
015  89  89
016  76 LBL
017  78 Σ+
018  69 OP
019  04  04
020  92 RTN
021  76 LBL
022  15  E
023  91 R/S
024  69 OP
025  06  06
026  72 ST*
027  01  01
028  76 LBL
029  87 IFF
030  69 OP
031  21  21
032  43 RCL
033  01  01
034  91 R/S
035  72 ST*
036  01  01
037  99 PRT
038  43 RCL
039  01  01
040  22 INV
041  67  EQ
042  87 IFF
043  98 ADV
044  98 ADV
045  92 RTN
046  76 LBL
047  11  A
048  42 STO
049  07  07
050  42 STO
051  00  00
052  33 X²
053  85  +
054  07  7
055  95  =
056  32 X:T
057  03  3
058  01  1
059  69 OP
060  04  04
061  43 RCL
062  07  07
063  69 OP
064  06  06
065  98 ADV
066  01  1
067  03  3
068  69 OP
069  04  04
070  08  8
071  42 STO
072  01  01
073  15  E
074  69 OP
075  00  00
076  01  1
077  06  6
078  01  1
079  07  7
080  03  3
081  07  7
082  69 OP
083  03  03
084  69 OP
085  05  05
086  36 PGM
087  02  02
088  13  C
089  98 ADV
090  98 ADV
091  91 R/S
092  76 LBL
093  12  B
094  01  1
095  36 PGM
096  02  02
097  14  D
098  43 RCL
099  07  07
100  42 STO
101  06  06
102  69 OP
103  00  00
104  01  1
105  04  4
106  69 OP
107  02  02
108  69 OP
109  05  05
110  25 CLR
111  76 LBL
112  67  EQ
113  91 R/S
114  36 PGM
115  02  02
116  71 SBR
117  03  03
118  55  55
119  97 DSZ
120  06  06
121  67  EQ
122  25 CLR
123  36 PGM
124  02  02
125  15  E
126  36 PGM
127  02  02
128  16 A'
129  04  4
130  04  4
131  00  0
132  02  2
133  42 STO
134  89  89
135  69 OP
136  04  04
137  43 RCL
138  07  07
139  42 STO
140  06  06
141  76 LBL
142  79  x̄
143  73 RC*
144  01  01
145  69 OP
146  06  06
147  69 OP
148  21  21
149  04  4
150  04  4
151  00  0
152  08  8
153  14  D
154  97 DSZ
155  06  06
156  79  x̄
157  98 ADV
158  98 ADV
159  98 ADV
160  25 CLR
161  91 R/S
162  76 LBL
163  13  C
164  25 CLR
165  36 PGM
166  02  02
167  17 B'
168  69 OP
169  00  00
170  02  2
171  04  4
172  03  3
173  01  1
174  04  4
175  02  2
176  69 OP
177  03  03
178  69 OP
179  05  05
180  98 ADV
181  01  1
182  05  5
183  00  0
184  02  2
185  42 STO
186  89  89
187  69 OP
188  04  04
189  01  1
190  76 LBL
191  88 DMS
192  36 PGM
193  02  02
194  18 C'
195  43 RCL
196  07  07
197  42 STO
198  06  06
199  76 LBL
200  89  π
201  69 OP
202  21  21
203  69 OP
204  24  24
205  73 RC*
206  01  01
207  69 OP
208  06  06
209  69 OP
210  36  36
211  76 LBL
212  77  GE
213  36 PGM
214  02  02
215  71 SBR
216  08  08
217  60  60
218  97 DSZ
219  06  06
220  77  GE
221  98 ADV
222  01  1
223  05  5
224  00  0
225  08  8
226  14  D
227  01  1
228  85  +
229  43 RCL
230  03  03
231  95  =
232  97 DSZ
233  00  00
234  88 DMS
235  98 ADV
236  98 ADV
237  98 ADV
238  25 CLR
239  91 R/S
```

Enhanced ML-02 - (cont)

User Instructions:

1. Enter the program. You must set the partitioning at 9-Op-17.

2. Enter the order of the matrix (n) and press A. The value of n is printed with annotation. An 8 is displayed which is the address of the first storage register to be used for storage of the matrix elements.

3. Enter the A_{11} element and press R/S. The element is printed and the address of the next register to be used is displayed. Continue to enter the remaining elements using columnwise catenation in the same manner as the instructions in the Master Library manual. Correct any erroneous entries before the final entry using direct STO techniques from the keyboard. When the final matrix element is entered "DET" is printed and the program proceeds directly to the calculation of the determinant. The value for the determinant is printed.

4.a. For simultaneous equations: If the value of the determinant was not zero, press B. "B" is printed. Enter the first element of the vector and press R/S. The value of the element is printed. Repeat for the remaining elements. When the last vector element has been entered the program proceeds and prints the solution with annotation.

4.b. To obtain the inverse of the matrix press C. When the calculations are complete the annotation "INV" is printed. The elements of the inverse are printed with annotation by column.

Of course, to have generated that program the user must have either downloaded and analyzed the ML-02 program, or must have had access to someone else's work in that area; an example is the Fred Fish book. An alternative is to use the Pgm-mm-R/S technique.

ENHANCED ML-02 USING PGM-MM-R/S - To illustrate how easy it is to use the Pgm-mm-R/S technique I have written an equivalent enhanced ML-02 program. With two exceptions the program was written by simply following the instructions for ML-02 in the Master Library manual. Both exceptions involve circumvention of the automatic printout mentioned at the bottom of page 12 of the manual so that annotation can be provided using Op-06 commands. The first exception appears at locations 002 through 020. If the sequence Pgm-02-A had been used instead there would have been an unwanted printout of the order of the solution without annotation. The second exception is at locations 116 through 142. A sequence:

RCL-07-STO-00-LBL-$\bar{x}$-Pgm-02-R/S-Dsz-0-$\bar{x}$

of only twelve steps would have sufficed if we had been satisfied with output of the solution without annotation. Other changes were made in the output format, both to the display and to the printer. For entry of both the matrix elements and the vector elements, the address into which the last element was entered is displayed. This permits the user to check the printout and correct with direct STO techniques. This display is

Enhanced ML-02 Using Pgm-mm-R/S - (cont)

particularly important for the entry of the vector elements. If pivoting had occurred during the calculation of the determinant the vector elements are not placed into memory registers in ascending order, but rather are placed according to a pivoting index. Thus, knowing the next memory register to be used is of no help in determining where to direct store a correction for the last element entered.

User Instructions:

1. Enter the program. You may use the startup partition for recording the program since the correct partitioning is set by locations 008-010.

2. Enter the order of the matrix (n) and press A. The value of n is printed with annotation. The printer advances, and prints an "A" to indicate it is ready to accept matrix elements. A "1" appears in the display.

3. Enter the A_{11} element and press R/S. The element is printed and the address into which the element was stored is displayed. Correct an erroneous entry with direct STO techniques. Continue to enter the remaining matrix elements using columnwise catenation in the same manner as the instructions in the Master Library manual. Columns are separated by an advance in the printout. When the final element has been entered the calculator stops with the storage location for the last element in the display. This permits correction of the last element prior to proceeding with the calculation of the determinant.

4. Press C or R/S when you are ready to calculate the determinant. "DET" is printed, then the determinant is calculated and printed.

5.a. For simultaneous equations: If the value of the determinant was not zero, press D or R/S. A "B" is printed to indicate the calculator is ready to accept vector elements. Enter the first element of the vector and press R/S. The value of the element is printed. The address into which the element was stored is displayed. Correct an erroneously entered element by direct STO techniques. (Note: If pivoting occurred during the calculation of the determinant then the elements will not be stored in ascending order.) When the final vector element has been entered the calculator stops with the address into which it was stored in the display. This permits correction of the last element before proceeding to calculation of the solution. Press D or R/S to solve. The solution is printed with annotation.

5.b. To obtain the inverse of the matrix press E after finding the determinant (step 4). "INV" is printed, and the elements of the inverse matrix are calculated and printed. Columns are separated by an advance.

Sample Printouts:

Sample printouts are provided for both the Vanderburgh "Enhanced ML-02 program and this "Enhanced ML-02 Using Pgm-mm-R/S" program on the next page. The problem solved is the same one as on page 12 of the Master Library manual.

Enhanced ML-02 Using Pgm-mm-R/S - (cont)

Simultaneous Equations

Pgm-02-SBR-nnn

```
        3.      N

        4.      A
        8.
        2.
        8.
        8.
        0.
        0.
        8.
        1.

         DET

       96.

       B
        4.
        4.
        6.

        4.      X1
     -1.5       X2
      -2.       X3
```

Pgm-02-R/S

```
        3.      N

       A

        4.
        8.
        2.

        8.
        8.
        0.

        0.
        8.
        1.

      DET

       96.

      B

        4.
        4.
        6.

        4.      X1
     -1.5       X2
      -2.       X3
```

Matrix Inverse

Pgm-02-SBR-nnn

```
        3.      N

        4.      A
        8.
        2.
        8.
        8.
        0.
        0.
        8.
        1.

          DET

       96.

         INV

 .0833333333    C1
 .0833333333
-.1666666667

-.0833333333    C2
 .0416666667
 .1666666667

 .6666666667    C3
-.3333333333
-.3333333333
```

Pgm-02-R/S

```
        3.      N

       A

        4.
        8.
        2.

        8.
        8.
        0.

        0.
        8.
        1.

      DET

       96.

      INV

 .0833333333
 .0833333333
-.1666666667

-.0833333333
 .0416666667
 .1666666667

 .6666666667
-.3333333333
-.3333333333
```

Program Listing:

```
000  76 LBL     039  02  02     078  00  00     117  04  4      156  69 OP
001  11  A      040  91 R/S     079  01  1      118  00  0      157  02  02
002  42 STO     041  43 RCL     080  04  4      119  02  2      158  69 OP
003  07  07     042  01  01     081  69 OP      120  42 STO     159  05  05
004  42 STO     043  97 DSZ     082  02  02     121  02  02     160  43 RCL
005  02  02     044  00  00     083  69 OP      122  43 RCL     161  07  07
006  42 STO     045  87 IFF     084  05  05     123  07  07     162  42 STO
007  00  00     046  98 ADV     085  98 ADV     124  42 STO     163  00  00
008  09  9      047  43 RCL     086  43 RCL     125  00  00     164  42 STO
009  69 OP      048  07  07     087  07  07     126  76 LBL     165  89  89
010  17  17     049  42 STO     088  42 STO     127  79  x̄      166  36 PGM
011  69 OP      050  00  00     089  00  00     128  43 RCL     167  02  02
012  00  00     051  43 RCL     090  01  1      129  02  02     168  17 B'
013  03  3      052  01  01     091  36 PGM     130  69 OP      169  36 PGM
014  01  1      053  97 DSZ     092  02  02     131  04  04     170  02  02
015  69 OP      054  02  02     093  14  D      132  73 RC*     171  18 C'
016  04  04     055  87 IFF     094  76 LBL     133  01  01     172  76 LBL
017  43 RCL     056  91 R/S     095  86 STF     134  69 OP      173  89  π
018  07  07     057  76 LBL     096  91 R/S     135  06  06     174  36 PGM
019  69 OP      058  12  B      097  36 PGM     136  69 OP      175  02  02
020  06  06     059  98 ADV     098  02  02     137  21  21     176  91 R/S
021  98 ADV     060  01  1      099  91 R/S     138  69 OP      177  97 DSZ
022  69 OP      061  06  6      100  43 RCL     139  22  22     178  00  00
023  00  00     062  01  1      101  01  01     140  97 DSZ     179  89  π
024  01  1      063  07  7      102  97 DSZ     141  00  00     180  43 RCL
025  03  3      064  03  3      103  00  00     142  79  x̄      181  07  07
026  69 OP      065  07  7      104  86 STF     143  98 ADV     182  42 STO
027  02  02     066  69 OP      105  98 ADV     144  98 ADV     183  00  00
028  69 OP      067  02  02     106  91 R/S     145  91 R/S     184  98 ADV
029  05  05     068  69 OP      107  76 LBL     146  76 LBL     185  97 DSZ
030  98 ADV     069  05  05     108  14  D      147  15  E      186  89  89
031  01  1      070  36 PGM     109  25 CLR     148  69 OP      187  89  π
032  36 PGM     071  02  02     110  36 PGM     149  00  00     188  98 ADV
033  02  02     072  13  C      111  02  02     150  02  2      189  98 ADV
034  12  B      073  98 ADV     112  15  E      151  04  4      190  91 R/S
035  76 LBL     074  91 R/S     113  36 PGM     152  03  3
036  87 IFF     075  76 LBL     114  02  02     153  01  1
037  91 R/S     076  13  C      115  16 A'      154  04  4
038  36 PGM     077  69 OP      116  04  4      155  02  2
```

LIST/TRACE UNDER PROGRAM CONTROL - Henri Verine of Marseille, France writes that he had become interested in the end of the first part of Peter Poloczek's Fast Grafik 3-D Plot program in V8N2. With experimentation he found that the effects observed when the program reached the end of the partition were a function of the last instruction in the partition:

* With LRN or R/S the program stops and the display does not flash.

* With a call to a user defined label (A, B, etc.) the program pointer searches for the label and the calculator runs from that point if the label is found.

* With a CP at the last location there is a flashing display but no clearing of the program.

* With a Cms at the last location there is a flashing display and a clearing of data registers.

* With a SBR at the last step you have several options. The calculator stops with a flashing display. If you press one of the user defined labels (A through E') the calculator goes to that location and continues. If you press one of the common labels (EE, sin, etc.,) the calculator goes to that location and continues. But if you press a valid three digit adddress the calculator goes to that address and stops.

Dr. Verine asked if this matter had already been investigated. My search found that a similar effect was used in the Hand-held Stop described in V8N1P6. In that case a programmed Op-08 was used to initiate the end of partition behaviour. The editor's note to that article referred the readers to two articles from 52 Notes. Both references are reproduced below. From V2N10P3:

"In the course of trying to find ways for a running program to determine printer connection, A. B. Winston examined some of the listing options both with and without the printer, when executed under program control. His results lead to the following observations: Contrary to the last statement on page VI-4 of Personal Programming, termination of INV-List executed under program control does not return control to the keyboard, and both program and label listing can be made under program control without relinquishing control to the keyboard upon completion. The latter are accomplished by having a List or Op-08 instruction at the beginning of a called subroutine which ends with an INV-SBR (RTN) as the last step in the current partition. For example, (with the printer connected) if program partitioning ends with step 479, the sequence

... SBR 476 ... 476: List Stflg 5 RTN

executes under program control beginning at the SBR call by listing steps 477-479, then resumes with the steps following the SBR 476 call. The sequence

...SBR 475... 475: Op 08 Stflg 5 RTN

List/Trace Under Program Control - (cont)

executes as a label search from step 477 to step 479, and returns to the calling program having effectively done nothing. In both cases, the SBR call without printer connection causes flag 5 to be set. Thus at a cost of only 8 steps, a flag can be automatically set when a program is running without printer connection, better than the 12 steps required by the HIR method. For this purpose, Op-08 is 'cleaner' than List. Then, as A. B. suggests, it only takes 8 steps to do

...SBR 475... 475: Prt Op 08 R/S RTN

which either prints and continues, or halts, depending upon printer connection. ... While program call-execution of the List and Op 08 functions may find occasional practical application apart from printer sensing, call-execution of INV List will probably find greater use: it's a cheap way to output tagged results, and do this without a forced halt."

And from V3N6P5 of 52 Notes:

"Jared Weinberger has found that the execution of a last-step RTN during a programmed List applies also to other last-step instructions. A sequence of the form:

Lbl-A-B-sequence1-RTN-Lbl-B-List-sequence2-f-sequence3-RTN

where f is an instruction located in the last step of the current partition, on a call to A lists sequence 2 and instruction f, and executes instruction f. If f is RTN, sequence 1 is executed following the listing of sequence 2 and f; if f is R/S, execution halts at the last step; if f is RST, execution continues at step 000; but if f is any other instruction an error condition is set and the program halts following its execution. Composite instructions beginning with f are completable manually from the keyboard. If f is C and routine C re-partitions memory such that sequence 3 is executable, a call to A lists sequence 2 C, executes C, then sequence 3, sequence 1. Members are invited to find new practical applications for any of this behaviour."

Other than Bill Skillman's use for automatic printer sensing the technique does not seem to have found further use. Again, club members are invited to investigate and report on applications.

ANOTHER BRAIN-TEASER - Myer Boland. The problem is to obtain the number 64 in the display in as few steps as possible, but without using any of the number keys. An example:

cos + CE) y^x (CE + CE + CE $\sqrt{x}$ =

This would seem to count as thirteen steps, but note that the routine assumes that the display contains a zero. Thus, in the general case, one must start the sequence above with a display register clearing instruction such as CLR, raising the number of steps to 14. Similarly, the routines which use the statistics functions such as Σ+ usually rely on the calculator being in the start-up condition. For this puzzle you are not allowed to make that assumption in your count of steps.

MORE TESTS OF NUMERICAL PRECISION - In V8N4P4 Bob Fruit proposed that a savings accumulation problem might be a good vehicle for comparing numerical precision. The specific problem he proposed was a dollar contributed to a savings account every month, with interest compounded every month for thirty years. The appropriate equation is

$$S_n = \frac{(1+i)^n - 1}{i}$$

An annual interest rate of ten per cent was assumed. The effective monthly interest rate is then 10%/12. Bob provided answers for various computers and calculators, but did not provide an exact solution. V8N5P26 provided exact solutions from Laurance Leeds, George Thomson, and Carl Rabe. Using those results we can make a more accurate assessment of the capability of various machnies For this evaluation we will not use some of the more exotic methods of calculation described in V8N4P4. Rather, we will use straightforward evaluation of the formula using the raise to a power function, and will restrict ourselves to the as delivered BASIC. Representative results include:

Exact Solution	2260.48792 47960 86067 64793 ...
Radio Shack Model III (S.P.)	2260.29
Apple II Plus	2260.48828 4
Commodore 64 or VIC-20	2260.48828 4
HP-11C or HP-41	2260.48764 1
Radio Shack Color Computer	2260.48801
TI-57	2260.48799 67
TI-55-II, TI-57-LCD	2260.48772 43
TI BA-55	2260.48782 43
TI MBA	2260.48790.9
TI CC-40	2260.48792 41984
TI-99/4A	2260.48792 43288
TI-58/59	2260.48792 4713
Radio Shack Model 100	2260.48792 47471
IBM PC (Double Precision)	2260.48792 47960 93

The above table clearly indicates the advantages that accrue to the TI calculators through the use of guard digits--note that every TI programmable yields better results than the HP calculators (HP-11, HP-41), and also better than the personal computers with the exception of those made by TI (the CC-40 and the 99/4A) or those which have double precision available (Model 100 and IBM PC). It always amazes me that TI hasn't been able to do more with this inherent advantage of their mechanizations.

A PGM-MM-R/S QUIRK - There are some unexpected quirks associated with the use of the Pgm-mm-R/S technique discussed elsewhere in this issue (see pages 4 and 7). The return address which is used by the Pgm-mm-R/S sequence is stored in a manner such that return to the point in the memory module at which the exit occurred does not even depend on use of the correct value for mm. The only requirements are that the value for mm not be 00, and that mm not exceed the highest library program in the module which is installed. As an example consider the program on page 9 of this issue. The 02 at locations 039, 098 or 175 could be replaced by any value from 01 through 25 and satisfactory performance will result. If a value greater than 25 is used an error will result since there are only 25 programs in the Master Library.

COLLATZ 3n + 1 Algorithm (W. J. Widmer, Box U-37, Univ. of Conn., Storrs, CT.)
In PPC Notes V6N9/10 P13 I gave a short program on an interesting unsolved conjecture which was questionably attributed to the Polish mathematician, Stanislaw Ulam. My search for the true author of this conjecture has been, itself, an interesting venture. When John Kennedy of Santa Monica could not retrieve his source for calling this "Ulam's Conjecture," I wrote directly to Dr. Stanislaw Ulam at the Los Alamos project. Ulam replied that "it's a beautiful conjecture; I wish it were mine."

Through Dr. Lynn Garner of the Mathematics Department at Brigham Young University in Provo, Utah, I learn, from a letter from Dr. Lothar Collatz (retired from mathematics at the University of Hamburg) that Collatz first posed this problem in personal communications at various international professional meetings in 1932. Because he was not able to solve this himself, he continued to pose this over the years; and in 1950, from discussions by Collatz at a meeting at Harvard University, there was a spurt of international interest when other mathematicians began also to write about it. It then became known as "The Syracuse Problem" and "The Kakutanis Problem"--but not "Ulam's Conjecture." Henry Mullish (Senior Research Scientist at the Courant Institute of Mathematical Sciences, New York University, and, I believe, head of NYU's Department of Computer Science) in both his 1976 "The Complete Pocket Calculator Handbook" (Collier Books, N.Y.) and, with Stephen Kochan, in "Programmable Pocket Calculators" (1980; Hayden Book Co.) writes that Ulam "not long ago published an interesting paper...", etc. Well, there was no such paper, and when I wrote to Dr. Mullish about this, he replied that "I have never seen it in print. I was told about it by a friend who very confidently spoke of it as Ulam's Conjecture." The 1982 book on "Mathematical Recreations for the Programmable Calculator" by Dr. Hoffman and L. Mohler (Hayden Book Co.) also refers to this as "Ulam's Problem"; I suspect that they might have seen it in Mullish & Kochan's book!

Perhaps PPC Notes is not the place to take up the cry for bona fide documentation in scientific reports. Certainly one would hope that "research scientists" would be more careful in this respect. Any PPC readers interested in the documentation of my search which finally laid the origin of the subject conjecture to Dr. Lothar Collatz can get this by sending me a 37¢ SASE.

Editor's Note: Professor Widmer's research provides an excellent example of being careful about attribution. If you submit material which is a review or an optimization of someone else's work you should identify the source. If you are not sure of the source, say so with your submission and I will search old issues of 52 Notes, TI PPC Notes, PPX Exchange, TISOFT, and the like for an appropriate reference.

MORE TI-59 AVAILABILITY - Member Don Lambert reports that his firm, Lambert Programming Services, has TI-59's and PC-100's available for sale. He also has a file of contacts who would like to sell used hardware. You may call him at (213)-658-MATH or write to:

Lambert Programming Service
Attn: Donald R. Lambert
434 North Crescent Heights Blvd.
Los Angeles CA 90048

OTHER SOLUTIONS FOR SYSTEMS OF LINEAR EQUATIONS

Several TI-59 programs have been published for solution of systems of linear equations without using the ML-02 library program. One example is the program by Douglas P. Anderson which appeared in the September/October 1980 issue of PPX Exchange. A distinguishing feature of the program is the use of the Math/Utilities module, and in particular the use of MU-07, to perform the matrix arithmetic needed to implement a row reduction solution. The program provides printout of the input parameters and of the solution, albeit in a somewhat cumbersome form. A sample printout for the problem on page 12 of the Master Library manual appears at the right. The solution appears as the last element of each row. Input data is accepted with very little immediate calculation, i.e., at a rate essentially limited by operator key-in speed. The solution for the problem illustrated requires about three minutes. The solution is limited to seventh order by storage limitations. There is one minor error in the instructions on V4N5P9 of PPX Exchange. The matrix elements are stored starting at R-07, not at R-08 as indicated by program note 4.

```
 1.       ROW

 4.
 8.
 0.
 4.

 2.       ROW

 8.
 8.
 8.
 4.

 3.       ROW

 2.
 0.
 1.
 6.



 1.       ROW

 1.
 0.
 1. -12
 4.

 2.       ROW

 0.
 1.
-1. -12
-1.5

 3.       ROW

 0.
 0.
 1.
-2.
```

V7N6P13 of TI PPC Notes contained a program by Sasa Nick and Dejan Ristanovic of Yugoslavia entitled "Gauss Reduction of Matrices M·N". Data entry for this program is very rapid. There is no printout of either the input data or of the solution. As indicated in the program description, a powerful program but not a very user-friendly one. The solution time for the problem from page 12 of the Master Library manual is about 1.5 minutes.

The programs based on ML-02 are limited to solutions where the order of the system is 8 or less, as is the Nick/Ristanovic program from V7N6P13. The Anderson program from PPX Exchange is limited to a 7x7 system. Both Dejan Ristanovic and Robert Prins called my attention to a program which is claimed to be able to solve for as many as sixteen unknowns. The original version of the program by Henrik Ohlsson appeared in the Utmanigen (Challenge) section the 81-2 issue of Programbiten. The program runs in fast mode, using the Pgm-02-SBR-240 method of initialization. The printout is well formatted. A sample for the problem from page 12 of the Master Library manual appears on the next page.

The limitations I have identified are (1) since calculations are performed as each element is entered the user is continually waiting on the calculator, and (2) the program seems to yield less accurate solutions than can be obtained with ML-02 based programs.

Other Solutions for Systems of Linear Equations (cont)

Where the order of the system is eight or less, an ML-02 based solution is recommended. If you require solutions for higher order systems with your TI-59 then the Ohlsson program is required, and you must accept the limitations.

To date I have not tested the capability of the program beyond tenth order. Readers are invited to report their experience with this outstanding program.

User Instructions for the Ohlsson Program:

1. Record the short initialization program at the right on a magnetic card and label it 1.a. Record the longer program listed below and label it 1.b. Record both programs using startup partitioning (6 Op 17).

2. Enter program 1.a into the calculator. Press A. A "0" will be displayed. This step initializes fast mode and prepares for entry of program 1.b under program control.

3. Enter program 1.b into the calculator. A "0" will be displayed.

4. Enter the order of the system (n) and press R/S. The order will be printed. " 1.01" will appear in the display indicating the program is ready to accept matrix element A_{11} .

5. Enter the first element of the matrix into the display and press R/S. The calculator will run for a short perid of time and stop with "1.02" in the display indicating the program is ready to accept matrix element A_{12} . Continue to enter the matrix elements by row in response to the prompting. When the column portion of the prompt becomes "i.n+1" enter vector element Bi and press R/S. The calculator will run for a period of time and stop with a prompt in the display asking for the first matrix element of the next row. Continue entry of the matrix and vector elements. When the last vector element has been entered the calculator will run for a period of time, print the solution, and stop with a "0" in the display. To start a new problem simply enter the new order (n), press R/S, and see the prompt for the first matrix element of the new problem.

6. To run without a printer change steps 228 through 235 of the program to

Pause-R/S-Op-26-Dsz-4-02-26 .

1.a

```
000  00   0
001  00   0
002  00   0
003  76  LBL
004  11   A
005  36  PGM
006  02   02
007  71  SBR
008  02   02
009  40   40
010  09   9
011  00   0
012  25  CLR
013  91  R/S
014  83  GO*
015  00   00
```

```
    3.

    4.
    8.
    0.
    4.

    8.
    8.
    8.
    4.

    2.
    0.
    1.
    6.

    4.
 -1.5
   -2.
```

Other Solutions for Systems of Linear Equations - (cont)

Listing for Part 1.b of the Ohlsson Program:

```
000  98 ADV   040  79  79   080  00  0    120  72 ST*   160  08  08   200  00  0
001  58 FIX   041  93  .    081  01  1    121  03  03   161  22 INV   201  63 EX*
002  09  09   042  00  0    082  44 SUM   122  69 OP    162  44 SUM   202  06  06
003  47 CMS   043  01  1    083  04  04   123  23  23   163  03  03   203  72 ST*
004  99 PRT   044  44 SUM   084  43 RCL   124  97 DSZ   164  32 X:T   204  03  03
005  42 STO   045  04  04   085  04  04   125  02  02   165  65  x    205  69 OP
006  01  01   046  43 RCL   086  66 PAU   126  00  00   166  32 X:T   206  23  23
007  42 STO   047  04  04   087  91 R/S   127  97  97   167  73 RC*   207  69 OP
008  07  07   048  66 PAU   088  99 PRT   128  43 RCL   168  06  06   208  26  26
009  69 OP    049  91 R/S   089  85  +    129  04  04   169  95  =    209  97 DSZ
010  27  27   050  99 PRT   090  73 RC*   130  59 INT   170  72 ST*   210  02  02
011  09  9    051  94 +/-   091  03  03   131  42 STO   171  03  03   211  02  02
012  42 STO   052  65  x    092  95  =    132  05  05   172  43 RCL   212  00  00
013  03  03   053  32 X:T   093  42 STO   133  22 INV   173  08  08   213  69 OP
014  69 OP    054  73 RC*   094  05  05   134  97 DSZ   174  44 SUM   214  37  37
015  17  17   055  06  06   095  69 OP    135  05  05   175  03  03   215  97 DSZ
016  98 ADV   056  95  =    096  32  32   136  02  02   176  69 OP    216  01  01
017  98 ADV   057  74 SM*   097  93  .    137  13  13   177  26  26   217  00  00
018  25 CLR   058  03  03   098  00  0    138  08  8    178  97 DSZ   218  17  17
019  42 STO   059  69 OP    099  01  1    139  42 STO   179  02  02   219  98 ADV
020  08  08   060  23  23   100  44 SUM   140  03  03   180  01  01   220  98 ADV
021  43 RCL   061  69 OP    101  04  04   141  43 RCL   181  53  53   221  09  9
022  07  07   062  26  26   102  43 RCL   142  01  01   182  43 RCL   222  42 STO
023  42 STO   063  32 X:T   103  04  04   143  42 STO   183  01  01   223  06  06
024  02  02   064  97 DSZ   104  66 PAU   144  02  02   184  22 INV   224  69 OP
025  09  9    065  02  02   105  91 R/S   145  69 OP    185  44 SUM   225  34  34
026  42 STO   066  00  00   106  99 PRT   146  23  23   186  06  06   226  73 RC*
027  06  06   067  52  52   107  69 OP    147  69 OP    187  42 STO   227  06  06
028  43 RCL   068  43 RCL   108  23  23   148  28  28   188  02  02   228  99 PRT
029  04  04   069  07  07   109  85  +    149  00  0    189  97 DSZ   229  69 OP
030  59 INT   070  42 STO   110  00  0    150  63 EX*   190  05  05   230  26  26
031  42 STO   071  02  02   111  63 EX*   151  03  03   191  01  01   231  97 DSZ
032  04  04   072  22 INV   112  03  03   152  32 X:T   192  45  45   232  04  04
033  42 STO   073  44 SUM   113  95  =    153  69 OP    193  43 RCL   233  02  02
034  05  05   074  03  03   114  69 OP    154  23  23   194  08  08   234  26  26
035  69 OP    075  97 DSZ   115  33  33   155  00  0    195  22 INV   235  98 ADV
036  24  24   076  05  05   116  55  ÷    156  63 EX*   196  44 SUM   236  25 CLR
037  29 CP    077  00  00   117  43 RCL   157  03  03   197  03  03   237  91 R/S
038  67  EQ   078  41  41   118  05  05   158  75  -    198  69 OP    238  83 GO*
039  00  00   079  93  .    119  95  =    159  43 RCL   199  23  23   239  04  04
```

Dejan Ristanovic and Robert Prins submitted revisions to the Ohlsson program to use the Stflg-Ind method for fast mode entry and compress the program storage requirement to one card side. The Prins program does not provide any printout. The Ristanovic program does provide a printout, but the prompts rather than the input values are printed--not an improvement relative to the original Ohlsson program. A quick look at the Ristanovic program suggests that it may be difficult to reprogram to provide printing of the input data. If that cannot be accomplished then I would recommend that users stay with the original version. In programs such as these, where there is no storage of the input data such that it can be retrieved or corrected later, it is important to have a printed record if a printer is available. Since calculations are performed as each element is entered there is no capability to edit the input as was possible with the ML-02 deriviatives. Therefore, it is important to be very careful during data entry. Listings for the Prins and Ristanovic programs appear on the next page. The user instructions for both programs are similar.

1. Record the Ristanovic program 9-Op-17. You may record the Prins program with turn-on partitioning since locations 225-228 change the partitioning to 9-Op-17.

2. Enter the order of the system (n) and press A. The Prins program will show a flashing "10". The Ristanovic program will show a flashing "2.12". Press 7 and then EE. See a prompt of "1.01". Proceed as with the Ohlsson program. The Ristanovic program will print the solution. The Prins program will display the first element of the solution; press R/S to display the remaining elements of the solution.

Other Solutions for Linear Equations - (cont)

Listing for the Ristanovic Program:

```
000 91 R/S   040 65  ×    080 42 STO   120 97 DSZ   160 44 SUM   200 37  37
001 25 CLR   041 32 X:T   081 05  05   121 05  05   161 03  03   201 97 DSZ
002 69 OP    042 73 RC*   082 69 OP    122 01  01   162 69 OP    202 01  01
003 27  27   043 06  06   083 32  32   123 99  99   163 26  26   203 00  00
004 09  9    044 95  =    084 93  .    124 08  8    164 97 DSZ   204 07  07
005 42 STO   045 74 SM*   085 00  0    125 42 STO   165 02  02   205 09  9
006 03  03   046 03  03   086 01  1    126 03  03   166 01  01   206 42 STO
007 25 CLR   047 69 OP    087 44 SUM   127 43 RCL   167 39  39   207 06  06
008 42 STO   048 23  23   088 04  04   128 01  01   168 43 RCL   208 69 OP
009 08  08   049 69 OP    089 43 RCL   129 42 STO   169 01  01   209 34  34
010 43 RCL   050 26  26   090 04  04   130 02  02   170 22 INV   210 73 RC*
011 07  07   051 32 X:T   091 99 PRT   131 69 OP    171 44 SUM   211 06  06
012 42 STO   052 97 DSZ   092 91 R/S   132 23  23   172 06  06   212 99 PRT
013 02  02   053 02  02   093 69 OP    133 69 OP    173 42 STO   213 91 R/S
014 09  9    054 00  00   094 23  23   134 28  28   174 02  02   214 69 OP
015 42 STO   055 40  40   095 85  +    135 00  0    175 97 DSZ   215 26  26
016 06  06   056 43 RCL   096 00  0    136 63 EX*   176 05  05   216 97 DSZ
017 43 RCL   057 07  07   097 63 EX*   137 03  03   177 01  01   217 04  04
018 04  04   058 42 STO   098 03  03   138 32 X:T   178 31  31   218 02  02
019 59 INT   059 02  02   099 95  =    139 69 OP    179 43 RCL   219 10  10
020 42 STO   060 22 INV   100 69 OP    140 23  23   180 08  08   220 25 CLR
021 04  04   061 44 SUM   101 33  33   141 00  0    181 22 INV   221 81 RST
022 42 STO   062 03  03   102 55  ÷    142 63 EX*   182 44 SUM   222 76 LBL
023 05  05   063 97 DSZ   103 43 RCL   143 03  03   183 03  03   223 11  A
024 69 OP    064 05  05   104 05  05   144 75  -    184 69 OP    224 47 CMS
025 24  24   065 00  00   105 95  =    145 43 RCL   185 23  23   225 42 STO
026 29 CP    066 30  30   106 72 ST*   146 08  08   186 00  0    226 01  01
027 67  EQ   067 93  .    107 03  03   147 22 INV   187 63 EX*   227 42 STO
028 00  00   068 00  0    108 69 OP    148 44 SUM   188 06  06   228 07  07
029 67  67   069 01  1    109 23  23   149 03  03   189 72 ST*   229 58 FIX
030 93  .    070 44 SUM   110 97 DSZ   150 32 X:T   190 03  03   230 00  00
031 00  0    071 04  04   111 02  02   151 65  ×    191 69 OP    231 02  2
032 01  1    072 43 RCL   112 00  00   152 32 X:T   192 23  23   232 85  +
033 44 SUM   073 04  04   113 84  84   153 73 RC*   193 69 OP    233 02  2
034 04  04   074 99 PRT   114 43 RCL   154 06  06   194 26  26   234 52 EE
035 43 RCL   075 91 R/S   115 04  04   155 95  =    195 97 DSZ   235 01  1
036 04  04   076 85  +    116 59 INT   156 72 ST*   196 02  02   236 02  2
037 99 PRT   077 73 RC*   117 42 STO   157 03  03   197 01  01   237 95  =
038 91 R/S   078 03  03   118 05  05   158 43 RCL   198 86  86   238 86 STF
039 94 +/-   079 95  =    119 22 INV   159 08  08   199 69 OP    239 40 IND
```

Listing for the Prins Program

```
000 92 RTN   040 73 RC*   080 22  22   120 01  01   160 03  03   200 06  06
001 24 CE    041 06  06   081 93  .    121 96  96   161 69 OP    201 97 DSZ
002 69 OP    042 54  )    082 00  0    122 08  8    162 26  26   202 01  01
003 37  37   043 74 SM*   083 01  1    123 42 STO   163 97 DSZ   203 00  00
004 09  9    044 03  03   084 44 SUM   124 03  03   164 08  08   204 09  09
005 42 STO   045 69 OP    085 04  04   125 43 RCL   165 01  01   205 69 OP
006 03  03   046 23  23   086 43 RCL   126 01  01   166 37  37   206 34  34
007 42 STO   047 69 OP    087 04  04   127 42 STO   167 43 RCL   207 73 RC*
008 06  06   048 26  26   088 66 PAU   128 08  08   168 01  01   208 06  06
009 43 RCL   049 32 X:T   089 91 R/S   129 69 OP    169 44 SUM   209 66 PAU
010 07  07   050 97 DSZ   090 72 ST*   130 23  23   170 06  06   210 91 R/S
011 42 STO   051 02  02   091 03  03   131 69 OP    171 42 STO   211 69 OP
012 02  02   052 00  00   092 69 OP    132 32  32   172 08  08   212 26  26
013 43 RCL   053 36  36   093 23  23   133 00  0    173 97 DSZ   213 97 DSZ
014 04  04   054 43 RCL   094 00  0    134 63 EX*   174 05  05   214 04  04
015 59 INT   055 07  07   095 63 EX*   135 03  03   175 01  01   215 02  02
016 42 STO   056 42 STO   096 03  03   136 32 X:T   176 29  29   216 07  07
017 04  04   057 02  02   097 69 OP    137 69 OP    177 43 RCL   217 00  0
018 42 STO   058 44 SUM   098 33  33   138 23  23   178 02  02   218 81 RST
019 05  05   059 03  03   099 74 SM*   139 53  (    179 44 SUM   219 76 LBL
020 69 OP    060 97 DSZ   100 03  03   140 00  0    180 03  03   220 11  A
021 24  24   061 05  05   101 43 RCL   141 63 EX*   181 69 OP    221 22 INV
022 29 CP    062 00  00   102 05  05   142 03  03   182 23  23   222 58 FIX
023 67  EQ   063 26  26   103 22 INV   143 75  -    183 00  0    223 94 +/-
024 00  00   064 93  .    104 64 PD*   144 43 RCL   184 63 EX*   224 32 X:T
025 64  64   065 00  0    105 03  03   145 02  02   185 06  06   225 09  9
026 93  .    066 01  1    106 69 OP    146 44 SUM   186 69 OP    226 69 OP
027 00  0    067 44 SUM   107 23  23   147 03  03   187 26  26   227 17  17
028 01  1    068 04  04   108 97 DSZ   148 32 X:T   188 72 ST*   228 47 CMS
029 44 SUM   069 43 RCL   109 02  02   149 65  ×    189 03  03   229 32 X:T
030 04  04   070 04  04   110 00  00   150 32 X:T   190 69 OP    230 42 STO
031 43 RCL   071 66 PAU   111 81  81   151 73 RC*   191 23  23   231 01  01
032 04  04   072 91 R/S   112 43 RCL   152 06  06   192 97 DSZ   232 42 STO
033 66 PAU   073 42 STO   113 04  04   153 54  )    193 08  08   233 07  07
034 91 R/S   074 05  05   114 59 INT   154 72 ST*   194 01  01   234 93  .
035 94 +/-   075 73 RC*   115 42 STO   155 03  03   195 83  83   235 01  1
036 53  (    076 03  03   116 05  05   156 43 RCL   196 69 OP    236 34 ΓX
037 46 INS   077 44 SUM   117 22 INV   157 02  02   197 27  27   237 33 X²
038 65  ×    078 05  05   118 97 DSZ   158 22 INV   198 09  9    238 35 1/X
039 32 X:T   079 69 OP    119 05  05   159 44 SUM   199 42 STO   239 86 STF
```

ACCURACY OF THE SOLUTIONS FOR SYSTEMS OF LINEAR EQUATIONS

Several different programs for solution of systems of linear equations with the TI-59 have been discussed in this issue. How does the user decide which program to use? The discussion in previous pages of this issue has addressed considerations such as user friendliness, system size, and the like. Another important issue is accuracy of the solution, and we will see that the Ohlsson program and its derivatives are less accurate. How do we measure accuracy? George Thomson provided some thoughts on that subject.

Here are some practical tips for testers of matrix inversion programs. The workhorse test matrices are the "Hilberts"; the first row is 1, 1/2, 1/3, ..., the second row is 1/2, 1/3, 1/4, ..., the third row is 1/3, 1/4, 1/5, ..., and so on. Their inverses have horrendously huge integers and are available. See for example, I. R. Savage and E. Lukacs, National Bureau of Standards AMS No. 39, pp. 107-108 (1954) for the inverses up to 10 x 10. The seventh row, seventh column of the 10 x 10 inverse is 348 06739 96800 . Others are almost as large. The "sub-Hilberts" with the first row 1/2, 1/3, 1/4, ..., the second row 1/3, 1/4, 1/5, ..., and so on are even harder to invert correctly. I suggest as a guinea pig the 7 x 7 sub-Hilbert, with ones on the right hand side:

1/2	1/3	1/4	1/5	1/6	1/7	1/8	1
1/3	1/4	1/5	1/6	1/7	1/8	1/9	1
1/4	1/5	1/6	1/7	1/8	1/9	1/10	1
1/5	1/6	1/7	1/8	1/9	1/10	1/11	1
1/6	1/7	1/8	1/9	1/10	1/11	1/12	1
1/7	1/8	1/9	1/10	1/11	1/12	1/13	1
1/8	1/9	1/10	1/11	1/12	1/13	1/14	1

The exact solution of the simultaneous equations is 56, -1512, 12600, -46200, 83160, -72072, and 24024. All the elements of the inverse are integers, the largest is 6915 58560. The most practical measure of the accuracy of a solution is to calculate the relative error, i.e., (answer - true result)/(true result) for each element and take the largest value. This measure is related to the number of meaningful significant digits in the results.

Readers who are familiar with 52 Notes will recall that V2N12P5 described the use of the Hilbert matrices ($A_{ij} = 1/(i+j-1)$ as a test of the ability of a matrix inversion routine to handle ill-conditioned matrices.

All the ML-02 deriviatives yield identical results. Therefore, description of the results from any one of the ML-02 programs defines the accuracy of all of them. Similarly, the Ohlsson program and the derivatives by Prins and Ristanovic yield identical results, and a single description of results will suffice for all three. For the 7 x 7 sub-Hilbert test suggested by George Thomson the various algorithms yield the following results:

Accuracy of the Solutions for Systems of Linear Equations - (cont)

Ohlsson/ Ristanovic/ Prins	TI-59 ML-02	Anderson Row Reduction	Nick and Ristanovic "Gauss"	CC-40 Mathematics Module
Programbiten		PPX V4N5P8	V7N6P13	V8N5P14
55.9233	56.0082	56.0081	56.0076	56.000032
-1510.2276	-1512.1896	-1512.1865	-1512.1732	-1512.000787
12587.0911	12601.3863	12601.3511	12601.2536	12600.0059
-46157.9673	-46204.5344	-46204.3822	-46204.0623	-46200.0192
83091.9632	83167.3718	83167.0718	83166.5503	83160.0311
-72018.4333	-72077.8274	-72077.5542	-72077.1412	-72072.0246
24007.6425	24025.7860	24025.6926	24025.5659	24024.0074
1.37E-3	1.46E-4	1.45E-4	1.35E-4	5.71E-7

The ML-02 solution, the Anderson row reduction solution, and the Nick/Ristanovic solution yield nearly identical results from an accuracy standpoint. The Ohlsson program and its derivatives yield a solution that is an order of magnitude less accurate. The CC-40 yields a much more accurate solution than any of the TI-59 programs. This is somewhat surprising since the manual for the CC-40 Mathematics Module indicates that the method of solution is the same as for ML-02, and the CC-40 carries only one additional digit. To attain that level of accuracy with the CC-40 it is necessary to calculate the matrix elements in the program. If one tries to enter the values from the keyboard then the quirk described in V8N3P5 takes over, and only ten digits are used. The error in the resulting solution is 6.94E-3. One can obtain similar errors with ML-02 by pressing EE-INV-EE after calculating each reciprocal, and before entering the element for use by the program.

As an additional comparison of the capability of the CC-40 I entered an old "workhorse" simultaneous equation solution into the CC-40 and several other home/personal computers. Gene Friel also provided a solution using the Math-Pac Application Module with the HP-41C which uses a Gauss elimination method. The results, again using George Thomson's 7 x 7 test were:

HP-41	Color Comp	Apple II+	CC-40	Model 100
56.6667	55.5926	56.1869	56.000198	55.999816
-1527.3832	-1502.465	-1516.2347	-1512.00461	-1511.99596
12712.2414	12529.8262	12630.3122	12600.0337	12599.9716
-46566.4960	-45969.5924	-46297.3343	-46200.1101	-46199.9102
83755.0102	82784.5266	83315.8117	83160.1785	83159.8577
-72541.8140	-71774.7464	-72193.5851	-72072.1406	-72071.8899
24167.8491	23932.811	24060.8602	24024.0429	24023.9669
1.19E-2	7.27E-3	3.34E-3	3.53E-6	3.28E-06

The superiority of the CC-40 and Radio Shack Model 100, both 14 decimal digit computers, is obvious. But this solution on the CC-40 is an order of magnitude less accurate than that from the program in the Mathematics module.

Accuracy of the Solutions for Systems of Linear Equations - (cont)

For reference the common program used to evaluate the four computers is:

```
100 DIM A(10,10),B(10)
110 INPUT "Enter order";N
120 N = N-1
130 K=0
135 FOR I = 0 TO N
140 FOR J = 0 TO N
145 A(I,J)=1/(J+K+2)
150 NEXT J
155 B(I)=1
160 K=K+1
165 NEXT I
200 FOR K = 0 TO N
210 P = A(K,K)
250 FOR J = K TO N
260 A(K,J) = A(K,J)/P
270 NEXT J
280 B(K) = B(K)/P
290 FOR I = 0 TO N
300 IF I = K THEN 360
310 F = A(I,K)
320 FOR J = K TO N
330 A(I,J) = A(I,J) - F*A(K,J)
340 NEXT J
350 B(I) = B(I) - F*B(K)
360 NEXT I
370 NEXT K
490 FOR I = 0 TO N
500 PRINT "X"+STR$(I)+" = "; B(I)
510 NEXT I
600 END
```

Lines 130 through 165 provide automatic entry of the appropriate sub-Hilbert problem as defined by George Thomson on page 18. If you wish to use the program for other solutions simply replace those steps with appropriate steps to accept the appropriate matrix elements.

TI-66 STATUS - Although there are advertisements for the TI-66 in many catalogs I have yet to see a device for sale in the Tampa Bay area. Dave Leising has received a TI-66. He reports that decimal code 71 shows as the mnemonic * IN . In a program code 71 seems to act as an inverse. When followed by a R/S the code leaves the keyboard as if 2nd were pressed. Dave will report in more detail in the next issue. He does not have a PC-200, but reports that Maurice Swinnen has found the printer for sale in the Washington area.

STATUS OF THE CC-40 AND PERIPHERALS -There is frequent mention of the CC-40 in catalogs. In the Tampa Bay area the only peripherals which are available are the cartridges for Mathematics, Statistics, and Finance. Maurice Swinnen has obtained an Electrical Engineering cartridge. He has promised a review article next year.

SORTING ON THE CC-40 - The Statistics cartridge for the CC-40 has a shell sort subprogram. The program requires that the elements to be sorted have already been assembled into a one dimensional array. The following program provides entry of data into an array, sorting, and display of the sorted elements:

```
100 DIM X(100)
110 INPUT "Number of Elements? ";K
120 FOR I = 1 TO K
130 INPUT "Enter X("&STR$(I)&"): ";X(I)
140 NEXT I: PRINT "Press <ENTER> to Sort":PAUSE
150 PRINT "Sorting"
160 CALL SORT(X(),K)
170 FOR I = 1 TO K
180 PRINT "SX("&STR$(I)&") = ";X(I)
190 PAUSE: NEXT I
200 END
```

This program is much faster than the sorting program in the Math/Utilities module for the TI-59 (MU-06). The CC-40 sorts 60 random numbers in 31 seconds. The TI-59 takes 4 minutes 55 seconds.

MORE SUBPROGRAMS FOR THE CC-40 STATISTICS CARTRIDGE - Experiments show that the CC-40 Statistics cartridge has a subprogram for input and edit of a two-dimensional array which is very similar to that in the Mathematics cartridge. Even the call MI is the same. The prompts are the same as those described on V8N5P15 except that at the end of an edit of all input the Statistics cartridge implementation leaves the subprogram, while the Mathematics cartridge implementation returns for additional editing.

There are obviously other unlisted subprograms in the Statistics cartridge. A call for an AK subprogram for input and entry of a one-dimensional array as in the Mathematics cartridge yields the error message "Program not found". A call for an AU subprogram for input and edit of two one-dimensional arrays as with the Mathematics cartridge yields the error message "Illegal Syntax", which suggests there is a subprogram in the Statistics cartridge with the AU name.

A NOTE ON THE STF IND FAST MODE ENTRY - Robert Prins. Did you notice that the fast mode entry for Robert's program on page 17 is accomplished with a STF at location 239? There is no IND. Robert writes:

> "...I found that the program counted 241 steps. Just before I decided to delete one left parenthesis I thought along this line. If in a program Stf h12 functions like Stf Ind h12, why wouldn't it function at the end of the partitioning (it doesn't work from the keyboard however!) I tried it and it functioned."

ENGLISH FLAG - Reginald Van Genechten. The "Stars and Stripes" program by Richard Snow was the first in a series of programs which used graphics mode to print a national flag (V6N4/5P8). V7N1/2P22 reported that members of Programbiten had written programs to print the flags of Sweden, Norway, Denmark, Finland and Iceland. The printouts for all five flags were presented. The program to print the Swedish flag was provided. Now Reginald Van Genechten of Belgium has provided a program which will print the English flag. I assume that he chose to print an English flag since the Belgian flag with three equal width vertical bars would not be much of a challenge, and would be difficult to distinguish from other flags with similar design.

Enter the program and the data registers as listed on the opposite page. Where more than ten digits are required you must synthesize the value. For example, the value in data register 05 can be formed with the keyboard sequence 1110051 + .510011 = . When entering the program steps be sure to use 2nd-CLR (code 20) for step 160. After the graphics initialization is completed the code 20 becomes the second part of an Op-20 command. Record banks 1, 3, and 4 using the turn-on partitioning.

Operating Instructions:

1. Load banks 1, 3, and 4 into the calculator.

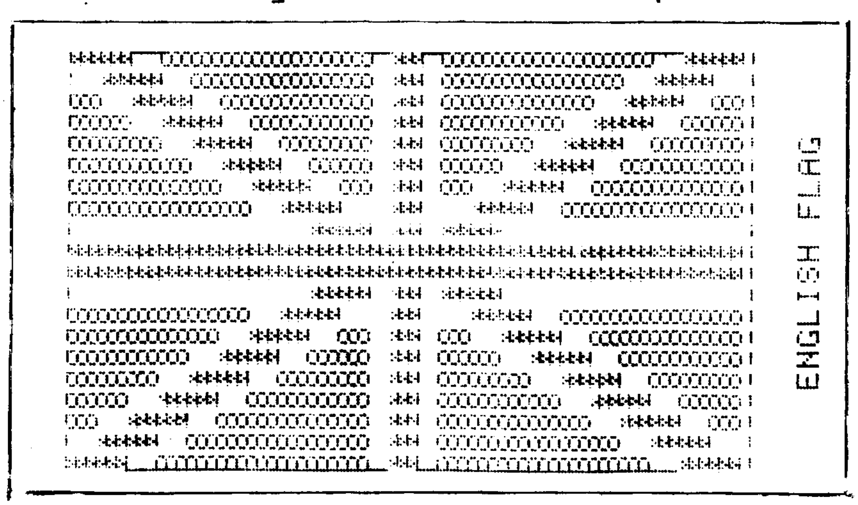
2. Initialize for graphics mode with the following keyboard sequence:

10 Op 17	See 159.99 in the display.
GTO 024 CLR	See 0 in the display.
Pgm 19 SBR 045	See flashing 0. in the display. Do not clear.
P/R	See flashing 0. in the display. Do not clear.
LRN	See 024 55 in the display.
Ins	See 024 55 in the display.
LRN RST CLR	See 0 in the display.
6 Op 17	See 479.59 in the display.

Note: The initialization process causes the program steps in locations 024 through 033 to be changed and the commands which had been in locations 033 through 158 to be pushed down one step. The Nop which had been in location 159 is deleted.

2. To start the printout press SBR 033. At the completion of the flag the calculator will stop with a 0 in the display.

3. Press R/S to see the legend "ENGLISH FLAG" printed.



English Flag - (cont)

Program Listing:

```
000  92 RTN     040  71 SBR     080  17 B'      120  04  04     160  25 CLR     200  32 X:T
001  43 RCL     041  01  01     081  71 SBR     121  17 B'      161  69 OP      201  85  +
002  56  56     042  73  73     082  00  00     122  17 B'      162  02  02     202  01  1
003  69 OP      043  71 SBR     083  01  01     123  17 B'      163  73 RC*     203  95  =
004  01  01     044  01  01     084  71 SBR     124  92 RTN     164  00  00     204  69 OP
005  05  5      045  83  83     085  01  01     125  43 RCL     165  69 OP      205  04  04
006  01  1      046  71 SBR     086  83  83     126  57  57     166  20  20     206  17 B'
007  69 OP      047  01  01     087  71 SBR     127  71 SBR     167  69 OP      207  17 B'
008  02  02     048  10  10     088  01  01     128  01  01     168  03  03     208  17 B'
009  43 RCL     049  71 SBR     089  57  57     129  73  73     169  17 B'      209  92 RTN
010  55  55     050  01  01     090  71 SBR     130  98 ADV     170  17 B'      210  32 X:T
011  69 OP      051  10  10     091  01  01     131  01  1      171  17 B'      211  42 STO
012  03  03     052  71 SBR     092  57  57     132  07  7      172  92 RTN     212  01  01
013  02  2      053  01  01     093  04  4      133  69 OP      173  69 OP      213  73 RC*
014  05  5      054  10  10     094  42 STO     134  01  01     174  01  01     214  00  00
015  69 OP      055  04  4      095  02  02     135  43 RCL     175  69 OP      215  69 OP
016  04  04     056  42 STO     096  71 SBR     136  58  58     176  02  02     216  20  20
017  17 B'      057  02  02     097  01  01     137  69 OP      177  69 OP      217  75  -
018  76 LBL     058  71 SBR     098  83  83     138  02  02     178  03  03     218  59 INT
019  17 B'      059  01  01     099  97 DSZ     139  43 RCL     179  69 OP      219  44 SUM
020  25 CLR     060  83  83     100  02  02     140  59  59     180  04  04     220  01  01
021  69 OP      061  97 DSZ     101  00  00     141  69 OP      181  17 B'      221  95  =
022  05  05     062  02  02     102  97  97     142  03  03     182  92 RTN     222  65  X
023  68 NOP     063  00  00     103  71 SBR     143  02  2      183  00  0      223  01  1
024  74 SM*     064  59  59     104  01  01     144  02  2      184  32 X:T     224  00  0
025  80  80     065  71 SBR     105  10  10     145  52 EE      185  71 SBR     225  22 INV
026  00  0      066  01  01     106  71 SBR     146  08  8      186  02  02     226  28 LOG
027  00  0      067  57  57     107  01  01     147  22 INV     187  10  10     227  22 INV
028  00  0      068  71 SBR     108  10  10     148  52 EE      188  69 OP      228  52 EE
029  00  0      069  01  01     109  73 RC*     149  69 OP      189  01  01     229  95  =
030  00  0      070  57  57     110  00  00     150  04  04     190  71 SBR     230  32 X:T
031  00  0      071  71 SBR     111  69 OP      151  69 OP      191  02  02     231  43 RCL
032  00  0      072  00  00     112  20  20     152  05  05     192  10  10     232  01  01
033  76 LBL     073  01  01     113  69 OP      153  98 ADV     193  69 OP      233  92 RTN
034  11  A      074  43 RCL     114  01  01     154  98 ADV     194  02  02
035  04  4      075  54  54     115  73 RC*     155  92 RTN     195  71 SBR
036  42 STO     076  71 SBR     116  00  00     156  73 RC*     196  02  02
037  00  00     077  01  01     117  69 OP      157  00  00     197  10  10
038  43 RCL     078  73  73     118  20  20     158  69 OP      198  69 OP
039  57  57     079  17 B'      119  69 OP      159  68  68     199  03  03
```

Data Register Contents:

0.	00	1000051.510000	20	110051.5100110	40
0.	01	5151.001111111	21	51.51001111100	41
0.	02	1411111111.005	22	1411005151.001	42
0.	03	1510051.510051	23	1110051.510011	43
5100111111.111	04	5100.111111111	24	1100.515100111	44
1110051.510011	05	1100515151.000	25	1400515100.111	45
1111.111111005	06	5151510011.000	26	1110051.510011	46
5151001111.000	07	1111005151.000	27	1111.005151001	47
1111005151.000	08	5151001111.000	28	2751510011.000	48
2751510011.000	09	1411111111.111	29	1100515125.000	49
1100515125.000	10	1005151.515100	30	5151001111.000	50
1400515100.000	11	1111.111111111	31	1111005151.000	51
51510011.00000	12	1100515151.000	32	5100111111.000	52
1411005151.001	13	5151510011.000	33	1111110051.000	53
1110051.510011	14	51510051.00000	34	5151515151.000	54
1100.515100111	15	5100515100.000	35	5100000000.000	55
1411110051.510	16	1411111100.515	36	2700000000.000	56
110051.5100110	17	1000051.510000	37	2020202020.000	57
51.51001111100	18	5151.001111111	38	3122272436.000	58
1411111100.515	19	1411110051.510	39	2300212713.000	59

CCL-144 CLEANING STRIPS - V8N5P11 described the CCL-144 cleaning strips which had been used to remedy a card reader problem in the editor's TI-59. Sample single strips were offered to members for two dollars each. Other members report success in using the strips:

George Thomson: "...It really seems to work for me."

Carl Raber "...thank you for promptly sending me the head cleaning strip. It worked like a charm. Absolutely NO read/write flaws since using it! Wish I'd had it a long time ago. ..."

The two dollar offer is still available for single strips from the club. The manufacturer, CMPI, Inc., 7200 Jersey Avenue North, Minneapolis, Minnesota 55428 offers a 15 per cent discount to club members on the twelve dollar retail price of a box of ten strips. If you send a check, or charge to VISA or Mastercard they will pay shipping. Otherwise they will ship via U.P.S., C.O.D. Mention TI PPC Notes when you order.

AN ALPHA CODE CHALLENGE - This problem is from the Utmaningen (Challenge) column of the Swedish newsletter Programbiten. Many print code converters have been published in TI PPC Notes. For examples see V5N1P2, V5N3P15, V5N6P10, etc. The object of all of these routines was to change a number into the equivalent print code. Such routines are an important ingredient in many programs; examples are the thirteen digit printers. Another frequently encountered printing problem is to change a number into a letter of the alphabet, say such that a "1" turns into an "A", a "2" into a "B", and so on up through a "26" into a "Z".

PROGRAMS NEEDED - Member A. E. Mackenzie would like to obtain copies of the General Annuities (198011D) and Annuities with Continuous Compounding" (198013D) programs. Can anyone help?

TABLE OF CONTENTS

Thats all for 1983. Happy Holidays Palmer Hanson