TI PPC NOTES

NEWSLETTER OF THE TI PERSONAL PROGRAMMABLE CALCULATOR CLUB

P. O. Box 1421, Largo, FL 34649

Volume 14, Number 3 Third Quarter 1990/1991

This issue has fewer pages than usual. I had to decide whether to delay the issue until after the holidays to permit publication of the normal number of pages or to publish a smaller issue and make up the missing pages in a later issue. I decided on the latter.

Correction of errors in past issues is the dominant theme of this issue. The case of the erroneous algorithm in the best fit circle program is a classic case of an error mindlessly carried forward. The error has been carried forward through three editions of Kolb's book, through Swinnen's book and into our newsletter. Perhaps that is just because the program hasn't received a lot of use. The history of the quadratic solutions for the TI-59 is another matter. The programs with problems were mostly concerned with using programming tricks to save memory and seem to have neglected the possibility of degraded accuracy in the results.

There is a sizeable amount of TI-81 material in this issue, but no graphs. Some members have questioned why I would cover a graphing calculator but include so little on graphing. The really unique feature of the TI-81 is not the graphing capability, but rather the powerful analytical tools (matrix manipulation, sorting, etc.) with a graphing capability and at a reasonable price. We covered the graphing functions quite thoroughly with earlier fx-7000G material. Most of that material is easily translatable for the TI-81. For newcomers we do offer a compilation of all the fx-7000G material published to date for four dollars.

TABLE OF CONTENTS

Happy Holidays!

ERRATA

References on TI-59 Test and Repair - V14N1P7 - Robert Lucas notes that the third reference under "Batteries" should be "V7N6P10 - Internal shorts in NiCad batteries and how to clear them" not V7N6P11 as originally shown.

Intersection of two circles - V14N1P16 - Robert Lucas notes that R should not be squared in the second term of the equation for B near the bottom of the page. The correct equation is

$$B = 2H_1H_3K_3 - 2K_3R_3 - 2K_1H_3^2$$

The equation was implemented correctly in the program at line 250.

Uniform Random Number Generator for the TI-59 - V14N1P19 - In the seventh line a square root sign was omitted. The statement should have read "The standard deviation should be $1/\sqrt{12}$ = 0.2887042612..."

In the eighth line an exponent of 31 was omitted. The statement should have read "... the generator yields a full cycle of $2^{31} - 2$ different values."

The TI-81 - V14N2P10 - Robert Lucas noted that the first line under "Benchmarks" reads "I tested the TI-68's arithmetic ...". It should read "I tested the TI-81's arithmetic ...".

HP Paper for the PC-324 - V14N1P13 discussed Hewlett-Packard's 82175 paper as a low cost alternative source for use with the PC-324. I noted that I did not remember which member had first called the paper to my attention. During a call to discuss additional subprograms in the TI-74 Statistics module Nick Pietras reminded me that he was the first.

Circle Best Fit - A program by this name appears in Maurice Swinnen's *Statistical Library* book which was reviewed in V12N3P4. The program listing on pages 11-12 of the book contains three errors:

1. The solution is not undefined when R8 = 0 as indicated by line 320. Rather, the solution is undefined when S0, which is calculated in line 330, is equal to zero. To correct the error interchange lines 320 and 330 and change the new line 330 to state IF S0=0 THEN PRINT "Solution Undefined":STOP.

2. The prediction calculations will return an answer for an input value which is beyond the range of values covered by the circle. The erroneous results can be avoided by removing the ABS commands in lines 430 and 510. Then, if an input value which is not included in the circle is entered the program will show an error due to the attempt to take the square root of a negative number.

3. The prediction calculations show only one of the two values which can be predicted from a single input value.

ERRATA - (cont)

Fitting a Circle - The equations for fitting a circle which appear in William Kolb's book *Curve Fitting for Programmable Calculators* contain an error. The documentation states that:

"The solution is undefined whenever R08 is equal to zero."

The statement should read:

The solution is undefined whenever R10 is equal to zero.

The reason the solution is undefined when R10 is zero can be seen in the equations for h and k, where the value R10 appears in the denominator of each equation. The error appears on page 96 of the first edition and on page 88 of the second edition. I suspect that it also occurs in the third edition.

The error appears to have carried over directly into the "Circle Best Fit" program in Maurice Swinnen's *Statistical Library* book (see page 2 of this issue.)

MORE BENCHMARK RESULTS - Lee Arrington sent in his results from running one of our benchmark calculations on other devices. The test, 1.0000001 squared 27 times, was originally proposed in the "Computer Recreations" column of the April 1984 issue of *Scientific American*, and described in detail in V9N2P11 and V9N4P3. Lee's results were:

	squared 27 times	1.0000001^134217728
Exact Answer	674530.47074 10845 59...	674530.47074 10845 59...
TI-30-II	674432.8206	*
BA-35	674432.8206	*
TI-35 +	674472.44161 1	674530.47074
TI-66	674520.60673 81	674530.47074
TI-74	674530.31804 225	674530.47074 01
TI-95	674530.31804 26	674530.47074 01
HP-22S	674514.86877	674530.47074 1
HP-28S	674514.86877	674530.47074 1
Casio fx-115	674294.1172	674530.47055
PS/2-80	674530.47552	674530.47603

where the asterisks for the second method for the TI-30-II and BA-35 indicate that the function is beyond the range of the device. His results are the same as those previously published for the TI-66 (V9N2P11), the TI-74 (V11N2P5) and the TI-95 (V11N3P16).

With an IBM-80 using Mathematica (copyright by Wolfram Research, Inc.) he received the same answer for both methods:

674530.47074 10845 59382 68917 80297 46812 84444 41434 10342 03174 23773

which agrees with the first 42 digits as published in V9N5P2 and also agrees with additional digits calculated by Carl Rabe but not published.

Editor's Note: I performed the first test with my SR-10 and obtained 671189.63 .

THE RANDOM NUMBER GENERATOR IN THE TI-81 - The program at the right was used to test the random number generator in the TI-81 in a manner similar to that used in evaluations other devices. The testing involves generating a set of random numbers and sorting them into ten equal width "bins". If the numbers are genuinely random then the quantity of numbers in each of the bins may not be exactly equal to 1/10 of the total quantity sorted, but should be close. See V11N1P8 for testing of the fx-7000G and V11N2P7 for testing of the TI-74. The program also calculates the mean and standard deviation for the generated numbers where the expected values are 0.5 and $1/\sqrt{12}$ respectively. Some comments on the program follow:

* Steps 1 through 5 clear the registers to be used in the analysis and accept a user selected seed.
* Steps 6 through 10 "open" the first ten statistical data points to be used as the "bins" for the sorting process.
* Steps 11 and 12 accept a user selected number of random numbers to be used in the analysis.
* Steps 13 through 21 generate the selected number of random numbers and accumulate the sums needed to calculate the mean and standard deviation. The numbers of random numbers which would fall into each bin are accumulated into the appropriate statistical data points by steps 18 and 19. The numbers falling into each bin are not sorted, but only counted.
* Steps 22 through 28 calculate and display the calculated mean and standard deviation.

```
   Prgm3:RANDOM
1  :ClrStat
   :0→S
   :0→T
   :Disp "SEED="
5  :Input Rand
   :1→I
   :Lbl A
   :0→{x}(I)
   :IS>(I,11)
10 :Goto A
   :Disp "N="
   :Input N
   :N→I
   :Lbl B
15 :Rand→A
   :A+S→S
   :A²+T→T
   :1+Int (10A)→A
   :{x}(A)+1→{x}(A)
20 :DS<(I,1)
   :Goto B
   :ClrHome
   :Disp "x̄="
   :S/N→M
25 :Disp M
   :Disp "σx="
   :√((T-S²/N)/N)→R
   :Disp R
29 :End
```

151 bytes

The values in the "bins" are read from the x values of the statistics data points in the DATA option of the STAT MENU. The following table presents the results of the analysis for four different seed values and ten thousand points.

Seed	1	2	3	4	5	6	7	8	9	10	Mean	Sigma
0	955	1002	994	1032	1055	1025	979	978	1016	964	0.49941	0.28602
1	1003	1027	992	997	1037	976	1017	984	953	1014	0.49744	0.28853
2	1005	974	986	1058	1017	959	970	980	1047	1004	0.50048	0.28908
	1028	997	1032	1013	989	1011	995	965	1032	938	0.49533	0.28803

PPX PROGRAM 798017 FOUND - V14N2P12 reported that Bill Wilburn was trying to find a copy of the PPX program "Draftsmen's Right Angle Trigonometry (Miter)". In late October Bill reported that he had received a copy of the porgram from H. D. Zempel, our member in Singapore. I am always gratified by the willingness of our club members to help others.

<u>SCRAMBLING WITH THE TI-81</u> - P. Hanson. The 100 random integer problem was originally proposed in V5N8P1 as "Produce a list of integers between two predetermined limits. Randomize them in such a way that there is no repetition. In other words, each randomly generated integer is unique. A minimum of 100 different integers are required, but any larger number is accepted with gratitude ..." Most solutions which have appeared in our newsletter followed a scheme described in Richard Borger's letter "Faster Algorithms" in the February 1982 issue of *BYTE*. The idea is to first generate a list of sequential numbers which meet the requirements, and then scramble the list. Solutions for the TI-59 by Richard Snow and Jeff Rosedale appeared in V6N1P9. A fast mode program by Sidney Hack appeared in V7N4/5P13. An improved fast mode version appeared in V10N2P23. It would generate a list of 100 scrambled integers (1 through 100) in one minute 35 seconds. An equivalent BASIC program by Larry Leeds appeared in V10N3P17. It would generate a list of 100 scrambled integers in 12 seconds on the CC-40.

An equivalent mechanization for the TI-81 demonstrates:

* storing and recalling of statistical data points in a program (page 7-16 of the manual).

* use the random number generator in a program (page 2-8 of the manual).

* use of the sorting function (page 7-6 of the manual).

The program, which appears at the right, will generate a list of 100 scrambled integers in 9 seconds. Lines 1 through 7 provide for user selection of the range of integers and a seed for the random number generator, and clear the statistical data registers. Lines 8 through 13 generate a sequential list of integers as the x values and a list of random numbers as the y values. Line 14 sorts the random numbers in the y value list, effectively scrambling the integers in the x list in the process. Lines 15 through 20 provide for output of the scrambled list of integers to the screen.

```
   Prgm5:SCRAMBLE
 1 :Disp "MIN Int = "
   :Input L
   :Disp "MAX Int = "
   :Input M
 5 :Disp "SEED ="
   :Input Rand
   :ClrStat
   :1→N
   :Lbl A
10 :L+N-1→{x}(N)
   :Rand→{y}(N)
   :IS>(N,M-L+1)
   :Goto A
   :ySort
15 :1→N
   :Lbl B
   :Disp {x}(N)
   :Pause
   :IS>(N,M-L+1)
20 :Goto B
   :Stop
```

107 bytes

<u>User Instructions</u>:

1. Start the program and enter the range of integers and the seed in response to the prompts.

2. The program stops with the first integer of the scrambled list in the display. Press enter to see the remainder of the list. The appearance of the prompt for a new minimum integer signals the end of the list.

An interesting situation occurs during the output. I have not found a way to return to user control other than to either complete the output list, or turn the calculator off. Has anyone else had similar experience.

<u>TI-81 PROGRAM LENGTHS</u> - The ERASE option of the PRGM key shows program lengths in bytes. I failed to show the lengths in the documentation for the two programs in V14N2. The lengths were 203 bytes for the bestfit program on V14N2P16 and 733 bytes for the elllipse program on V14N2P26-27.

<u>BEST</u> <u>FIT</u> <u>CIRCLE</u> - V13N1P26-27 presented a best fit circle program for the TI-74. The program was a translation of a program by the same name in Maurice Swinnen's *Statistical Library* book. During some additional work with the translated program it was found that the program would indicate that the solution was undefined for the case (4,0), (0,4), (-4,0) and (0,-4), where clearly a circle with the center at the origin and a radius of 4 units would fit the input data. The problem was traced to the use of the wrong value in the test at line 320 of the program listing on V13N1P27. Eventually, several errors were found in both the translation and in the original program as noted on page 2 of this issue:

* An incorrect variable was used in the determination of when the solution is not defined.

* The prediction calculations would return an answer for an input value which was beyond the range of values covered by the circle.

* The prediction calculations would show only one of the two values which can be predicted from an input value.

Those errors were slavishly copied into the translation in V13N1P26-27. A corrected program appears on page 7 of this issue. The changes relative to the program on V13N1P27 are:

* Line 100 is changed so that the prompt indicates that a null string is used to indicate the end of input data.

* Lines 130 and 135 are changed to accept a null string as the end of input data.

* The value S0 is calculated at line 320 instead of at line 330.

* The test for the undefined condition is moved from line 320 to 330 so that the value of S0 can be used in the test.

* Lines 500 through 690 were changed in several places to correct the errors in the prediction section of the original program:

    * Lines 520 and 620 are changed to use null strings instead of the letter E to indicate that no input is intended.

    * Lines 540 and 640 include a test which determines if the input values for the prediction are within the range of values encompassed by the circle solution.

    * Lines 570-575 and 670-675 print both output values for a prediction.

<u>Examples</u>:

The printout for the same example which appeared on V13N1P27 appears on page 7. Note that two outputs appear in response to a prediction. One of printouts at the right checks for a correct solution by the program for the case of (4,0), (0,4), (-4,0), and (0,4). The other printout at the right checks that an undefined solution results for four points on a straight line; e.g., (1,1), (2,2), (3,3) and (4,4).

```
Best Fit Circle

X =  4
Y =  0

X =  0
Y =  4

X = -4
Y =  0

X =  0
Y = -4

 h =  0
 k =  0
 r =  4
```

```
Best Fit Circle

X =  1
Y =  1

X =  2
Y =  2

X =  3
Y =  3

X =  4
Y =  4

Solution Undefined
```

Best Fit Circle - (cont)

Program Listing

```
10 A$="Best Fit Circle":
PRINT A$:PAUSE 1
20 DIM X(50),Y(50)
25 INPUT "Use Printer? Y
/N ";Z$
30 IF Z$="Y"OR Z$="y"THE
N PN=1 ELSE 100
35 PRINT "Device Numbers
:":PAUSE 1
40 PRINT "For the HX-100
0 enter 10":PAUSE 1
45 PRINT "For the PC-324
 enter 12":PAUSE 1
50 INPUT "Enter device n
umber ";D$
55 OPEN #1,D$,OUTPUT
60 IF D$="10"THEN PRINT
#1,CHR$(18)
65 PRINT #1:PRINT #1,A$
70 PRINT #1
100 PRINT "End Input by
Entering "&CHR$(255):PAU
SE 1
110 N=1
120 X$="X = "
125 Y$="Y = "
130 INPUT X$;XX$:IF XX$=
""THEN 190
135 INPUT Y$;YY$:IF YY$=
""THEN 190
140 X(N)=VAL(XX$)
145 Y(N)=VAL(YY$)
150 IF PN=0 THEN 180
155 PRINT #PN,X$;X(N)
160 PRINT #PN,Y$;Y(N)
170 PRINT #PN
180 N=N+1:GOTO 130
190 N=N-1
200 S6=0:S7=0:S8=0:S9=0:
T0=0:T1=0:U6=0:U9=0:V0=0
:V2=0
210 FOR I=1 TO N
220 U=X(I):V=Y(I)
230 S6=S6+U:S7=S7+U*U:S8
=S8+V:S9=S9+V*V:T0=T0+U*
V:T1=T1+1
240 U6=U6+V*U*U:U9=U9+U*
V*V:V0=V0+U*U*U:V2=V2+V*
V*V
250 NEXT I
300 R5=T0*T1-S6*S8:R6=(S
7+S9)*S8-(U6+V2)*T1
310 R7=S8*S8-S9*T1:R8=(S
7+S9)*S6-(U9+V0)*T1:R9=S
6*S6-S7*T1
320 S0=R7*R9-R5*R5
330 IF S0=0 THEN PRINT #
PN,"Solution Undefined":
PAUSE:GOTO 700
340 H=(R5*R6+R7*R8)/(2*S
0)
350 K=(R5*R8+R6*R9)/(2*S
0)
360 AA=2*(H*S6+K*S8):BB=
(H*H+K*K)*T1
370 R=SQR((S7+S9-AA+BB)/
T1)
400 PAUSE ALL
410 PRINT #PN," h = ";H
420 PRINT #PN," k = ";K
430 PRINT #PN," r = ";R
440 PAUSE 0:PRINT #PN
450 Q$="No Solution"
500 P$="Predict y for x"
505 PRINT #PN,P$:PRINT #
PN
510 INPUT P$&" = ";XX$
520 IF XX$=""THEN 600
530 XX=VAL(XX$)
540 D=R*R-(XX-H)^2:IF D>
=0 THEN 550
545 PRINT Q$:PAUSE:GOTO
510
550 PAUSE ALL
560 PRINT #PN,X$;XX
570 PRINT #PN,"Y1= ";K-S
QR(D)
575 PRINT #PN,"Y2= ";K+S
QR(D)
580 PAUSE 0:PRINT #PN
590 GOTO 510
600 P$="Predict x for y"
605 PRINT #PN,P$:PRINT #
PN
610 INPUT P$&" = ";YY$
620 IF YY$=""THEN 700
630 YY=VAL(YY$)
640 D=R*R-(YY-K)^2:IF D>
=0 THEN 650
645 PRINT Q$:PAUSE:GOTO
610
650 PAUSE ALL
660 PRINT #PN,Y$;YY
670 PRINT #PN,"X1= ";H-S
QR(D)
675 PRINT #PN,"X2= ";H+S
QR(D)
680 PAUSE 0:PRINT #PN
690 GOTO 610
700 P$="Add more data (Y
/N) ? "
710 INPUT P$;Z$
720 IF Z$="N"OR Z$="n"TH
EN 999
730 PRINT #PN,"Additiona
l Input Data"
740 PRINT #PN
750 N=N+1
760 GOTO 120
999 END
```

Examples

```
Best Fit Circle

X =  12
Y =  1

X =  14
Y =  3

X =  15
Y =  6

X =  14
Y =  9

 h =  9.357142857
 k =  5.948979592
 r =  5.577611132

Predict y for x

X =  4
Y1=  4.396315318
Y2=  7.501643866

X =  5
Y1=  2.466875047
Y2=  9.431084137

Predict x for y

Y =  9
X1=  4.687986573
X2=  14.02629914

Additional Input Data

X =  12
Y =  11

X =  9
Y =  12

 h =  8.976331361
 k =  6.035502959
 r =  5.89669835

Predict y for x

X =  4
Y1=  2.872090621
Y2=  9.198915297

Predict x for y

Y =  9
X1=  3.878997869
X2=  14.07366485
```

<u>IMPROVED SOLUTION FOR THE QUADRATIC</u> - R. Prins, P. Hanson, C. Williamson. In V12N1P21 Robert Prins reported that the "QAD" function on the "FUNC" menu of the TI-95 would give incorrect answers if c = 0. He speculated that the quirk was a result of the use of a more accurate version of the standard quadratic formula. V14N1P17 reported that the quirk would cause problems for special cases in the TI-95 implementation of an intersection of two circles program. Additional testing showed that if c = 0 then the solution fails due to overflow if b > 0 but the solution is correct if b < 0. After some searching I found three references which mention an improved solution for the quadratic. One was in Maurice D. Weir's *Calculus by Calculator* published in 1982. Page 31 gives the standard algebra book solution but then states:

> "... The program QUADS calculates the real roots (if they exist) for any specified quadratic equation. The formula used is a modification of Equation 1.11 [the standard form] to reduce roundoff error."

The first 78 steps of the QUADS program on page 338 provide for calculation of real roots using the modified equation. Analysis of the program shows that the solution sequence is:

1. Calculate the determinant $D = B^2 - 4AC$
2. Calculate the value $E = (\sqrt{D} + |B|)/2A$
3. Change the sign of E if B<0.
4. Calculate the real roots as C/AE and E.

A more thorough treatment appears in the appendix "Accuracy of Numerical Calculations" to the *HP-15C Advanced Functions Handbook*, also published in 1982. The discussion on page 191 presents the usual formula and the algebraically equivalent formula. (Readers should note that for some reason H-P chooses to define the quadratic as $c - 2bz + az^2$ where the -2 added to the coefficient of the first degree term results in somewhat different equations that those usually encountered. Charlie Williamson notes that Hamming's books use $ax^2 + 2bx + c$.) The H-P discussion states:

> "... the algebraically equivalent formula ... translates into a much more accurate program F whose errors do no more damage than would a perturbation in the last (10th) significant digit of c. Such a program will be listed later (page 205) and must be used in those instances, common in engineering, when the smaller root y is needed accurately despite the fact that the quadratic's other unwanted root is relatively large."

Pages 205-211 of the handbook continue the discussion. Two HP-15 programs are actually provided. More importantly, sample problems are presented which can be used to test the capability of any quadratic program. The printout at the right was obtained with a new TI-59 program which uses the solution from *Calculus by Calculator*. The first six cases are from page 207-208 of the *HP-15C Advanced Functions Handbook*. The reader should note that

```
            1.      A
           -4.      B
            3.      C

            3.      R1
            1.      R2

            1.      A
            0.      B
            4.      C

            0.      Re
            2.      IM

            1. -13  A
           -2.      B
            1.      C

            2.  13  R1
           0.5      R2

       654323.      A
     -1308644.      B
       654321.      C

            1.      R1
  .9999969434       R2

        11713.      A
     -1470492.      B
     46152709.      C

  62.77179203       Re
  .0000853752       IM

        80841.      A
     -1975288.      B
     12066163.      C

  12.21711755       Re
  .0013745136       IM

  4877361379.       A
 -9754525226.       B
  4877163849.       C

  .9999797501       R1
  .9999797501       R2

            1.      A
      -222223.      B
   1.2193329 10     C

       123458.      R1
        98765.      R2

            1.      A
     10000000.      B
            1.      C

    -10000000.      R1
   -0.0000001       R2

            0.      A
            1.      B
           -2.      C

            2.      R1
   9.9999999 99?    R2
```

Improved Solution for the Quadratic - (cont)

* the values for B in the printout are different by a factor of -2 from those in the table on page 207 due the use of different forms of the quadratic equation as discussed above, and

* the solutions for the fourth through sixth cases are the correct values from page 208 of the handbook, not the incorrect values from page 207 of the handbook which are obtained with the shorter HP-15 program. The success of the TI-59 program is directly related to its 13 digit arithmetic.

The seventh and eighth cases are from page 211 of the *HP-15C Advanced Functions Handbook*. They were devised to illustrate two cases which the first, shorter HP-15 program cannot solve, but which the second, longer HP-15 program (using double precision techniques) can solve. It turns out that the new TI-59 program using only single precision techniques cannot solve the seventh case successfully. The value of the discriminant is the difference between two very large numbers:

$$b^2 - 4ac = 9{,}754{,}525{,}226^2 - 4 \times 4{,}877{,}361{,}379 \times 4{,}877{,}163{,}849$$

$$= 95{,}150{,}762{,}384{,}670{,}351{,}076 - 95{,}150{,}762{,}384{,}670{,}351{,}084 = -8$$

which, of course, means that there must be two complex roots with very small imaginary parts (Im = 2.8995E-10). The new TI-59 program does obtain the correct answer for the eighth problem where the reader must know that the value for C is obtained with the keyboard sequence 1,219,332,937 x 10 = which is carried correctly inside the calculator but is truncated to the display as 1.2193329E+10.

The ninth case is the one proposed by Robert Prins in V12N1P21. The tenth case illustrates the printout when A=0. For that case the built-in routine in the TI-95 displays the message "NOT QUADRATIC".

In 1978 a discussion of solving quadratic equations appeared in the V3N9P2 issue of *52 Notes*. Editor Richard Vanderburgh wrote:

> "... But just having an efficient routine won't always guarantee good results. When working with real numbers it is important to keep an eye out for critical data dependencies. For quadratics, when the 4AC term is small (but not zero) compared to B^2, $\sqrt{d}$ can easily be indistinguishable from B, even though 4AC is measurable, resulting in the smaller root being miscalculated to be zero. The best approach to detecting this problem depends on whether the user expects to catch troublesome inputs by eye, or expects the machine to flag potential trouble. Solving the problem depends on by how much $B^2$and 4AC can be expected to differ in relation to machine precision, among other things, and members are invited to suggest viable approaches."

A quadratic-cubic-quartic program by Bill Skillman which appeared in a subsequent issue (V3N12) of *52 Notes* included a routine similar to that in the *HP-15C Advanced Functions Handbook*. There was no discussion of the technique in the text.

Before developing a new program I reviewed some previously published quadratic solutions for the TI-59. The first one is from page IV-82 of *Personal Programming*. That solution uses the standard formula for the roots of a quadratic formula. It obtains the correct answers for test cases 1, 2, 4, 5, 6 and 8. For test cases 3 and 9 it finds the larger root, but returns zero for the smaller root. For test case 7 it fails to find the small imaginary part. It was not designed to handle the case where a = 0 so it fails to solve test case 10 due to divisions by zero.

Improved Solution for the Quadratic - (cont)

Another previously published quadratic solution for the TI-59 is by Stuart Cox in V5N4/5P13. It is an adaptation of an SR-52 solution originally published by Richard Vanderburgh in V2N2P2 of *52 Notes*. Richard's programs were adaptations of an HP-65 program by John Herro. The 45 step routine uses INV P-R and P-R conversions in the solution of complex roots as a means to save program steps. It obtains the correct answers for test cases 1 and 8. For test case 2 it finds the imaginary part satisfactorily, but returns the value 7.9185003e-12 for the real part. For test cases 3 and 9 it finds the larger root, but returns zero for the smaller root. For test case 4 it returns the values 0.9999993084 and 0.999997635 as real roots. For test case 5 it returns the real part correctly and returns 0.0001048809 as the imaginary part. For case 6 it returns the real part correctly and 0.0013745908 (correct to five significant digits) for the imaginary part. For test case 7 it returns the real part correctly and returns the imaginary part as 0.0000009487 (over 3000 times too large). It was not designed to handle test case 10 where a = 0. V5N4/5P13 called this a "quick and dirty" solution. I couldn't agree more. V3N9P2 of *52 Notes* discusses the INV P-R, P-R technique. Send a SASE for a copy.

Yet another quadratic solution for the TI-59 was included in Bill Skillman's quadratic-cubic-quartic program mentioned earlier. The first version of the program appeared in V3N9P2 of *52 Notes*. An "improved" version which appeared in V3N12P2 included a routine similar to that which appeared later in the *HP-15C Advanced Functions Handbook* for better accuracy in cases where the two roots are far apart. The solution also used the INV P/R and P/R conversion techniques which result in reduced accuracy for some test cases. It obtains the correct answers for test cases 1, 2, 3, 8 and 9. For test case 4 it indicates that the roots are complex with a real part of 0.9999984717 and an imaginary part of 0.000002881 when real roots are the answer. For test case 5 it returns the real part correctly and returns 0.0001048809 as the imaginary part, the same solution as the Cox program. For case 6 it returns the real part correctly and returns 0.0013747363 (correct to four significant digits) for the imaginary part. For test case 7 it returns the real part correctly and returns the imaginary part as 0.000002429 (over 8000 times too large). It was not designed to handle test case 10 where a = 0. I can offer magnetic cards for the complete 720 step program which includes an automatic printer sensing routine for $2.00 plus a SASE.

A quadratic solution for the TI-57 by Peter Van Roy appeared in V5N6P11. He used the statistics functions $\bar{x}$ and σ^2 to save program steps. A conversion for use with the TI-59 yields the 48 step program at the right. To use the program enter the value for a and press A. Enter the value for b and press R/S. Enter the value for c and press R/S. If the calculator stops with a flashing display the roots are complex and the real part is displayed. Press RCL 07 to see the imaginary part. If the calculator stops with a steady display the roots are real and one root is displayed. Press RCL 07 to see the other real root.

000	76	LBL	024	77	GE
001	11	A	025	12	B
002	42	STO	026	34	√X
003	03	03	027	22	INV
004	29	CP	028	44	SUM
005	25	CLR	029	07	07
006	42	STO	030	75	-
007	07	07	031	79	$\bar{x}$
008	91	R/S	032	22	INV
009	42	STO	033	44	SUM
010	01	01	034	07	07
011	02	2	035	95	=
012	22	INV	036	91	R/S
013	49	PRD	037	76	LBL
014	01	01	038	12	B
015	25	CLR	039	94	+/-
016	91	R/S	040	34	√X
017	42	STO	041	42	STO
018	02	02	042	07	07
019	69	OP	043	79	$\bar{x}$
020	11	11	044	94	+/-
021	94	+/-	045	61	GTO
022	29	CP	046	13	C
023	22	INV	047	91	R/S

The program obtains the correct answers for test cases 1, 2 and 8. For test cases 3 and 9 it finds the larger root, but returns zero for the smaller root. For test case 4 it returns the values 0.9999996964 and 0.999997247. For test case 5 it returns the real part correctly but returns 0.0001048809 as the imaginary part, the same solution as with the Cox program. For case 6 it returns the real part correctly and 0.0013745908 for the imaginary part, the same answer as with the Cox program. For test case 7 it returns the real part correctly and returns the imaginary part as 0.0000007071, over 2400 times too large. It is not designed to handle test case 10 where a = 0.

Improved Solution for the Quadratic - (cont)

So we see that the three programs which use programming tricks to reduce program steps yield inferior results to those achieved with a straightforward implementations of the quadratic formula for many cases. The QUAD program from *Calculus by Calculator* solves those cases but does not solve quadratics with complex roots. It also does not solve some special cases such as when a = 0.

A new TI-59 program based on the QUAD program in Weir's book but which will also find complex roots and solve the special cases appears below. The program also prints input and output, and automatically senses the use of a printer. Such a program is what the editor calls a "bells and whistles" program, the antithesis of a "quick and dirty" program. The penalty is the use of many more program steps.

* Steps 017 through 025 are the printer sensing routine. An error is induced by an Op 07 command with a value outside the range 0-19 in the display, but only if the printer is connected. The following Op 19 sets flag 07 if an error condition is present. Then, steps 114-122 and 208-213 stop the calculator to display results, but only if the printer is not used.

* Steps 061 through 076 solve the case when a = 0. A second root of ∞ is generated consistent with page 207 of the HP-15C handbook.

* Steps 125-128 set up the print code for the annotation "IM". The 38 in steps 127-128 is from the extended print code table which avoids possible problems if the calculator has been placed in EE mode.

* Steps 146-147 stop the calculator with a flashing display by calling a non-existent label.

* Steps 186-191 provide the correct solution when b = c = 0.

000	76	LBL	040	69	OP	080	02	02	120	91	R/S	160	43	RCL	200	00	0
001	11	A	041	04	04	081	33	X²	121	76	LBL	161	01	01	201	02	2
002	42	STO	042	43	RCL	082	75	-	122	38	SIN	162	54	)	202	69	OP
003	01	01	043	02	02	083	04	4	123	25	CLR	163	95	=	203	04	04
004	91	R/S	044	69	OP	084	65	×	124	29	CP	164	42	STO	204	43	RCL
005	76	LBL	045	06	06	085	43	RCL	125	02	2	165	04	04	205	04	04
006	12	B	046	01	1	086	01	01	126	04	4	166	43	RCL	206	69	OP
007	42	STO	047	05	5	087	65	×	127	03	3	167	02	02	207	06	06
008	02	02	048	69	OP	088	43	RCL	128	08	8	168	22	INV	208	87	IFF
009	91	R/S	049	04	04	089	03	03	129	69	OP	169	77	GE	209	07	07
010	76	LBL	050	43	RCL	090	95	=	130	04	04	170	18	C'	210	39	COS
011	13	C	051	03	03	091	42	STO	131	43	RCL	171	01	1	211	91	R/S
012	42	STO	052	69	OP	092	00	00	132	00	00	172	94	+/-	212	76	LBL
013	03	03	053	06	06	093	77	GE	133	50	I×I	173	49	PRD	213	39	COS
014	91	R/S	054	98	ADV	094	17	B'	134	34	ΓX	174	04	04	214	03	3
015	76	LBL	055	29	CP	095	03	3	135	55	÷	175	76	LBL	215	05	5
016	14	D	056	43	RCL	096	05	5	136	53	(	176	18	C'	216	00	0
017	22	INV	057	01	01	097	05	5	137	02	2	177	43	RCL	217	03	3
018	86	STF	058	22	INV	098	04	4	138	65	×	178	03	03	218	69	OP
019	07	07	059	67	EQ	099	69	OP	139	43	RCL	179	55	÷	219	04	04
020	02	2	060	16	A'	100	04	04	140	01	01	180	53	(	220	43	RCL
021	00	0	061	43	RCL	101	43	RCL	141	54	)	181	43	RCL	221	05	05
022	69	OP	062	03	03	102	02	02	142	95	=	182	01	01	222	69	OP
023	07	07	063	55	÷	103	55	÷	143	69	OP	183	65	×	223	06	06
024	69	OP	064	43	RCL	104	53	(	144	06	06	184	43	RCL	224	98	ADV
025	19	19	065	02	02	105	02	2	145	98	ADV	185	04	04	225	91	R/S
026	25	CLR	066	95	=	106	65	×	146	61	GTO	186	22	INV			
027	69	OP	067	94	+/-	107	43	RCL	147	15	E	187	67	EQ			
028	00	00	068	42	STO	108	01	01	148	76	LBL	188	19	D'			
029	98	ADV	069	04	04	109	54	)	149	17	B'	189	25	CLR			
030	01	1	070	00	0	110	95	=	150	34	ΓX	190	76	LBL			
031	03	3	071	35	1/X	111	94	+/-	151	85	+	191	19	D'			
032	69	OP	072	42	STO	112	69	OP	152	43	RCL	192	54	)			
033	04	04	073	05	05	113	06	06	153	02	02	193	95	=			
034	43	RCL	074	25	CLR	114	87	IFF	154	50	I×I	194	42	STO			
035	01	01	075	61	GTO	115	07	07	155	95	=	195	05	05			
036	69	OP	076	10	E'	116	38	SIN	156	55	÷	196	76	LBL			
037	06	06	077	76	LBL	117	32	X:T	157	53	(	197	10	E'			
038	01	1	078	16	A'	118	35	1/X	158	02	2	198	03	3			
039	04	4	079	43	RCL	119	32	X:T	159	65	×	199	05	5			

Improved Solution for the Quadratic - (cont)

To use the program:

1. Enter a and press A, enter b and press B, and enter c and press C. You may do this in any coefficient order.

2. Press D to solve. With the TI-59 on a PC-100 the input coefficients and the roots are printed. Without a printer and with real roots the calculator stops with the first root in the display. Press R/S to see the second real root. With complex roots the calculator stops with the real part in a flashing display. Press R/S to see the imaginary part.

3. The stored values for the input coefficients are not changed during the solution. Thus, you only have to re-enter those coefficients that are changed in order to solve another problem.

The test case results at the right hand side of page 8 were obtained with this program.

Equivalent programs which were written for the TI-74, the TI-81 and the fx-7000G appear on page 13. The 182 byte fx-7000G program appears to be more compact than the 224 byte TI-81 program due to the ability of an fx-7000 program listing to

1. include both a test and the optional instruction on the same program line, and

2. accept an expression as the argument for a Display instruction.

All three programs yield the same results as those from the new TI-59 program, except that only one root is output when a = 0. The programs also give an incorrect answer for test case 7 where double precision techniques would be needed to find the very small imaginary part.

RECENT PRICES - Page 98 of Volume 22 of the Elek-Tek catalog lists the TI-68 for $36.00 and the TI-81 for $99.00. Page 96 lists the fx-7000G for $63.00 ad the fx-7500G for $79.00 . Add $2.09 per unit for surface shipping and $2.00 per order for insurance and handling. Call 1-800-395-1000.

Page 67 of the #50 issue of the EduCALC catalog continues the offer of a TI-95 for $49.95 and the mathematics, statistics and chemical engineering cartridges for $5.00 each. Page 32 lists the TI-74 for $99.95 and the PC Interface Cable for $54.95 . Page 30 lists the TI-68 for $49.95 . Page 31 lists the TI-81 at $89.95 which is about $11.00 more than I paid in July 1990. For surface shipment there is a $2.00 shipping and handling charge per shipment. Credit card orders accepted at 1-(800)-633-2252. Please mention our club when you order.

TI-59 AND PC-100 AVAILABILITY - 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 present owner, ship the calculator to you, and return anything remaining. We also have two PC-100C's. Neither includes the cloth cover. One needs the eraser fix to print properly - $35 plus shipping. The other is complete except that one of the pegs for the cover over the paper compartment is broken - $45 plus shipping. Add $15 each for shipping and we will return any unused shipping costs. All of the TI-59's shipped will have passed the extended diagnostics in V12N1P4/5. Even so, all used hardware is provided entirely at the buyer's risk.

Improved Solution for the Quadratic - (cont)

TI-81

```
Prgm4:QUDRATIC
:Lbl S
:ClrHome
:Disp "aX² + bX
+ C"
:Disp "a ="
:Input A
:Disp "b ="
:Input B
:Disp "C ="
:Input C
:If A≠0
:Goto A
:-C/B→R
:Disp "R ="
:Disp R
:Pause
:Goto S
:Lbl A
:B²-4AC→D
:If D≥0
:Goto B
:-B/2A→R
:√abs D/2A→S
:Disp "RE ="
:Disp R
:Disp "+/-IM ="
:Disp S
:Pause
:Goto S
:Lbl B
:(√D+abs B)/2A→R

:If B≥0
:-R→R
:0→S
:If R=0
:Goto C
:C/AR→S
:Lbl C
:Disp "R1 ="
:Disp R
:Disp "R2 ="
:Disp S
:Pause
:Goto S
:End
```

224 bytes

fx-7000G

```
"AX² + BX + C"◢
Lbl 9
"A ="?→A
"B ="?→B
"C ="?→C
A≠0⇒Goto 1
"R =":-C/B◢
Goto 9
Lbl 1
B²-4AC→D
D≥0⇒Goto 2
"R  =":-B/2A◢
"+-Im =": Abs D/
2A◢
Goto 9
Lbl 2
(√D+Abs B)/2A→E
B≥0=-E→E
0→F
E=0⇒Goto 3
C/AE→F
Lbl 3
"R1=":E◢
"R2=":F◢
Goto 9
```

TI-74

```
10 X$="More Accurate Qua
dratic":PRINT X$:PAUSE 1
15 REM 8 November 1990
20 Y$="Ax^2 + Bx + C":PR
INT Y$:PAUSE 1
25 INPUT "Use Printer? Y
/N ";Z$
30 IF Z$="Y"OR Z$="y"THE
N PN=1 ELSE 100
35 PRINT "Device Numbers
:":PAUSE 1
40 PRINT "For the HX-100
0 enter 10":PAUSE 1
45 PRINT "For the PC-324
 enter 12":PAUSE 1
50 INPUT "Enter device n
umber ";D$
55 OPEN #1,D$,OUTPUT
60 IF D$="10"THEN PRINT
#1,CHR$(18)
65 PRINT #1:PRINT #1,Y$
100 A$="A = ":B$="B = ":
C$="C = "
110 INPUT A$;A
120 INPUT B$;B
130 INPUT C$;C
140 IF PN=0 THEN 200
150 PRINT #1:PRINT #1
160 PRINT #1,A$;A
170 PRINT #1,B$;B
180 PRINT #1,C$;C
190 PRINT #1
200 IF A<>0 THEN 300
210 PRINT #PN,"R = ";-C/
B
220 IF PN=0 THEN PAUSE
230 GOTO 100
300 PAUSE ALL
310 D=B*B-4*A*C
320 IF D<0 THEN 400
330 E=(SQR(D)+ABS(B))/2/
A
340 IF B<0 THEN 350 ELSE
 E=-E
350 IF E=0 THEN F=0 ELSE
 F=C/A/E
360 PRINT #PN,"R1 = ";E
370 PRINT #PN,"R2 = ";F
380 PAUSE 0
390 GOTO 100
400 PRINT #PN,"Re = ";-B
/2/A
410 PRINT #PN,"Im = +/-
";SQR(ABS(D))/2/A
420 PAUSE 0
430 GOTO 100
450 END
```

CC-40 PERIPHERALS AVAILABLE FROM TI - Charles Good has found that CC-40 peripherals are available from TI at reduced prices. Examples include:

Hex Bus Printer 80	$70	Finance Cartridge	$20
16K Expansion RAM	$40	EE Cartridge	$20
8K RAM Cartridge	$30	Statistics Cartridge	$20
AC9201 Adapter	$19	Mathematics Cartridge	$20
Memo Processor Cartridge	$20	Games I Cartridge	$20

Those prices were effective as of September 1990. Call 1-806-747-1882 for current availability and prices.

Editor's Note: In October 1990 I purchased a Printer 80 (an 80 column printer) and a Memo Processor Cartridge. I plan to review these items in the next issue.

ANOTHER SOURCE OF CC-40 HARDWARE - Charles Good also provided a copy of a catalog of L. L. Conner Enterprise. The firm specializes in support of the TI-99/4A hardware and software, but also provides some support for the TI-59, TI-74 and CC-40. Some sample items and prices are:

Page 4:	CC-40 Manuals (each)	$ 8.00
	CC-40 and peripherals - call for availability	
Page 5:	TI-59 Calculator (used)	45.00
	PC-100C Printer for the TI-59	75.00
	TI-74 Cassette Cable	19.00
	TI-74 8K Constant Memory	24.00
	TI-74 to Hex-Bus Cable	24.00
	TI-74 Printer	49.00
	CC-40 Computer	95.00
	CC-40 RS232 Hex-Bus	95.00
	CC-40 80 Column Printer	95.00

Current prices may be higher or lower depending upon the supply. Used items have been tested and have a 30 day warranty. Shipping charges are via U.P.S. You can order by telephone using Mastercard or VISA by calling (317)-742-8146. There is also a FAX number (317)-423-4879, or you can write to 1521 Ferry Street, Lafayette, IN 47904.

Charles Good writes: "Larry Conner runs this mainly TI-99/4A business from his home. ... He is very knowledgeable about the products he sells. I purchased most of my CC-40 hardware from him."

ADDITIONAL VARIABLES ON THE TI-81 - V14N2P10 noted that elements of the matrices and of the statistical data can be used to supplement the 27 single letter variables. Those variables also have the advantage that they are subscripted. When used as ordinary (unscripted) variables there is a penalty in memory useage:

Letter variables	A	1 byte
Statistics elements	{x}(1)	3 bytes
Matrix elements	[A](1,2)	6 bytes

where we note that the {x}(portion of the statistics element variable is only one byte and the [A] portion of the matrix element is only one byte.

PROTECTING TI-81 MEMORY DURING BATTERY CHANGE - P. Hanson. The paragraph titled "Effects of Replacimg the Batteries" on page B-2 of the *TI-81 Graphics Calculator Handbook* states:

> "The calculator cannot hold data in its memory when the batteries are removed or become discharged. Replacing the batteries has the same effects as resetting the calculator."

As Mark Antony said of Caesar "...If it were so, it was a grievous fault. ...". Fortunately, my experience says that it isn't necessarily so. I was unable to retain the memory if I removed all four cells and then installed four new ones. However, if I removed and replaced one cell at a time, and gave the TI-81 some time to recover before I removed and replaced the next cell, then the memory would be retained. I could keep the power off time to two or three seconds for each cell replacement. The recovery time that was needed was about ten seconds.

HEIGHT OF TIDE INTERPOLATION - Tom Jefferis asked if the graphics capability of the TI-81 might be useful in modeling tidal variations. I had studied tides and currents at Navy OCS way back in 1953 but hadn't done anything with that subject since. So I wrote for help to two of our navigation experts, Hewlett Ladd and Al Mackenzie. Hewlett replied that he really hadn't done much with tides and currents. Al sent in a thick package of material including excerpts from several books, the "Height of Tide at a Given Time" program for the TI-59 (PPX 948002A) and his TI-95 program which will calculate either the height of tide at a given time or the time at which a certain height will be realized.

Both the TI-59 and TI-95 programs require input of the times and heights of adjacent high and low tides, for example, as derived from the annual tide tables published by the National Ocean Service. Those tables include information such as:

1. The predicted times and heights of high water and low water at places known as reference stations.

2. Differences and ratios which can be used to modify the information for the reference stations to make it applicable for subordinate stations.

3. A table for interpolation between adjacent high water and low water events.

Example tables appear in Appendix L of *American Practical Navigator* ("Bowditch", Pub. No. 9, DMA, 1984 Edition) or on pages 118-120 of *Dutton's Navigation and Piloting* (Naval Institute Press, 1972). The reference station data is very similar to that found in the tide tables in daily newspapers. The interpolation table uses a cosine curve to model the variation betwen high and low water. The period of the cosine curve is equal to twice the time difference between high and low water. The peak-to-peak variation of the cosine curve is equal to the difference between the heights of high and low water. For many years seamen have used a rule of thumb for interpolation; e.g., from page 208 of Volume 6 of *The Popular Book of Science* (Grolier, 1961):

> "... the 1-2-3-3-2-1 rule. Note that the sum of these figures is 12. The rule states that, following low tide, the water rises by 1/12 of the range in the first hour, by 2/12 of the range in the second hour, by 3/12 in the third hour, by 3/12 in the fourth hour, by 2/12 in the fifth hour, and by 1/12 in the sixth hour."

That rule work reasonably well where semidiurnal tides, those due to the effects of the moon, predominate. It does not work where diurnal tides predominate. The following programs work with either condition.

Height of Tide Interpolation - (cont)

TI-59 Program Listing from PPX 948002

000	76	LBL	024	17	B'	048	76	LBL	072	95	=	096	39	COS	120	42	STO
001	11	A	025	99	PRT	049	19	D'	073	42	STO	097	94	+/-	121	15	15
002	58	FIX	026	88	DMS	050	99	PRT	074	11	11	098	85	+	122	75	-
003	02	02	027	42	STO	051	42	STO	075	43	RCL	099	01	1	123	43	RCL
004	99	PRT	028	04	04	052	08	08	076	01	01	100	95	=	124	14	14
005	88	DMS	029	92	RTN	053	92	RTN	077	85	+	101	55	÷	125	95	=
006	42	STO	030	76	LBL	054	76	LBL	078	43	RCL	102	02	2	126	42	STO
007	01	01	031	13	C	055	15	E	079	02	02	103	95	=	127	16	16
008	92	RTN	032	99	PRT	056	99	PRT	080	95	=	104	42	STO	128	65	×
009	76	LBL	033	42	STO	057	88	DMS	081	42	STO	105	13	13	129	43	RCL
010	16	A'	034	05	05	058	42	STO	082	12	12	106	43	RCL	130	13	13
011	99	PRT	035	92	RTN	059	09	09	083	75	-	107	07	07	131	85	+
012	88	DMS	036	76	LBL	060	92	RTN	084	43	RCL	108	85	+	132	43	RCL
013	42	STO	037	18	C'	061	43	RCL	085	10	10	109	43	RCL	133	14	14
014	02	02	038	99	PRT	062	03	03	086	95	=	110	08	08	134	95	=
015	92	RTN	039	42	STO	063	85	+	087	35	1/X	111	95	=	135	58	FIX
016	76	LBL	040	06	06	064	43	RCL	088	65	×	112	42	STO	136	01	01
017	12	B	041	92	RTN	065	04	04	089	43	RCL	113	14	14	137	98	ADV
018	99	PRT	042	76	LBL	066	95	=	090	11	11	114	43	RCL	138	99	PRT
019	88	DMS	043	14	D	067	42	STO	091	65	×	115	05	05	139	98	ADV
020	42	STO	044	99	PRT	068	10	10	092	01	1	116	85	+	140	91	R/S
021	03	03	045	42	STO	069	75	-	093	08	8	117	43	RCL	141	00	0
022	92	RTN	046	07	07	070	43	RCL	094	00	0	118	06	06	142	00	0
023	76	LBL	047	92	RTN	071	09	09	095	95	=	119	95	=	143	00	0

To use the program:

1. Enter the time of high water at the reference station and press A.
2. Enter the high water time difference for the substation and press 2nd A'.
3. Enter the time of low water at the reference station and press B.
4. Enter the low water time difference for the substation and press 2nd B'.
5. Enter the height of high water at the reference station and press C.
6. Enter the high water height difference for the substation and press 2nd C'.
7. Enter the height of low water at the reference station and press D.
8. Enter the low water height difference for the substation and press 2nd D'.
9. Enter the time at which the height of tide is needed and press E. Then press R/S to calculate the desired height.

You can perform steps 1 through 8 in any order; however, since input data and the solution are printed without annotation it would probably be best to enter the data in some consistent manner.

Sample problem from PPX 948002:

Determine the height of the tide at Peekskill, New York (on the Hudson River) at 1715 (5:15 PM) EST on 8 October 1976. Data extracted from the tide tables is:

Reference Station		Subordinate Station		
0535	-0.1 feet	Time Differences:	H.W.	+2h24m
1127	7.8 feet		L.W.	+3h00m
1759	-0.2 feet	Height Differences:	H.W.	-1.3 feet
2246	7.2 feet		L.W.	+0.3 feet

```
11.27
 2.24
17.59
 3.00
 7.80
-1.30
-0.20
 0.30
17.15

  3.5
```

The printout at the right is for the data entered in the order prescribed above.

Height of Tide Interpolation - (cont)

Al Mackenzie's program for the TI-95 implements equations (1) and (2) from pages 49-50 of Shufeldt and Newcomer's *The Calculator Afloat* (Naval Institute Press, 1980). A program listing appears at the right.

```
0000 FIX 9 CLR `Ht & Tm`
0010 ` of Tide` PAU
0019 LBL M1
0022 DFN F1:TLR@A1
0029 DFN F2:THR@A2
0036 DFN F3:HLR@A3
0043 DFN F4:HHR@A4
0050 DFN F5:-->@M2 HLT
0058 LBL M2
0061 DFN F1:DTH@A5
0068 DFN F2:DTL@A6
0075 DFN F3:DHH@A7
0082 DFN F4:DHL@A8
0089 DFN F5:-->@M3 HLT
0097 LBL M3 DFN CLR
0102 DFN F1:Td @A9
0109 DFN F2:Hd @B1
0116 DFN F5:-->@M1 HLT
0124 LBL A1 DMS STO 001
0131 HLT
0132 LBL A2 DMS STO 002
0139 HLT
0140 LBL A3 STO 003 HLT
0147 LBL A4 STO 004 HLT
0154 LBL A5 DMS STO 005
0161 + RCL 002 = STO 012
0169 HLT
0170 LBL A6 DMS STO 006
0177 + RCL 001 = STO 011
0185 HLT
0186 LBL A7 STO 007 +
0193 RCL 004 = STO 014
0200 HLT
0201 LBL A8 STO 008 +
0208 RCL 003 = STO 013
0215 HLT
0216 LBL A9 DMS STO 009
0223 RCL 014 - RCL 013 =
0231 STO 017 ( RCL 009 -
0239 RCL 011 )/( RCL 012
0248 - RCL 011 )*180=
0258 COS STO 015 .5*(
0266 RCL 014 + RCL 013 )
0274 -(.5* RCL 017 *
0283 RCL 015 )= HLT
0289 LBL B1 STO 010 ((
0297 RCL 010 - RCL 013 )
0305 /( RCL 014 -
0311 RCL 013 )*2 +/- +1)
0321 INV COS /180=
0328 STO 016 *( RCL 012
0336 - RCL 011 )+
0342 RCL 011 = INV DMS
0348 HLT
```

User Instructions

1. With the first set of definitions in the function key windows:

   a. Enter the time of low water at the reference station and press F1 (TLR).

   b. Enter the time of high water at the reference station and press F2 (THR).

   c. Enter the height of low water at the reference station and press F3 (HLR).

   d. Enter the height of high water at the reference station and press F4 (HHR).

2. Press F5 (-->) and see a new set of definitions in the function key windows.

   a. Enter the high water time difference for the substation and press F1 (DTH).

   b. Enter the low water time difference for the substation and press F2 (DTL).

   c. Enter the high water height difference for the substation and press F3 (DHH).

   d. Enter the low water height difference for the substation and press F4 (DHL).

3. Press F5 (-->) and see a new set of definitions in the function key windows.

   a. To find the height of the tide at a given time, enter the time and press F1 (Td). The height is returned to the display.

   b. To find the time at which a given height of tide occurs, enter the height and press F2 (Hd). The time is returned to the display.

NOTE: The input of data for the reference station in step 1 may be accomplished in any order. Similarly, the input of data for the substation in step 3 may be accomplished in any order. However, the input of data for the reference station (step 1) must be completed before the input of data for the substation (step 2).
