TI PPC NOTES

NEWSLETTER OF THE TI PROGRAMMABLE CALCULATOR CLUB

P.O. Box 1421, Largo, FL 34294

Volume 10, Number 4 Fourth Quarter 1985

This issue includes the first departure from our tradition of coverage limited to products manufactured by Texas Instruments. The illustration at the right is of the CASIO fx-7000G, the newest entry in the hand-held area, with truly amazing display capability for a hand-held. The illustration is slightly undersize. The actual unit is very nearly the same size as the TI-59.

The issue also includes an extended tutorial treatment of a probability problem with emphasis on how one might use various library module routines to obtain the solution. You might ask "Why the emphasis on tutorials?" The answer is that nearly 40 per cent of the current subscribers joined in 1985.

A highlight for the more experienced TI-59 users is Robert Prins' 800 digit square root program.

CASIO Hand-held Programmable 8 lines, 16 characters per line plus 95 x 63 graphics. See pages 6 and 10.

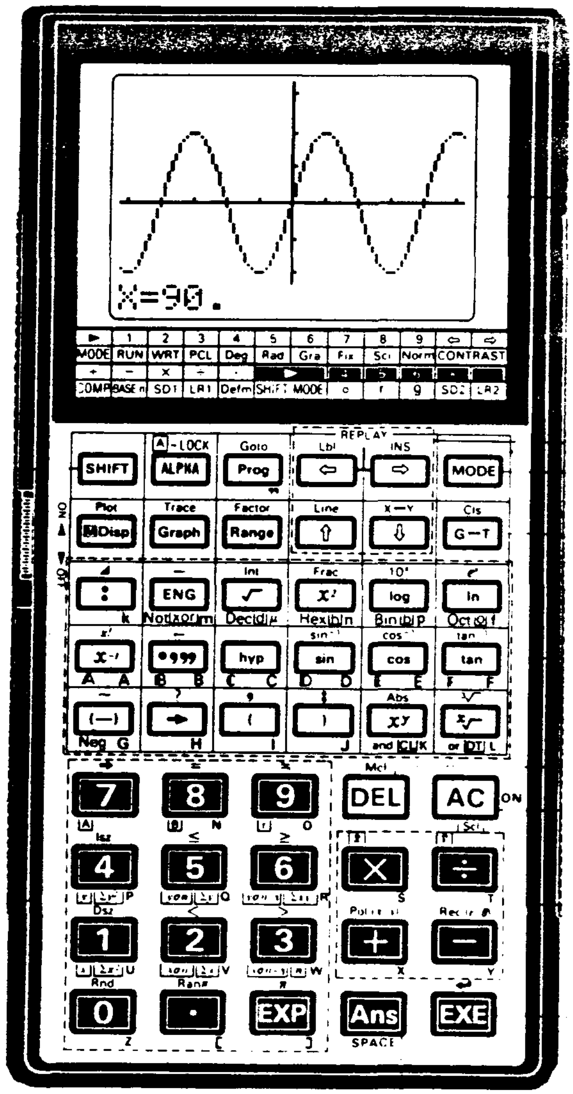


TABLE OF CONTENTS

USED HARDWARE - A complete TI-59 kit. Calculator, case, charger, magnetic cards, documentation, Master library module. $75.00 or make an offer. Barry Lerich, P.O. Box 15038, 3540 Kings Way - Room 1C, Sacramento CA 95821. Telephone (916)-972-3378 between 8 AM and 4 PM, Monday through Friday.

MORE USED HARDWARE - TI-59, like new - never programmed for anything. Make an offer. Bob Harris, c/o BCD ELECTRO, P.O. Box 830119, Richardson TX 75083-0119.

NEW HARDWARE - Jim Carter of Educalc writes that they do not have any more CC-40 peripherals, but do have a few TI-59 modules and solution books. Write to Educalc, 27953 Cabot Road, Laguna Niguel, CA 92677.

A TI-59 EMULATOR FOR THE MACINTOSH? - Dave Leising writes: I wish that you could place the following request in the newsletter - that someone write a fully functional TI-59 calculator to run on Macintosh. There is already one for the HP-12C, I have a copy of it."

Dave sent a clipping of an advertisement for the HP-12C program which states "This desk accessory exactly emulates the Hewlett-Packard 12C Financial Calculator, including programming! Whether you're in business school or on your way to your first million, you'll find this to be the perfect analysis tool. A must for people in the world of finance, real estate, and investment. Works on the 128K Macintosh." The only price shown in the clipping is in the display of the HP-12C - 39.95. For information write to Dreams of the Phoenix, P.O. Box 10273, Jacksonville FL 32247 or call (904)-396-6952.

MORE USED HARDWARE - TI-59 in excellent condition with a one year old battery, ML module, magnetic cards, complete manuals and two Specialty Packettes. $100 or make an offer. Also have a non-working PC-100C. TI quoted a price of about $40 to check it out last year. I will include it with the TI-59 if the purchaser wants it, or ship it separately to anyone who will pay $5 to cover shipping. Joseph Williford, 895 Rushmeade, Jackson TN 38305.

PRINTER PAPER - In V9N1P2 I reported that some of the old stype printer paper was available at a local discount house. I have found that printing on that paper stays legible with age better than some of the newer, whiter paper. Lem Matteson reported a similar experience in V9N2P13. Hewlett Ladd was also having problems with fading and I purchased some of the "old" paper for him. He writes:

> "I ran a program strip on the new paper and the same on a strip of Elek-Tek's and placed them both in front of a sunny window. After a month much of Elek-Tek's had faded badly while your yellowish paper is still like new!"

The last time I looked the paper was on sale at $5.10 for a three roll pack, but there were only about ten packs left.

2532242937.	60
2316632336.	61
2863361733.	62
3624292227.	63
37131427.	64
1700000000.	65
1.000013424061	66
1.000037323361	67
1.0000371344	68
1.00003740244	69
0.	70
0.5	71
0.	72
0.	73
81800.28835	74
85130.300091	75
88270.31116	76
0.	77
0.	78
0.	79

<u>1986 FEDERAL INCOME TAX RETURN</u> - V10N1P16 presented an income tax program by Hewlett Ladd which would accept schedules for either 1984 or 1985, and annotate the output to show which year was being used. A set of magnetic cards has been generated for the 1986 schedules, based on the information in the 1986 Estimated Tax for Individuals (Form 1040-ES). The appropriate print-outs are at the right for Bank 2, and below for bank 4 for the various schedules. Note that the only addition to bank 2 is the 88270.31116 for data register 76.

Joint		Head of Household		Married Filing Separate		Single	
60.6	00	61.6	00	62.6	00	63.6	00
3670.	01	2480.	01	1835.	01	2480.	01
5940.002497	02	4750.002497	02	2970.0012485	02	3670.001309	02
8200.005209	03	7010.05209	03	4100.0026045	03	4750.002605	03
12840.011705	04	9390.008541	04	6420.0058525	04	7010.005759	04
17270.018793	05	12730.014219	05	8635.0093965	05	9170.009009	05
21800.026947	06	16190.020447	06	10900.0134735	06	11650.012977	06
26550.037397	07	19640.027347	07	13275.0186985	07	13920.017063	07
32270.051697	08	25360.041075	08	16135.0258485	08	16190.021603	08
37980.067685	09	31080.057091	09	18990.0338425	09	19640.029538	09
49420.105437	10	36800.075395	10	24170.0527185	10	25360.04441	10
64750.163691	11	48240.115435	11	32375.0818455	11	31080.06157	11
92370.279695	12	65390.187465	12	46185.1398475	12	36800.081018	12
118050.395255	13	88270.290425	13	59025.1976275	13	44780.111342	13
175250.675535	14	116870.427705	14	87625.3377675	14	59670.17388	14
0.	15	0.	15	0.	15	0.	15
0.11	16	0.11	16	0.11	16	0.11	16
0.12	17	0.12	17	0.12	17	0.12	17
0.14	18	0.14	18	0.14	18	0.14	18
0.16	19	0.17	19	0.16	19	0.15	19
0.18	20	0.18	20	0.18	20	0.16	20
0.22	21	0.2	21	0.22	21	0.18	21
0.25	22	0.24	22	0.25	22	0.2	22
0.28	23	0.28	23	0.28	23	0.23	23
0.33	24	0.32	24	0.33	24	0.26	24
0.38	25	0.35	25	0.38	25	0.3	25
0.42	26	0.42	26	0.42	26	0.34	26
0.45	27	0.45	27	0.45	27	0.38	27
0.49	28	0.48	28	0.49	28	0.42	28
0.5	29	0.5	29	0.5	29	0.48	29

For the program listing and user instructions see V10N1P16.

<u>MORE ON A NEW CALCULATOR/COMPUTER FROM TI</u> - V10N3P16 reported that an announcement on the CHUU bulletin board had stated that "...rumor has it that TI will reenter the hand-held market with new products, ala the Sharp 5100 and the TI-59. Will the TI-59 be reborn? ..." Page 9 of this issue reports that the TISOFT newsletter is on hold pending release of a new BASIC-calculator. Robert Stucker reports that page 54 of the <u>Executive Photo Catalog</u> (Volume 5, Number 4) contains the following listings:

TI-74 (BASICALC)	$119.90	Math Module	$39.50
CI-7 Cassette Interface	29.50	Stat Module	39.50
PC-324 Printer	99.90	PASCAL module	39.50
8K RAM	45.95		

Meanwhile, there has been no announcements from TI that I have seen.

A CC-40 DISC MEMORY

Maurice Swinnen sends this information on an external memory for the CC-40. See the ad at the right which is from a TI brochure on the CC-40 from Germany. The Quickdisk is made in Germany by:

MECHATRONIC GMBH
Dresdner strasse 21
D-7032 SINDELFINGEN

but Maurice says they will not sell except in large quantities. The German address where everything is available is:

REISS, Program Service
Bergstrasse 80
D-5584 BULLAY
West Germany

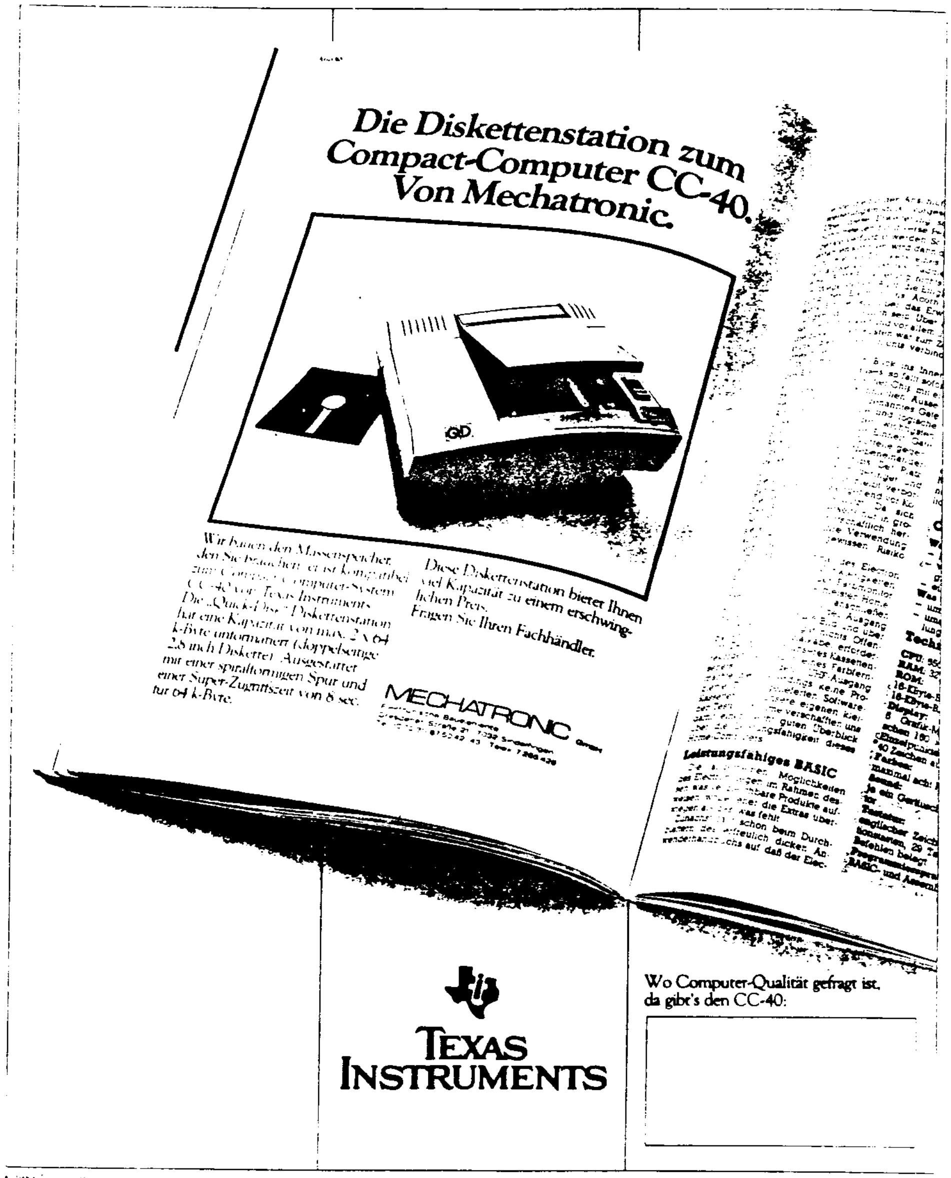
In the United States the unit is available from:

DIGITAL MATRIX SYSTEMS
1761 International Pkwy
Richardson TX 75081

You can call Frederick Winters at

(214)-997-0000

for prices and for more details. The storage capability is 128 KBytes. The unit runs on 4 size D batteries which will last about two hours on continuous use.



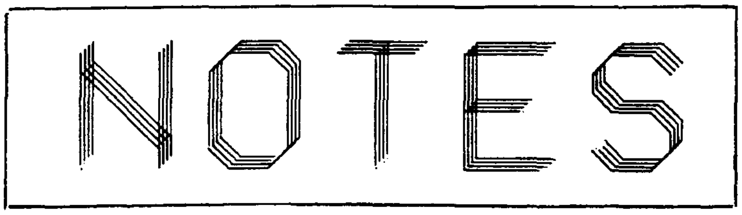
Digital Matrix Systems also has CC-40's and some peripherals for sale. I suggest that you call or write for current status.

CC-40 REPAIR - Maurice also writes that he found someone who repairs and modifies CC-40's. Write to MICROREP, 4413 Cornell Drive, Garland TX 75042. Maurice reports that they repaired his RS-232 Interface so that it now works!

A MINI-PUZZLE - Larry Leeds. Express 2^{2503} in scientific notation using the TI-59 to perform the calculations. You should be able to show that the answer is $3.006\ 624\ 187\ 161 \times 10^{753}$.

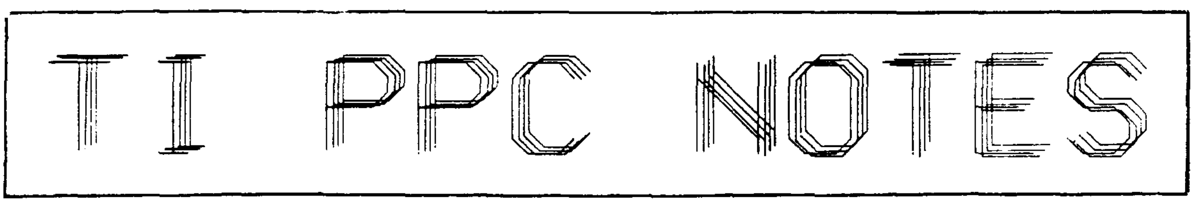
BANNER PROGRAM FOR THE CC-40/HX-1000

This program accepts a string of characters and prints a banner of one-half inch high letters. The program was used to generate the "TI PPC NOTES" logo on the first page of this issue. The letters in the banner can be enlarged without losing their sharpness as is illustrated by the word "NOTES" from the logo.




```
100 INPUT "Enter the banner characte
rs: ":B$
110 OPEN #1,"10",OUTPUT
120 PRINT #1,CHR$(19)
130 PRINT #1,"A1"
140 PRINT #1,"S9"
150 PRINT #1,"M(60,-60)"
160 PRINT #1,"O"
200 FOR J=1 TO LEN(B$)
210 PRINT #1,"M(0,-65)"
220 PRINT #1,"O"
300 FOR I=0 TO 6 STEP 2
310 P$="M("&STR$(I)&","&STR$(-I)&")"
320 PRINT #1,P$
330 P$="T("&SEG$(B$,J,1)&")"
340 PRINT #1,P$
350 NEXT I
360 NEXT J
370 PRINT #1,"M(0,-120)"
900 CLOSE #1
999 END
```

If you watch the banner making process you will see that all of the strokes are completed for one letter before proceeding to the next letter. An alternate process would be to complete single lines for all of the characters before offsetting the printing for subsequent printing; however, this places a requirement on the retracing characteristics of the printer/plotter which is not adequate for well formed letters in the banner. The banner below illustrates the loss of sharpness which results.



NO TEA PLEASE - Charlie Williamson. This is another of Charlie's programming challenges for the TI-58/59. As with his earlier max/min sorter challenge (See V7N1/2P9) t register comparisons are not allowed in the routine. Here's the challenge:

> Given integers X, A and B where B is greater than or equal to A and the integers are smaller in absolute value than $L = 5*10^{12}$ and have been stored in R0, R1, and R2 respectively, write a program with no direct comparisons that returns F(X) as

$$F(X) = \begin{cases} -1 & \text{if } X < A \\ 0 & \text{if } A \le X \le B \\ +1 & \text{if } B < X \end{cases}$$

Charlie believes he has a program which also works for numbers other than integers as well. He also asks for a search of your program for numbers where the program fails. For computers/calculators with fewer digits it may be necessary to decrease the value of L .

THE CASIO fx-7000G - Palmer Hanson. This computer is one the most exciting developments in hand-held devices in recent years. The unit is about the same width as the TI-59, about a half inch longer than the TI-59, and only a half inch thick. It can be carried in your shirt pocket, albeit a little uncomfortably.

The first thing you notice is the display, which accomodates either 8 lines of text with 16 characters per line, or a 95 by 63 dot graphics display. The graphics display capabilities are almost limitless, ranging from built-in plotting of 20 frequently used mathematical functions to the construction of a histogram from single variable input statistics. For the experienced TI-59 programmer the key features are:

* Thirteen digit data registers with ten digit displays that act similarly to the TI-59.

* 26 data memories and a maximum of 422 program steps in normal partitioning. Program steps can be traded for data memories at a 8 for 1 ratio.

* A programming language similar to BASIC, but with calculator-like features as well. To sum the value pi into data register A the user writes π + A → A. Branching capabilities include DsZ, Isz, and labels. Labels are limited to the integers 0 through 9. There are no addresses.

* There are ten program areas also labeled 0 through 9.

* Built-in single and paired value statistics capabilities including linear regression. As with the TI-59 the regression sums are available for recall.

* Pre-programmed conversions such as polar-to-rectangular and rectangular-to-polar.

* The base-n mode provides decimal-octal-hexadecimal-binary conversions, and several logical functions. An important feature for use as a programmer's aid is the sixteen digit display word.

* Functions which are not directly available from the keyboard of a TI-59 (but which may be available depending upon the library module installed) include hyperbolic functions, factorial, cube root, and a random number generator.

* There is no printer interface, and no capability for off-unit program storage.

To illustrate the programming feature of the fx-7000G I selected an old problem from the inertial navigation field. One of the trouble-shooting techniques requires the decomposition of velocity errors into sin wt and (1 - cos wt) components. where w is the Schuler frequency. I already had a least-squares program for my Model 100 which would read input data pairs from a file (see the BASIC program on page 7). I wrote equivalents for the TI-59 and the fx-7000G with one major difference. To maximize the number of input data points without using data packing techniques I imposed the requirement for equally spaced X data with the starting point at the origin. The TI-59 program, instructions for using the TI-59 program, and a sample solution appears on page 8.

The Casio fx-7000G (cont)

The fx-7000G program appears at the left below. My original Model 100 program appears at the right. The first six lines are entered into one program area. The text in quotation marks in the second, fourth and fifth lines provides annotation and prompting. The question marks in the second and fifth lines ask for input from the keyboard.

The second grouping of seventeen lines is entered into another program area. In the second line the notation "~" means enter the value in data memories C[1] through C[5] where subscripts are enclosed in brackets, not in parentheses. Note the similarity between the seventh through eleventh lines of the fx-7000G program and lines 515 through 535 of the BASIC program for the Model 100. In the last two lines the small triangle at the end of each line is the command to stop with the result in the display.

Assume that the two groups of commands have been entered in program areas 2 and 3. Then to use the program press Prog 2 EXE and see the prompt "DELTA-X = ?" in the display. Enter the x increment (say k) and press EXE two times and see the display X = k and the prompt "Y(X) = ?". Enter the value of Y when X is equal to k and press EXE two times. The computer will stop with the display X = 2k and the prompt "Y(X) = ?" asking for the next y input. When all the y values have been entered press Mode 1 Prog 3 EXE. When the solution is complete the sin(wt) component will appear. For the sample problem illustrated for the TI-59 on page 8 the display will read "ACC = 0.8934012669". Press EXE again to see the (1-cos(wt)) component, where for the sample problem the display will read "RATE = 2.786304468". The speed of execution is impressive. The TI-59 program requires 27 seconds for the sample problem; the fx-7000G requires less than 2 seconds.

```
1→I
"DELTA-X = "?→K
Lbl 1
"X = ":IxK◢
"Y(X) = "?→K[I]
I+1→I:Goto 1


Rad
0→C[1]~C[5]
I-1→J:Lbl 2
2xπxJxK÷84→C
sin C→A
1-cos C→B
C[1]+A²→C[1]
C[2]+B²→C[2]
C[3]+AxB→C[3]
C[4]+AxK[J]→C[4]
C[5]+BxK[J]→C[5]
Dsz J:Goto 2
C[1]xC[2]-C[3]²→J
(C[4]xC[2]-C[5]xC[3])÷J→A
(C[1]xC[5]-C[3]xC[4])÷J→B
"ACC = ":A◢
"RATE = ":B◢
```

```
100 DIM X(30),Y(30),R(30)
105 P1 = 4*ATN(3D13)
300 CLS:PRINT:M=1
305 PRINT "      Fit for sinA and (1 - cosA)"
310 FOR I = 1 TO 800:NEXT I
315 PRINT:FILES
320 PRINT:INPUT "   Name of input file ";A$
325 OPEN A$ FOR INPUT AS 1
330 INPUT #1,X(M),Y(M)
335 IF EOF(1) THEN 350
340 M = M + 1
345 GOTO 330
350 CLOSE
355 PRINT
360 PRINT M;"data pairs found"
400 FOR I = 1 TO 6
405 C(I) = 0
410 NEXT I
500 FOR I = 1 TO M
505 A = SIN(P1*X(I)/84)
510 B = 1 - COS(P1*X(I)/84)
515 C(1) = C(1) + A*A
520 C(2) = C(2) + B*B
525 C(3) = C(3) + A*B
530 C(4) = C(4) + A*Y(I)
535 C(5) = C(5) + B*Y(I)
540 NEXT I
545 DET = C(1)*C(2) - C(3)*C(3)
550 IF DET = 0 THEN PRINT "Determinant = 0":GOTO 900
555 A = (C(4)*C(2) - C(5)*C(3))/DET
560 B = (C(1)*C(5) - C(3)*C(4))/DET
565 PRINT "ACC  = ";A
570 PRINT "RATE = ";B
900 END
```

SIN & (1 - COS) FIT ON THE TI-59 - Page 7 contains listings for fx-7000G and Model 100 programs which examine input data pairs to find the components of sin wt and (1 - cos wt) in a least squares sense. A listing for an equivalent TI-59 program appears below. As with the fx-7000G program I imposed the requirement for equally spaced X data with the starting point at the origin.

User Instructions:

1. Enter the x increment and press A. The increment and the annotation " X" will be printed, followed by a prompt for the first y input which includes the number of the point and the x input.

2. Enter the first y input and press R/S. The y input value will be printed, followed by the prompt for the next input. Continue to enter input y data in response to the prompts.

3. When all the input y data has been entered press C. The "ACC" annotation indicates the sin wt part, and the "RATE" annotation indicates the (1 - cos wt) part.

A sample printout from the program appears at the right for a problem with an x increment of 6 and a ramp function. The last prompt for N = 7 is ignored.

```
           6.       ΔX

           1.       N
           6.       X
           1.       Y

           2.       N
          12.       X
           2.       Y

           3.       N
          18.       X
           3.       Y

           4.       N
          24.       X
           4.       Y

           5.       N
          30.       X
           5.       Y

           6.       N
          36.       X
           6.       Y

           7.       N
          42.       X

 .8934012669        ACC

 2.786304468        RATE
```

```
000 76 LBL
001 11 A
002 47 CMS
003 98 ADV
004 70 RAD
005 32 X:T
006 02 2
007 00 0
008 42 STO
009 08 08
010 69 OP
011 00 00
012 07 7
013 05 5
014 04 4
015 04 4
016 69 OP
017 04 04
018 32 X:T
019 42 STO
020 10 10
021 69 OP
022 06 06
023 76 LBL
024 12 B
025 98 ADV
026 69 OP
027 27 27
028 03 3
029 01 1
030 69 OP
031 04 04
032 43 RCL
033 07 07
034 69 OP
035 06 06
036 04 4
037 04 4
038 69 OP
039 04 04
040 43 RCL
041 07 07
042 65 ×
043 43 RCL
044 10 10
045 95 =
046 69 OP
047 06 06
048 91 R/S
049 32 X:T
050 04 4
051 05 5
052 69 OP
053 04 04
054 32 X:T
055 72 ST*
056 08 08
057 69 OP
058 28 28
059 69 OP
060 06 06
061 61 GTO
062 12 B
063 76 LBL
064 13 C
065 36 PGM
066 01 01
067 71 SBR
068 25 CLR
069 98 ADV
070 43 RCL
071 07 07
072 42 STO
073 00 00
074 85 +
075 01 1
076 08 8
077 95 =
078 42 STO
079 08 08
080 69 OP
081 30 30
082 76 LBL
083 78 Σ+
084 02 2
085 65 ×
086 89 π
087 65 ×
088 43 RCL
089 10 10
090 65 ×
091 43 RCL
092 00 00
093 55 ÷
094 08 8
095 04 4
096 95 =
097 42 STO
098 06 06
099 38 SIN
100 42 STO
101 11 11
102 33 X²
103 44 SUM
104 01 01
105 01 1
106 75 -
107 43 RCL
108 06 06
109 39 COS
110 95 =
111 42 STO
112 12 12
113 33 X²
114 44 SUM
115 02 02
116 43 RCL
117 11 11
118 65 ×
119 43 RCL
120 12 12
121 95 =
122 44 SUM
123 03 03
124 43 RCL
125 11 11
126 65 ×
127 73 RC*
128 08 08
129 95 =
130 44 SUM
131 04 04
132 43 RCL
133 12 12
134 65 ×
135 73 RC*
136 08 08
137 95 =
138 44 SUM
139 05 05
140 69 OP
141 38 38
142 97 DSZ
143 00 00
144 78 Σ+
145 43 RCL
146 01 01
147 65 ×
148 43 RCL
149 02 02
150 75 -
151 43 RCL
152 03 03
153 33 X²
154 95 =
155 42 STO
156 06 06
157 01 1
158 03 3
159 01 1
160 05 5
161 01 1
162 05 5
163 69 OP
164 04 04
165 43 RCL
166 04 04
167 65 ×
168 43 RCL
169 02 02
170 75 -
171 43 RCL
172 05 05
173 65 ×
174 43 RCL
175 03 03
176 95 =
177 55 ÷
178 43 RCL
179 06 06
180 95 =
181 69 OP
182 06 06
183 98 ADV
184 03 3
185 05 5
186 01 1
187 03 3
188 03 3
189 07 7
190 01 1
191 07 7
192 69 OP
193 04 04
194 43 RCL
195 01 01
196 65 ×
197 43 RCL
198 05 05
199 75 -
200 43 RCL
201 03 03
202 65 ×
203 43 RCL
204 04 04
205 95 =
206 55 ÷
207 43 RCL
208 06 06
209 95 =
210 69 OP
211 06 06
212 98 ADV
213 91 R/S
```

MEMBERSHIP LISTING - V1ON3P1 noted that several members had asked about a membership list. Those who wished to have their names and addresses listed were invited to write. Only eight members responded:

Richard Gibbons
3632 Chelton Road
Shaker Heights OH 44120

Charles Williamson
P. O. Box 7177
Sacramento CA 95826
(916)-363-1238

George Wm. Thomson
15093 Faust Boulevard
Detroit MI 48223
(313)-835-0400

Peter Stromgren
Hesselogade 56 - 311
DK-2100 COPENHAGEN O
DENMARK

Fred Shetler
977 Oak Street
Indiana PA 15701-1760
(412)-349-2557

Robert A. H. Prins
Alfred Nobellaan 112
3731 DX DE BILT
NETHERLANDS

Laurance M. Leeds
10232 El Dorado Drive
Sun City AZ 85351

William Bowen
14210 Torrey Village Drive
Houston TX 77014
(713)-440-1743

We currently have 182 members in 34 states and 10 foreign countries. California is by far the best represented state with 31 members.

TISOFT STOPS PUBLICATION - Thomas Coppens, the editor of TISOFT writes: "... So the PPC-notes are still alive! I must confess we stoped all activities of TISOFT by January 1, 1986. TISOFT is not dead, but is in a hold position to restart activities on a new BASIC-calculator that is announced for the second quarter of 1986."

To my knowledge that leaves only three newsletters; TI PPC Notes, Programbiten (Swedish) largely dedicated to the 99/4, and Peter Poloczek's TI58/59 Software Club (German).

Is the device mentioned by Thomas the same one as in V1ON3P16? I have received no information from TI.

A 5/6 SPEED TI-59? - William Hawes of Montreal, Quebec writes: "I recently bought a used spare TI-59 that had been manufactured in Holland. It appears to function the same as our North American units except it runs noticeably slower. I timed it on several programs and noted that it runs at almost exactly 5/6ths the speed of U.S. models. Presumably the crystal oscillator is different. Is the difference related to the different A.C. line frequencies in the two countries -- 50 Hz in Holland and 60 Hz in the U.S.A.? The Dutch unit runs at 5/6 speed whether on battery or connected via a 110 volt transformer to the 60 Hz North American line frequency. I'm curious whether this has been noted before and whether its an indication of even more significant differences lurking inside." Can anyone help with an explanation?

MORE ON THE FX-7000G - P. Hanson. I tested the fx-7000G with some some of the benchmarks which we have discussed in earlier issues of TI PPC Notes. The results were quite good, but there were some interesting idiosyncrasies:

$e \times \pi = \pi \times e$

The non-commutative multiply on the TI-59 has been discussed in detail in V9N2P15. That idiosyncrasy was eliminated in the TI-66. Limited tests show that the multiply in the fx-7000G is commutative.

$\sin 45° = \cos 45°$

This inconsistency with the TI-59 is discussed on page C-1 of Personal Programming. Tests confirm that in general on the TI-59 sin(A) is not equal to cos(90-A) to thirteen digits. Of course, TI only promised accuracy to ten digits and cautioned against the use of the guard digits. Personal Programming suggested the use of the EE-INV-EE sequence to truncate the guard digits and leave only the rounded display. However, in V8N3P15 I showed that this would not make sin(39) = cos(51), and furthermore that arguments could be found which would cause the tenth digit to differ for sin(A) and cos(90-A) no matter what truncation scheme was used. We were left with the restriction from page C-2 of Personal Programming which indicated that the trigonometric functions were within +/-1 in the tenth digit. V9N2P12 reported that the TI-66 delivers identical values for sin(A) = cos(90-A).

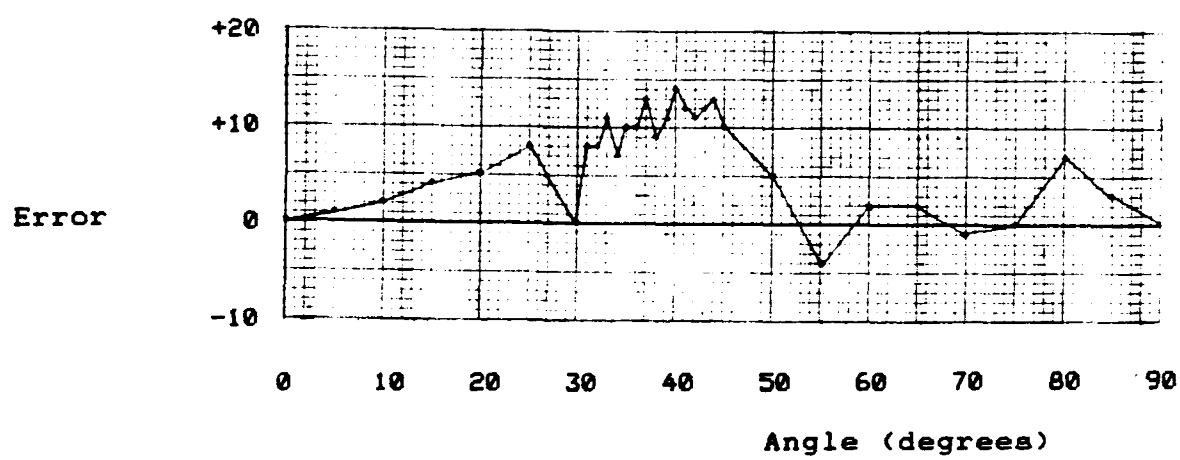
For the fx-7000G Sin(45) - cos(45) = 1.5E-12 , but tests reveal that sin(A) is equal to cos(90-A) everywhere except in the neighborhood of 45 degrees:

sin(45 + 7E-11) - cos(45 - 7E-11) = 1.5E-12

sin(45 + 8E-11) - cos(45 - 8E-11) = 0

Accuracy of the Sine Function

V8N3P18/19 examined the TI-59 sine function at one degree increments over the range from 0 degrees to 90 degrees and found a peak error of 17E-13 and an RMS error of 6.8E-13. V8N4P11 reported that for the same test on the CC-40 the peak error was 5.9E-13 and the RMS error was 1.8E-13. Examination of the sine function of the fx-7000G at five degree intervals shows a peak error of 14E-13 and an RMS error of 5.7E-13. The figure below shows the error curve for the five degree increments, plus the errors at one degree increments from 30 through 45 degrees:



More on the fx-7000G - (cont)

The Square Root - Squared Test

V8N3P13/14 described this test. It is a derivative of the ($\sqrt{2}$) test by Brian Hayes on page 136 of the January 1981 issue of BYTE. For our test we start with an integer, take the square root five times, take the square five times, and compare the result to the original integer. Admittedly the test favors devices which carry guard digits. I tested integers 2 through 25. The display returned the integer in each case. V8N3P15 reported that the display was off by one in the tenth digit for 5, 7, 8, 9, 10 and 23 for the TI-59. Of course the contents of the display register in the TI-59 or the "Ans" register in the fx-7000G will be slightly different, reflecting the truncation effects during the square root and square operations. Sample results for several input integers include:

2	1.999999999921	12	11.99999999969
3	2.999999999904	13	12.99999999942
5	5.000000000132	15	14.99999999882
7	6.999999999765	17	16.99999999940

where some answers are better and some worse than with the TI-59. If one truncates to the displayed value at each step one gets the HP-41C results.

1.0000001 to the 27th Power

V9N2P11 described this test from the "Computer Recreations" column of the April 1984 issue of Scientific American.

Exact	674530.4707410 84559 ...
Mode A (A^2)	674530.3180426
Mode B (AxA)	674516.1465850
Mode C (A^134217728)	674530.4707411

The Mode C answer is better than either the Model 100 or the TI-66.

The Bob Fruit Benchmark

In V8N4P4 Bob Fruit proposed a compound interest problem as another benchmark. The appropriate equation is that for the sum of a geometric series S = [(1 + i)^n - 1]/i . An annual interest rate of ten per cent (i =.10/12) and compounding monthly for thirty years (n = 360) yields:

Exact	2260.48792 47960 86067 ...
fx-7000G using x^y function	2260.48792 4512

which is not quite as good as either the TI-59 or the Model 100.

Accuracy of the Ln Function

V9N6P14 noted that both the Bob Fruit test and the Scientific American test depend upon the accuracy of the Ln function in the vicinity of 1. That accuracy appears to be quite good for the fx-7000G. Detailed results will be published in a coming issue.

THE SOCIAL SECURITY NUMBER PUZZLE

Social security numbers in the US always consist of 9 digits.
My friend has an unusual social security number:
The first two digits on the left are evenly divisible by 2.
And the first three digits are evenly divisible by 3.
As you have guessed, this goes on up to the nineth digit,
such that the entire number is evenly divisible by 9.
What is his social security number?
Note: His number does not contain zeroes, nor are any digits repeated in it. Good luck!

Maurice E.T. Swinnen

A possible, but slow, solution on the CC-40 would run as follows:

```
100 OPEN #1,"16",OUTPUT
110 FOR N=121111119 TO 989999991
120 N$=STR$(N)
130 FOR I=9 TO 2 STEP -1
140 IF VAL(SEG$(N$,1,I))/I<>INT(VAL(SEG$(N$,1,I))/I)THEN 170
150 NEXT I
160 PRINT #1,N
170 NEXT N
180 CLOSE #1
190 END
```

A slightly faster solution, this time requiring only 3 years running time would read as follows:

```
100 OPEN #1,"16",OUTPUT
110 FOR N=121111119 TO 989999991
120 FOR I=0 TO 7
130 A(I)=INT(N/10^I)
140 IF A(I)/9-I<>INT(A(I)/9-I)THEN 170
150 NEXT I
160 PRINT #1,N
170 NEXT N
180 CLOSE #1:END
```

Can you write a FASTER solution, either in calculator language or in any dialect of Basic?

By the way, this entire article was writttten on a CC-40 and printed on a TI HX-3030 companion printer.

A BIT OF HISTORY - Hal Halvorsen. I wonder if the group knows that Asimov's first "Foundation" story (later novel) had the future mentor Hari Seldon using a little wizard calculator with bright red display digits. This was around 1941 in the old Astounding Science Fiction magazine. Asimov once wrote that that was one of the few science fiction predictions he knew that came out right on the money, now dated of course.

PLUMBING DESIGN - D. H. Berry. I have many programs available for the TI-59 which deal with HVAC, piping, and plumbing design. For a free catalog, send a large self-addressed envelope with two stamps to: D. H. Berry, 7693 Ceres Drive, Orlando FL 32822

MORE CAPABILITY FOR THE TI-66 - Robert Prins writes:

1. Is it possible to Dsz on more than ten registers?

2. Is it possible to run a program in a 0.63 partitioning?

3. Is it possible to create a partitioning with 455 steps AND ten registers?

4. Is it possible to create non-sequential programs without using jump or subroutine instructions?

Now, if you know Robert, you know that the answers to all of those questions will be yes. Dave Leising and Patrick Acosta are also working with extending the capability of the TI-66 through the use of pseudo-code. We will try to cover much of the work in this area in the next issue.

THE fx-7000G Ran# FUNCTION - P. Hanson. One of the first programs I tried to write to demonstrate the plotting capability of the fx-7000G was one to demonstrate random walk. I used the Ran# function to generate a pseudorandom number from 0.000 to 0.999, and compared the number to 0.5 to decide whether the next step was positive or negative. To my surprise the resulting "random walk" was anything but random, but rather demonstrated a decided tendency to walk in one direction. Experimentation showed that a comparison of the generated "random number" with a value of about 0.58 would yield a more acceptable random walk characteristic. I wrote a short program to sum 1000 numbers from the Ran# function, and included a variable delay loop which could change the execution time between subsequent calls of the Ran# function. Depending on the amount of the delay I could get mean values of the random numbers generated ranging from 0.37 to 0.72. I also noted that the Ran# function produces normally distributed numbers, not uniformly distributed numbers.

CALCULATOR AVAILABILITY - While there appears to be a new programmable calculator line from TI in the offing the prices for existing calculators are falling rapidly. The latest Educalc catalog (Issue #30) lists the HP-15C at $75.95 and the TI-66 at $49.95 . The TI-66 was on sale at a local discount house for only $39.97 last week. The CASIO fx-7000G is available from Elek-Tek for $52.00 plus $4.00 shipping.

The TI-55III - This unit has appeared on local discount house shelves. The Educalc catalog offers that unit for $39.95. The TI-55III seems to be a repackaged TI-55II. The keyboard has an entirely different feel; hopefully the key bounce problems reported in V9N6P2 and V10N1P2 will be eliminated. Tests at the dealer's counter showed that all the statistics quirks reported for the TI-55II in V8N1P27 persist with the TI-55III.

The TI-55III appears side by side with the TI-55II on some dealer shelves, and at the same price! I can't imagine why anyone would spend forty dollars for a TI-55II or TI-55III when with some shopping one can obtain a TI-66 for essentially the same price.

ODDS. - Recently someone at work claimed to have thrown seventy heads in one hundred tosses of a coin. My first impression was that his coin must have been loaded. The odds against doing that are just too high. So, I polity suggested a closer scrutiny of the coin in question. The "fenomenal tosser" wanted to know what exactly were the odds against having seventy hits in one hundred trials. Somebody in the, always inevitable, audience even taunted with "Well, Mr. Math Wizz, what are the odds?" I couldn't answer that one just off the top of my head, but I promised to have an answer "before the sun would set in the West." I am not used to terms like "the odds against achieving something." My familiarity with probabilty and statistics limits itself to Mean Time Between Failures and an occasional Least Squares Fit.

So, I consulted the people at the Medical Statistics department, people who are concerned with "how many people might possibly succumb to heart desease in a given population, given a specific age......" In spite of their aversion to the obvious gaming application of the solution to my problem, they were very helpful and even let me roam through their library and indicated in which of their references I would most likely find the magic equation. That one turned out to be:

$$Z = \frac{K-NP}{\sqrt{NP(1-P)}}$$

This Z number tells us how many standard deviations the result lies above the chance level. In this equation, K is the number of hits scored in N trials with a probability of P. (P=0.5 for a coin toss, i.e. there is a chance of 1/2 of achieving either heads or tails.)

Various references only gave tables to estimate the significance of Z:

Z	ODDS
1.65	20:1
2.17	50:1
2.33	100:1
3.09	1,000:1
3.71	10,000:1
4.26	100,000:1
4.75	1,000,000:1

Obviously, those tables date from a "pre-programmable-calculator" era, maybe even "pre-computer." So, the solution was to run the data through a curve fitting program. The equation Y=A*e ↑ B*X fits the data with an R^2 of .9878368297. Admittedly, that is not a perfect fit, but it will do in a pinch. I further found that A=.0299807178 and B=3.529593035; e is, of course, the base for the natural logarithms and can be entered as 1 INV LNX. X represents Z and Y are the odds to be found.

Once all of this "hard" work had been done, it was easy, and fun, to write a TI-59 program to compute the odds against having a specific number of hits in a given number of trials with a given probability.

A simple, hand-held version, using truncated values for A and B and presupposing a probability of .5, would read as : LBL A X:T .03 X (1 INV LNX Y^X (3.53 X X:T = R/S .

But the "general public" is never happy with such a simple solution. They want to see something more dramatic, in which the calculator demonstrates clearly "his intelligence" by asking questions and providing answers "in clear English". Hence, the larger, interactive program. A couple of runs are included to show you what to expect. Press A to initialize; enter requested values through R/S.

I highly recommend the program to be used at late night parties to impress the uniniciated, or to enable you to engage in "sure win" bets with aforementioned "suckers".

Maurice E.T. Swinnen

Odds - (cont)

Program Listing:

```
000  76 LBL    040  02  2     080  42 STO    120  04  4     160  01  1     200  69 OP
001  11  A     041  04  4     081  02  02    121  06  6     161  03  3     201  04  04
002  22 INV    042  03  3     082  03  3     122  69 OP     162  69 OP     202  43 RCL
003  58 FIX    043  07  7     083  03  3     123  04  04    163  02  02    203  05  05
004  03  3     044  03  3     084  69 OP     124  43 RCL    164  02  2     204  58 FIX
005  07  7     045  06  6     085  04  04    125  04  04    165  02  2     205  02  02
006  03  3     046  07  7     086  43 RCL    126  69 OP     166  01  1     206  69 OP
007  05  5     047  01  1     087  02  02    127  06  06    167  03  3     207  06  06
008  69 OP     048  69 OP     088  69 OP     128  93  .     168  02  2     208  98 ADV
009  03  03    049  04  04    089  06  06    129  00  0     169  04  4     209  91 R/S
010  02  2     050  69 OP     090  43 RCL    130  03  3     170  03  3
011  04  4     051  05  05    091  00  00    131  65  ×     171  09  9
012  01  1     052  25 CLR    092  65  ×     132  53  (     172  03  3
013  03  3     053  91 R/S    093  43 RCL    133  01  1     173  06  6
014  02  2     054  42 STO    094  02  02    134  22 INV    174  69 OP
015  07  7     055  01  01    095  65  ×     135  23 LNX    175  03  03
016  03  3     056  02  2     096  53  (     136  45 YX     176  03  3
017  06  6     057  06  6     097  01  1     137  53  (     177  07  7
018  07  7     058  69 OP     098  75  -     138  03  3     178  00  0
019  09  9     059  04  04    099  43 RCL    139  93  .     179  00  0
020  69 OP     060  43 RCL    100  02  02    140  05  5     180  01  1
021  04  04    061  01  01    101  95  =     141  03  3     181  03  3
022  69 OP     062  69 OP     102  34 ΓX     142  65  ×     182  03  3
023  05  05    063  06  06    103  42 STO    143  43 RCL    183  05  5
024  69 OP     064  03  3     104  03  03    144  04  04    184  01  1
025  00  00    065  03  3     105  43 RCL    145  95  =     185  07  7
026  25 CLR    066  03  3     106  01  01    146  42 STO    186  69 OP
027  91 R/S    067  05  5     107  75  -     147  05  05    187  04  04
028  42 STO    068  03  3     108  43 RCL    148  03  3     188  69 OP
029  00  00    069  02  2     109  00  00    149  02  2     189  05  05
030  03  3     070  01  1     110  65  ×     150  69 OP     190  69 OP
031  09  9     071  04  4     111  43 RCL    151  01  01    191  00  00
032  69 OP     072  07  7     112  02  02    152  01  1     192  03  3
033  04  04    073  01  1     113  95  =     153  06  6     193  07  7
034  43 RCL    074  69 OP     114  55  ÷     154  01  1     194  03  3
035  00  00    075  04  04    115  43 RCL    155  06  6     195  02  2
036  69 OP     076  69 OP     116  03  03    156  03  3     196  00  0
037  06  06    077  05  05    117  95  =     157  06  6     197  00  0
038  02  2     078  25 CLR    118  42 STO    158  00  0     198  00  0
039  03  3     079  91 R/S    119  04  04    159  00  0     199  02  2
```

Editor's Note: A sample printout from the program appears at the right. The odds against 7 hits or more out of 10 are not very high, the odds against 70 hits or more out of 100 are substantially higher, and the odds against 700 hits out of 1000 are extremely high. This is consistent with the idea that the longer the required string of hits, the more unlikely the occurrence.

```
            TRIALS?
      10.         N
              HITS?
       7.         K
              PROB?
      0.5         P
1.264911064       Z
   ODDS AGAINST ARE
    2.61       TO 1

            TRIALS?
     100.         N
              HITS?
      70.         K
              PROB?
      0.5         P
       4.         Z
   ODDS AGAINST ARE
 40677.98      TO 1

            TRIALS?
    1000.         N
              HITS?
     700.         K
              PROB?
      0.5         P
12.64911064       Z
   ODDS AGAINST ARE
     7.40 17   TO 1
```

The problem can also be solved very directly on the TI-59 with the Applied Statistics library module. After you call Program 20 for the binomial distribution analysis, plug in the numbers, and the probability of success Q(z) is returned. To et the odds against, you calculate (1 - Q(z))/Q(z). The results do not agree very well with those from Maurice's program:

Case	Maurice's Solution	ST-20 Solution
7 of 10	2.61 to 1	4.82 to 1
70 of 100	40678 to 1	25476 to 1

In the following pages I will examine the reasons.

Notes on the Use of ST-20 - P. Hanson. The ST-20 routine in the Applied Statistics module for the TI-59 provides an easy way to solve binomial distribution problems. For the odds against problem the user can:

a. Select program 20.

b. Enter the number of trials (n) and press A.

c. Enter the probability of success of each trial (p) and press B.

d. Enter a value one less than the number of successes (k) and press D. The use of a number one less recognizes that mode D calculates the probability of k or fewer successes.

e. When the solution appears in the display, press

DIV (+/- + 1 =

and see the odds against in the display.

For the 7 out of 10 coin toss problem, the solution in step d requires about 6 seconds. For the 70 out of 100 coin toss problem the solution in step d requires 63 seconds. For the 700 out of 1000 coin toss problem the calculator stops after about ten minutes with a flashing 9.9999999 99 in the display. To determine the source of the error I downloaded the ST-20 program into user memory, pressed RST to return to keyboard control, and set flag 8 to stop execution at an error condition. Then, operating the ST-20 program from user memory for the 700 of 1000 problem resulted in a flashing 1. -99 in the display in about a second. Analysis of the program showed that the first calculation in either mode D or E is (1 - p)^n which underflows for n > 328 when p = 0.5 .

Clearly, it would have been user-friendly to have provided an error trap for this condition in ST-20. You might think that the user can simply set flag 8 before selecting ST-20 from the Solid State module. You will find that the program will NOT stop when the error occurs; rather the program continues to run until the exit from the module occurs. That idiosyncrasy does not seem to be discussed in Personal Programming or elsewhere in the literature. The best advice I can give to avoid the situation where you would wait for a long time for a solution only to find an error is to use ST-20 by first downloading it into user memory and setting flag 8; but remember to return to user memory with a RST before setting flag 8. Alternatively, you can return to user memory with Pgm 00, before or after setting flag 8.

There are some other idiosyncrasies in ST-20. Suppose you try to solve the 210 out of 300 problem using the procedure above (remember to enter 209 not 210 at step d. The calculator will stop after about 3 minutes with a "1." in the display. Step e yields an "odds against" of -7.6923077e10 . Yes, that is a negative number. The problem is that the "1." is really 1.000000000013 . If instead we use mode E to get the probability of more than k successes the program returns "-1.3 -11" , a negative number. So one must use ST-20 with care.

MORE ON ODDS AGAINST - P. Hanson. For a limited range of problems you can use ST-20 to solve "odds against" problems. What if you don't have the Applied Statistics module and want to use the binomial distribution technique? The answer which follows relies heavily on the treatment in Chapter 7 of the Schaum's Outline Series *Theory and Problems of Statistics* by Murray R. Spiegel.

For n statistically independent trials where each trial has a probability of success p and a probability of failure q = (1 - p) then the probability that the event will happen exactly x times in n trials is:

$$p(x) = {}_nC_x p^x q^{n-x} = \frac{n!}{x!(n-x)!} p^x q^{n-x}$$

To find the probability of k or more successes we sum terms from x = k through x = n , remembering that 0! is by definition equal to 1. Many calculators, some programmable and some not, can directly find the combination function ${}_nC_x$. Examples include the HP-11, the TI-55II, and the ML-16 program in the Master Library module for the TI-58/59. All of these mechanizations obtain greater range by alternating multiplication for the terms in the numerator with division by terms in the divisor.

Consider the 7 out of 10 coin flip problem. Since p = q = 0.5 the solution can be simplified to the product of a summation of the combinations ${}_nC_x$ multiplied by 0.5 . Then,

$${}_{10}C_7 = 120 \qquad {}_{10}C_9 = 10$$

$${}_{10}C_8 = 45 \qquad {}_{10}C_{10} = 1$$

The probability for seven or more hits is

(120 + 45 + 10 + 1)*(0.5)^10 = 0.171875

and the odds against are

(1 - 0.171875)/0.171875 = 4.81818....

Of course, hand calculations will become tedious for all but the simplest problem. A program which will calculate the combinations using ML-16 and combine the results with the probability of success is:

```
000  76 LBL    021  04  04    042  43 RCL    063  16  16    084  05  05    105  00  0
001  11  A     022  32 X:T    043  06  06    064  12  B     085  75  -     106  00  0
002  42 STD    023  69 DP     044  42 STD    065  36 PGM    086  43 RCL    107  02  2
003  05  05    024  06  06    045  08  08    066  16  16    087  08  08    108  69 DP
004  32 X:T    025  91 R/S    046  69 DP     067  15  E     088  54  )     109  04  04
005  03  3     026  76 LBL    047  38  38    068  65  ×     089  95  =     110  01  1
006  01  1     027  13  C     048  00  0     069  43 RCL    090  44 SUM    111  75  -
007  69 DP     028  42 STD    049  42 STD    070  07  07    091  09  09    112  43 RCL
008  04  04    029  07  07    050  09  09    071  45 Y^X    092  43 RCL    113  09  09
009  32 X:T    030  32 X:T    051  76 LBL    072  43 RCL    093  05  05    114  95  =
010  69 DP     031  03  3     052  15  E     073  08  08    094  32 X:T    115  55  ÷
011  06  06    032  03  3     053  69 DP     074  65  ×     095  43 RCL    116  43 RCL
012  91 R/S    033  69 DP     054  28  28    075  53  (     096  08  08    117  09  09
013  76 LBL    034  04  04    055  43 RCL    076  01  1     097  22 INV    118  95  =
014  12  B     035  32 X:T    056  05  05    077  75  -     098  77  GE    119  69 DP
015  42 STD    036  69 DP     057  36 PGM    078  43 RCL    099  15  E     120  06  06
016  06  06    037  06  06    058  16  16    079  07  07    100  03  3     121  91 R/S
017  32 X:T    038  98 ADV    059  11  A     080  54  )     101  07  7
018  04  4     039  91 R/S    060  43 RCL    081  45 Y^X    102  03  3
019  04  4     040  76 LBL    061  08  08    082  53  (     103  02  2
020  69 DP     041  14  D     062  36 PGM    083  43 RCL    104  00  0
```

More On Odds Against - (cont)

To use the program:

a. Enter the number of trials (n) and press A. The entry is printed with annotation.

b. Enter the number of successes (x) and press B. The entry is printed with annotation.

c. Enter the probability of success (p) and press C. The entry is printed with annotation.

d. Press D to solve for the odds against x or more successes. The solution and annotation are printed and the calculator stops with the odds against in the display. Some sample outputs are printed at the right.

```
        10.      N
         7.      X
        0.5      P

4.818181818   TO 1


        20.      N
        14.      X
        0.5      P

16.34330136   TO 1


       100.      N
        70.      X
        0.5      P

25476.25379   TO 1
```

This program retains the disadvantages of the ST-20 routine: (1) the range is severely limited, and (2) the execution time is even longer. The time with ST-20 increased linearly with the number of trials. With this program the execution time increases as an $A + Bn^2$ function of the number of trials such that the 7 out of 10 coin toss problem reuires 63 seconds and the 70 out of 100 coin toss problem requires 48 minutes.

To obtain an extended range and much faster execution time the user must go to one of the approximations of the binomial distribution. The one typically used, which is also the basis of Maurice Swinnen's program on pages 14-15, is the normal distribution. An easy solution can be obtained using the ML-14 routine from the Master Library module, and calculating the entry to ML-14-A using the Maurice's formula for Z from page 14. An appropriate program is:

```
000  76 LBL    018  04  4     036  69 OP     054  07  07    072  95  =     090  35 1/X
001  11  A     019  04  4     037  06  06    055  54  )     073  69 OP     091  32 X:T
002  42 STO    020  69 OP     038  98 ADV    056  55  ÷     074  06  06    092  03  3
003  05  05    021  04  04    039  91 R/S    057  53  (     075  98 ADV    093  07  7
004  32 X:T    022  32 X:T    040  76 LBL    058  43 RCL    076  36 PGM    094  03  3
005  03  3     023  69 OP     041  14  D     059  05  05    077  14  14    095  02  2
006  01  1     024  06  06    042  04  4     060  65  ×     078  11  A     096  00  0
007  69 OP     025  91 R/S    043  06  6     061  43 RCL    079  36 PGM    097  00  0
008  04  04    026  76 LBL    044  69 OP     062  07  07    080  14  14    098  00  0
009  32 X:T    027  13  C     045  04  04    063  65  ×     081  12  B     099  02  2
010  69 OP     028  42 STO    046  53  (     064  53  (     082  55  ÷     100  69 OP
011  06  06    029  07  07    047  43 RCL    065  01  1     083  53  (     101  04  04
012  91 R/S    030  32 X:T    048  06  06    066  75  -     084  24 CE     102  32 X:T
013  76 LBL    031  03  3     049  75  -     067  43 RCL    085  75  -     103  69 OP
014  12  B     032  03  3     050  43 RCL    068  07  07    086  01  1     104  06  06
015  42 STO    033  69 OP     051  05  05    069  54  )     087  54  )     105  91 R/S
016  06  06    034  04  04    052  65  ×     070  54  )     088  95  =     106  00  0
017  32 X:T    035  32 X:T    053  43 RCL    071  34 ΓX     089  94 +/-    107  00  0
```

The user instructions are identical to those at the top of the page. The output in step d was modified to include a printout of Z. Execution time is less than 5 seconds and is independent of the input values. Sample outputs appear at the right. The results are not very close to those obtained with the binomial distribution. Better results for small values of Z can be achieved by using the technique from problem 24 on page 133 of Theory and Problems of Statistics. Compensation for the continuous nature of the normal distribution is obtained by using an x value reduced by 1/2 when calculating Z.

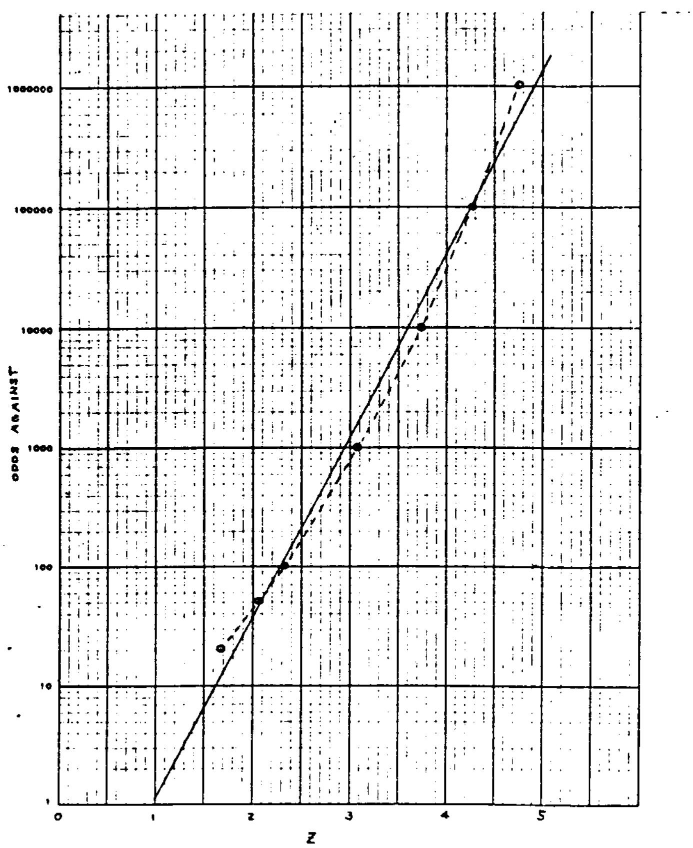
```
        10.      N
         7.      X
        0.5      P

1.264911064      Z

8.713295477   TO 1


        10.      N
        6.5      X
        0.5      P

.9486832981      Z

4.834617506   TO 1
```

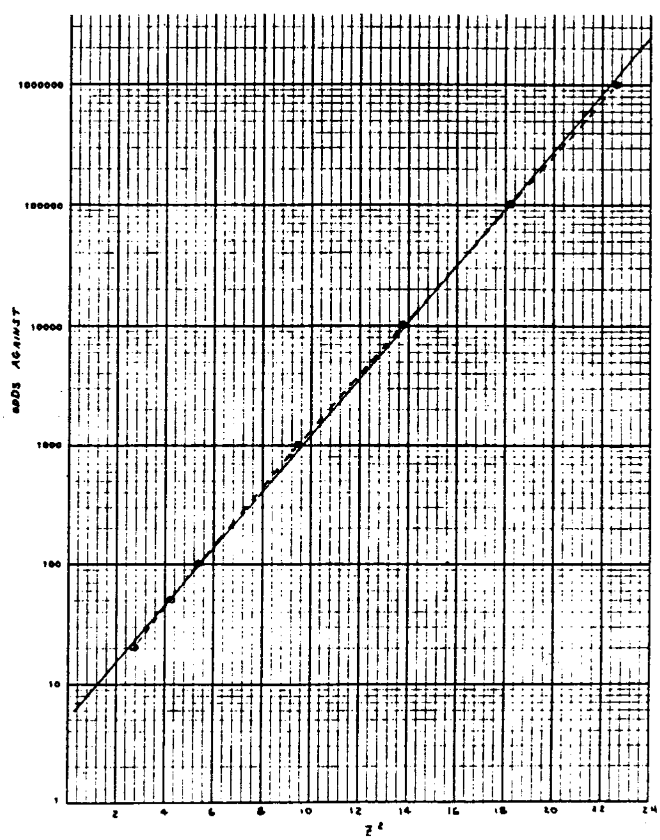
More on Odds Against - (cont)

What if the user only has a calculator which does not accept the library modules; e.g., the TI-57, TI-55II or the TI-66? The user might modify the programs to incorporate downloaded versions of the library module routines. The resulting programs would be several hundred steps long, acceptable but inconvenient for the TI-66 but beyond the capability of the others. A more accurate version of Maurice's curve fit routine might be an appropriate solution.

When I took my first curve fitting course way back in 1948 Professor Eggers emphasized that the first step was always to plot the data. If a curve fit of the form y = Ae^Bx was expected to be appropriate then the data should plot as a straight line on semi-log paper. The plot at the left below includes both the data and Maurice's solution. Maurice noted that the coefficient of determination (R^2) for his curve fit was a not very good 0.9878368297 and the plot confirms that. The plot at the right below for the "odds against" versus the square of Z looks better. The corresponding curve fit function is of the form y = Ae^Bx^2. The solution using the Forecasting - Automatic Curve Choice routine (RE-11) of the Real Estate/Investment module (you simply enter x^2 rather than x for the independent variable) yields A = 5.000691596 and B = 0.5439871991 . The coefficient of determination is a much better 0.9995778561. You will get exactly the same result from curve 6 of Maurice's curve fitting program from V7N1/2P18, which is not surprising since the mathematics is identical to that in RE-11.

Why would we have tried that function? A second old curve fitting rule is to "consider the physics of the problem". Since we are dealing with the normal distribution which has the form y = Ae^($-x^2/2$) it seemed reasonable to try an e^x^2 form. In the plots below "odds against" versus Z or Z^2 yield bows of opposite sense. One can interpolate and find that a function of the form y = Ae^Bx^1.85 yields an even better coefficient of determination (0.9999789512) but not much better solutions.





128	05	5
129	65	×
130	53	(
131	93	.
132	05	5
133	04	4
134	03	3
135	09	9
136	08	8
137	07	7
138	65	×
139	43	RCL
140	04	04
141	33	X²
142	54	)
143	22	INV
144	23	LNX
145	95	=

More on Odds Against - (cont)

A modification for the steps 128 through 145 of Maurice's program from page 15 which will mechanize the Z-squared solution appears at the right. The following table compares the results from the various solutions for "odds against". The results from the ML-16 routine on page 16 will be the same as those from ST-20.

Case	ST-20	ML-14 - .5	ML-14	e^z^2 - .5	e^z^2	Swinnen
7/10	4.82	4.83	8.71	8.16	11.94	2.61
14/20	16.34	16.01	26.16	18.96	28.51	16.58
28/40	119	112	174	107	163	227
42/60	748	669	1027	605	927	1687
56/80	4428	3783	5768	3443	5284	9161
70/100	25476	20782	31558	19604	30129	40678
84/120	143915	112105	169768	111673	171783	156549
98/140	802708	597068	902363	636311	979450	540609
112/160	4436418	3140102	4753493	3626271	5584498	1713393

Why are the results of the curve fit solutions so poor at the low end? One answer may be in the data used for the curve fit. For the 7 of 10 solution the value for Z is only 1.2649 while our curve fit only included Z values of 1.65 or higher. Proponents of curve fitting techniques like to talk of forecasting, e.g., see the comments on V7N1/2P15, the examples in the Real Estate/Investment module instructions, and the example with Thomas Wysmuller's "Polynomial Regression" article in the November/December 1980 issue of PPX Exchange. Forecasting involves extrapolation beyond the range of the input data used to determine the curve fit parameters. That can be dangerous. The "From the Analyst's Desk" column in the January/February 1981 issue of PPX Exchange cautions "WARNING - polynomial regression should not be used for extrapolation". My experience suggests that extrapolation should be used with care no matter what the form of the regression.

An appropriate challenge for the members would be to find the simplest function which provides a rapid solution for the odds against problem. If you choose to try the problem you should note that there was a minor error in Maurice's function table on page 14. The Z corresponding to "odds against" of 50:1 should be 2.06 not 2.17. Changing that value did not improve his solution very much. For the "odds against" versus Z-squared solution I used Z values to three decimal places for the seven "odds against" values from 20 through 1,000,000 of 1.669, 2.062, 2.330, 3.091, 3.719, 4.265, and 4.754. To my way of thinking a useful solution should be particularly accurate for the lower values of "odds against".

THE ST-20 ANOMALY - Other readers may wish to try their hand on an explanation of the negative outputs from ST-20 as described at the bottom of page 16. To that end the ST-20 listing is:

000	76	LBL	026	16	A'	052	92	RTN	078	43	RCL	104	43	RCL	130	43	RCL
001	11	A	027	53	(	053	53	(	079	23	23	105	21	21	131	24	24
002	61	GTO	028	43	RCL	054	42	STO	080	45	Y^x	106	75	-	132	92	RTN
003	00	00	029	21	21	055	22	22	081	43	RCL	107	43	RCL	133	71	SBR
004	48	48	030	65	×	056	94	+/-	082	21	21	108	25	25	134	00	00
005	76	LBL	031	43	RCL	057	85	+	083	54	)	109	54	)	135	65	65
006	12	B	032	22	22	058	01	1	084	42	STO	110	55	÷	136	53	(
007	61	GTO	033	54	)	059	54	)	085	26	26	111	53	(	137	01	1
008	00	00	034	92	RTN	060	42	STO	086	44	SUM	112	43	RCL	138	75	-
009	53	53	035	76	LBL	061	23	23	087	24	24	113	25	25	139	43	RCL
010	76	LBL	036	17	B'	062	43	RCL	088	01	1	114	85	+	140	24	24
011	13	C	037	71	SBR	063	22	22	089	44	SUM	115	01	1	141	54	)
012	61	GTO	038	00	00	064	92	RTN	090	25	25	116	54	)	142	92	RTN
013	00	00	039	27	27	065	29	CP	091	43	RCL	117	55	÷	143	53	(
014	65	65	040	53	(	066	22	INV	092	25	25	118	43	RCL	144	43	RCL
015	76	LBL	041	24	CE	067	77	GE	093	77	GE	119	23	23	145	21	21
016	14	D	042	65	×	068	01	01	094	01	01	120	54	)	146	85	+
017	61	GTO	043	43	RCL	069	43	43	095	24	24	121	61	GTO	147	01	1
018	01	01	044	23	23	070	32	X:T	096	53	(	122	00	00	148	54	)
019	27	27	045	54	)	071	42	STO	097	43	RCL	123	84	84	149	61	GTO
020	76	LBL	046	34	√X	072	24	24	098	26	26	124	43	RCL	150	00	00
021	15	E	047	92	RTN	073	01	1	099	65	×	125	26	26	151	65	65
022	61	GTO	048	59	INT	074	94	+/-	100	43	RCL	126	92	RTN			
023	01	01	049	50	I×I	075	42	STO	101	22	22	127	71	SBR			
024	33	33	050	42	STO	076	25	25	102	65	×	128	00	00			
025	76	LBL	051	21	21	077	53	(	103	53	(	129	65	65			

The instructions for running the down-loaded program to investigate the negative output anomaly are:

1. Enter the number of trials (n) and press A. See n in the display.

2. Enter the probability of success on each trial (p) and press B. See p in the display.

3. To calculate and display the probability of k successes enter k in the display and press C.

4. To calculate and display the probability of k or fewer successes enter k in the display and press D.

5. To calculate and display the probability of more than k successes enter k in the display and press E.

Steps 3 through 5 can be performed at any time and in any order following steps 1 and 2.

MORE ON THE 5/6 SPEED TI-59 - Page 9 reported William Hawes' experience with a TI-59 which seemed to run at 5/6 of the normal speed. Thomas Coppens, the editor of TISOFT, received the following information from the Brussels repair center:

Machine	TI-58	TI-58C	TI-59
Frequency (Khz)	385.8	404.0	461.3

Armed with the information that the TI-58 and TI-59 could use different frequencies I went to the TI58/59 Service Manual and found the following statement on page 4:

"... The ceramic resonator Z1 resonates at a frequency of 455 Khz +/- 1%, ...(The TI-58 can use a 384 Khz resonator which is the preferred part.)"

384/455 is very close to 5/6. This would suggest that a TI-58 resonator is installed in William's TI-59. It does not explain how.

<u>800 DIGIT SQUARE ROOT</u> - Robert Prins. V1ON2P20 and V1ON3P6 presented earlier square root programs. Robert has reorganized his earlier program to include options which extend the capability to 800 digits, and which also permit the entry of many digit arguments. He reports that about 35 hours are required to generate 800 digits. Three different programs are involved. The listings and instructions are on the following pages.

To illustrate the rootfinding technique used to extend the range Robert suggests finding $\sqrt{3}$ = 1.7320 50807 56887 ...

```
        3. 00 00 00 00 00

? x ? =                          Find the largest ? so that ? x ? is as
                                 close to 3 as possible.
1 x 1 = 1

        2. 00                    Add both ones.

2? x ?                           Find the largest ? so that 2? x ? is as
                                 close to 200 as possible.
27 x 7  1. 89

           11 00                 Add 27 and 7

34? x ?                          Find the largest ? so that 34? x ? is
                                 as close to 1100 as possible.
343 x 3    10 29

              71 00
```

If you continue the above process with 3 more steps you will end up with 346410 and a remainder of 27975. While doing so you will notice that the first factor only changes in the last digits. It seemed very natural to me to divide 27975 by 346410. The result is 0.08075690 ... The first six digits after the decimal point are the next six digits of 3; that is, we can find 3 to 12 digits by finding 6 digits using the above algorithm and then divide the remainder by two times the first six digits. I wonder if any software designer is aware of this capability which could save a lot of time.

The instructions for the baseline and long number versions of the square root programs suggest replacing a Nop with R/S when operating without a printer. That will stop the calculator as each ten digit group is completed. That is not user friendly. The short program at the right will allow the user to re-display or re-print the solution after all the desired digits have been calculated. Simply enter the program, press B, followed by R/S as required to print or display the solution. The program can be used at any time, without disturbing any registers, so you may use it after generating say 200 digits, reload the baseline program, and proceed to generate additional digits in accordance with the instructions for the "continue" mode. The program as listed stops as each group of ten is printed and/or displayed. If the R/S at step 033 is changed to a Nop then the program will print all of the calculated digits before stopping.

```
000  76 LBL      021  85  +
001  12  B       022  43 RCL
002  53  (       023  01  01
003  09  9       024  48 EXC
004  00  0       025  04  04
005  75  -       026  75  -
006  43 RCL      027  22 INV
007  07  07      028  59 INT
008  42 STO      029  49 PRD
009  02  02      030  04  04
010  54  )       031  54  )
011  42 STO      032  99 PRT
012  03  03      033  91 R/S
013  00  0       034  69 OP
014  42 STO      035  23  23
015  04  04      036  97 DSZ
016  53  (       037  02  02
017  73 RC*      038  00  00
018  03  03      039  16  16
019  55  ÷       040  00  0
020  02  2       041  92 RTN
```

800 Digit Square Root - (cont)

Baseline Version: This program is essentially equivalent to the program in V10N3P6, but has been reorganized for interaction with the options on pages 24 and 25. After entering the program below:

1. Enter the argument and press A. The argument must be greater than 1 and less than 100. The argument is printed.

2. Enter the number of groups of ten digits and press R/S. The number must not exceed 40. When a flashing "1." appears press 7 and then EE to start the calculation in fast mode.

3. If a printer is being used then each group of ten digits is printed as it is completed. The calculator stops with a zero in the display.

4. To generate additional groups of ten digits enter the number of additional groups and press C (for continue). The total number of groups entered using modes A and C must not exceed 40. When a flashing "1. 10" appears in the display press 7 and then EE to continue calculations in fast mode. Additional groups of ten digits will be printed as they are completed.

5. You may repeat the continue mode as many times as you like, but the total number of groups of ten digits must not exceed 40.

6. For operation without a printer you can change the command at program step 087 from Nop to R/S. The program will stop to permit read-out of each group of ten digits as it is completed. Press R/S to proceed to the next group.

Program Listing:

```
000  92 RTN    040  69 OP     080  01  1     120  13  13    160  00  0     200  00  0
001  09  9     041  35  35    081  00  0     121  02  2     161  97 DSZ    201  76 LBL
002  00  0     042  69 OP     082  48 EXC    122  42 STO    162  07  07    202  11  A
003  42 STO    043  36  36    083  00  00    123  07  07    163  01  01    203  32 X:T
004  03  03    044  22 INV    084  48 EXC    124  05  5     164  26  26    204  69 OP
005  42 STO    045  74 SM*    085  09  09    125  00  0     165  09  9     205  00  00
006  05  05    046  06  06    086  99 PRT    126  42 STO    166  94 +/-    206  69 OP
007  05  5     047  73 RC*    087  68 NOP    127  06  06    167  61 GTO    207  05  05
008  00  0     048  05  05    088  69 OP     128  43 RCL    168  00  00    208  09  9
009  42 STO    049  22 INV    089  28  28    129  08  08    169  70  70    209  69 OP
010  04  04    050  74 SM*    090  97 DSZ    130  48 EXC    170  76 LBL    210  17  17
011  42 STO    051  06  06    091  02  02    131  03  03    171  13  C     211  47 CMS
012  06  06    052  73 RC*    092  00  00    132  29 CP     172  42 STO    212  32 X:T
013  43 RCL    053  06  06    093  96  96    133  22 INV    173  02  02    213  42 STO
014  08  08    054  77  GE    094  00  0     134  74 SM*    174  53  (     214  49  49
015  42 STO    055  00  00    095  81 RST    135  06  06    175  43 RCL    215  99 PRT
016  07  07    056  62  62    096  43 RCL    136  69 OP     176  01  01    216  92 RTN
017  94 +/-    057  43 RCL    097  07  07    137  36  36    177  85  +     217  42 STO
018  44 SUM    058  01  01    098  42 STO    138  32 X:T    178  01  1     218  02  02
019  03  03    059  74 SM*    099  03  03    139  74 SM*    179  01  1     219  01  1
020  44 SUM    060  06  06    100  01  1     140  06  06    180  93  .     220  00  0
021  04  04    061  01  1     101  00  0     141  53  (     181  07  7     221  42 STO
022  73 RC*    062  93  .     102  49 PRD    142  73 RC*    182  02  2     222  00  00
023  03  03    063  97 DSZ    103  09  09    143  06  06    183  54  )     223  22 INV
024  69 OP     064  07  07    104  69 OP     144  55  ÷     184  61 GTO    224  28 LOG
025  23  23    065  00  00    105  35  35    145  43 RCL    185  02  02    225  42 STO
026  32 X:T    066  40  40    106  64 PD*    146  01  01    186  39  39    226  01  01
027  73 RC*    067  69 OP     107  05  05    147  54  )     187  00  0     227  01  1
028  04  04    068  29  29    108  97 DSZ    148  53  (     188  00  0     228  42 STO
029  69 OP     069  02  2     109  03  03    149  59 INT    189  00  0     229  08  08
030  24  24    070  44 SUM    110  01  01    150  65  ×     190  00  0     230  42 STO
031  67  EQ    071  89  89    111  04  04    151  32 X:T    191  00  0     231  89  89
032  00  00    072  61 GTO    112  33 X²     152  43 RCL    192  00  0     232  69 OP
033  22  22    073  00  00    113  69 OP     153  01  01    193  00  0     233  05  05
034  22 INV    074  01  01    114  36  36    154  54  )     194  00  0     234  60 DEG
035  77  GE    075  97 DSZ    115  64 PD*    155  97 DSZ    195  00  0     235  04  4
036  00  00    076  00  00    116  06  06    156  03  03    196  00  0     236  05  5
037  75  75    077  00  00    117  97 DSZ    157  01  01    197  00  0     237  30 TAN
038  29 CP     078  96  96    118  07  07    158  33  33    198  00  0     238  33 X²
039  00  0     079  24 CE     119  01  01    159  09  9     199  00  0     239  86 STF
```

800 Digit Square Root - (cont)

Long Number Version: This program permits the user to enter many digit arguments. After entering the program below:

1. Enter a value equal to the number of digits in the arg ument divided by ten and rounded to the next greater integer and press A. The input value must be greater than one and less than or equal to 40.

2. Enter the argument in groups of ten digits and press R/S. The groups are printed. The argument must be entered from left to right with a decimal point after the leftmost digit of each group of ten. Pad the last group with zeroes at the right as required. When a flashing "1. 10" appears press 7 and then EE to start calculations in fast mode.

3. If a printer is being used each group of ten digits of the square root is printed as it is completed. The calculator stops with a zero in the display. The program will find the first 10n digits of the root of the argument where n = INT((# of digits + 10)/10. A sample printout appears at the right for one hundred digits of pi as the argument.

4. For operation without a printer you can change the command at program step 087 from Nop to R/S. The program will stop to permit read-out of each group of ten digits as it is completed. Press R/S to proceed to the next group.

5. Once program execution stops you can enter the baseline version and use the C option to find more digits of the argument. The total number of groups of ten digits can not exceed 40.

```
3.141592653
5.897932384
6.264338327
9.502884197
1.693993751
0.582097494
4.592307816
4.062862089
9.862803482
5.342117067

1772453850.
9055160272.
9816748334.
1145182797.
5494561223.
8712821380.
7789852911.
2845910321.
8137495065.
6738544663.
```

```
000  92 RTN   040  69 OP    080  01  1    120  33 X²    160  55  ÷    200  04  4
001  09  9    041  35  35   081  00  0    121  69 OP    161  43 RCL   201  09  9
002  00  0    042  69 OP    082  48 EXC   122  36  36   162  01  01   202  42 STO
003  42 STO   043  36  36   083  00  00   123  64 PD*   163  54  )    203  00  00
004  03  03   044  22 INV   084  48 EXC   124  06  06   164  53  (    204  32 X:T
005  42 STO   045  74 SM*   085  09  09   125  97 DSZ   165  59 INT   205  42 STO
006  05  05   046  06  06   086  99 PRT   126  07  07   166  65  ×    206  02  02
007  05  5    047  73 RC*   087  68 NOP   127  01  01   167  32 X:T   207  42 STO
008  00  0    048  05  05   088  69 OP    128  21  21   168  43 RCL   208  03  03
009  42 STO   049  22 INV   089  28  28   129  02  2    169  01  01   209  92 RTN
010  04  04   050  74 SM*   090  97 DSZ   130  42 STO   170  54  )    210  72 ST*
011  42 STO   051  06  06   091  02  02   131  07  07   171  97 DSZ   211  00  00
012  06  06   052  73 RC*   092  00  00   132  53  (    172  03  03   212  69 OP
013  43 RCL   053  06  06   093  96  96   133  05  5    173  01  01   213  20  20
014  08  08   054  77  GE   094  00  0    134  00  0    174  49  49   214  99 PRT
015  42 STO   055  00  00   095  81 RST   135  85  +    175  09  9    215  97 DSZ
016  07  07   056  62  62   096  69 OP    136  43 RCL   176  00  0    216  03  03
017  94 +/-   057  43 RCL   097  32  32   137  02  02   177  97 DSZ   217  02  02
018  44 SUM   058  01  01   098  43 RCL   138  42 STO   178  07  07   218  09  09
019  03  03   059  74 SM*   099  07  07   139  03  03   179  01  01   219  01  1
020  44 SUM   060  06  06   100  42 STO   140  54  )    180  41  41   220  00  0
021  04  04   061  01  1    101  03  03   141  42 STO   181  09  9    221  42 STO
022  73 RC*   062  93  .    102  43 RCL   142  06  06   182  94 +/-   222  00  00
023  03  03   063  97 DSZ   103  02  02   143  43 RCL   183  69 OP    223  22 INV
024  69 OP    064  07  07   104  44 SUM   144  08  08   184  22  22   224  28 LOG
025  23  23   065  00  00   105  06  06   145  44 SUM   185  61 GTO   225  42 STO
026  32 X:T   066  40  40   106  44 SUM   146  03  03   186  00  00   226  01  01
027  73 RC*   067  69 OP    107  07  07   147  29 CP    187  70  70   227  01  1
028  04  04   068  29  29   108  01  1    148  00  0    188  00  0    228  42 STO
029  69 OP    069  02  2    109  00  0    149  22 INV   189  76 LBL   229  08  08
030  24  24   070  44 SUM   110  49 PRD   150  74 SM*   190  11  A    230  42 STO
031  67  EQ   071  89  89   111  09  09   151  06  06   191  32 X:T   231  89  89
032  00  00   072  61 GTO   112  69 OP    152  69 OP    192  69 OP    232  69 OP
033  22  22   073  00  00   113  35  35   153  36  36   193  00  00   233  05  05
034  22 INV   074  01  01   114  64 PD*   154  32 X:T   194  69 OP    234  60 DEG
035  77  GE   075  97 DSZ   115  05  05   155  74 SM*   195  05  05   235  04  4
036  00  00   076  00  00   116  97 DSZ   156  06  06   196  09  9    236  05  5
037  75  75   077  00  00   117  03  03   157  53  (    197  69 OP    237  30 TAN
038  29 CP    078  96  96   118  01  01   158  73 RC*   198  17  17   238  33 X²
039  00  0    079  24 CE    119  12  12   159  06  06   199  47 CMS   239  86 STF
```

800 Digit Square Root - (cont)

800 Digit Extension: The program below, when used after the baseline version, extends the capability for finding square roots from 400 digits to 800 digits. The very last digit of the result, that is digit 2n with n the number of digits generated by the baseline version, may not be correct.

1. Use the baseline program to obtain at least half of the digits needed. Then load the program below.

2. Enter the number of additional groups of ten digits and press E'. The number of additional groups should not exceed the total number of groups that were generated with the A and C modes of the baseline version.

3. When a flashing "1. 10" appears press 7 and then EE to start calculations in fast mode.

4. The program will stop with every group of ten digits in the display. If a printer is connected it will print the group. The calculator will stop with a zero in the display when calculations are complete.

5. To generate additional groups of ten digits (the limitation in step 2 still applies) enter the number of additional groups and press E. When a flashing "1. 10" appears press 7 and then EE to resume calculations in fast mode.

Program Listing:

000	92	RTN	040	69	OP	080	00	00	120	53	(	160	00	0	200	00	0
001	09	9	041	35	35	081	48	EXC	121	73	RC*	161	00	0	201	00	0
002	00	0	042	69	OP	082	09	09	122	06	06	162	00	0	202	00	0
003	42	STO	043	36	36	083	99	PRT	123	55	÷	163	00	0	203	00	0
004	03	03	044	22	INV	084	91	R/S	124	43	RCL	164	00	0	204	00	0
005	42	STO	045	74	SM*	085	97	DSZ	125	01	01	165	00	0	205	00	0
006	05	05	046	06	06	086	02	02	126	54	)	166	00	0	206	00	0
007	05	5	047	73	RC*	087	00	00	127	53	(	167	00	0	207	00	0
008	00	0	048	05	05	088	91	91	128	59	INT	168	00	0	208	00	0
009	42	STO	049	22	INV	089	00	0	129	65	×	169	00	0	209	00	0
010	04	04	050	74	SM*	090	81	RST	130	32	X:T	170	00	0	210	00	0
011	42	STO	051	06	06	091	01	1	131	43	RCL	171	00	0	211	00	0
012	06	06	052	73	RC*	092	00	0	132	01	01	172	00	0	212	00	0
013	43	RCL	053	06	06	093	49	PRD	133	54	)	173	00	0	213	00	0
014	08	08	054	77	GE	094	09	09	134	97	DSZ	174	00	0	214	00	0
015	42	STO	055	00	00	095	69	OP	135	07	07	175	00	0	215	00	0
016	07	07	056	62	62	096	36	36	136	01	01	176	00	0	216	00	0
017	94	+/-	057	43	RCL	097	64	PD*	137	12	12	177	00	0	217	00	0
018	44	SUM	058	01	01	098	06	06	138	61	GTO	178	00	0	218	00	0
019	03	03	059	74	SM*	099	97	DSZ	139	00	00	179	00	0	219	00	0
020	44	SUM	060	06	06	100	07	07	140	01	01	180	00	0	220	00	0
021	04	04	061	01	1	101	00	00	141	00	0	181	00	0	221	76	LBL
022	73	RC*	062	93	.	102	95	95	142	00	0	182	00	0	222	10	E'
023	03	03	063	97	DSZ	103	05	5	143	00	0	183	00	0	223	69	OP
024	69	OP	064	07	07	104	00	0	144	00	0	184	00	0	224	38	38
025	23	23	065	00	00	105	42	STO	145	00	0	185	00	0	225	76	LBL
026	32	X:T	066	40	40	106	06	06	146	00	0	186	00	0	226	15	E
027	73	RC*	067	69	OP	107	43	RCL	147	00	0	187	00	0	227	42	STO
028	04	04	068	29	29	108	08	08	148	00	0	188	00	0	228	02	02
029	69	OP	069	61	GTO	109	48	EXC	149	00	0	189	00	0	229	53	(
030	24	24	070	00	00	110	07	07	150	00	0	190	00	0	230	43	RCL
031	67	EQ	071	01	01	111	29	CP	151	00	0	191	00	0	231	01	01
032	00	00	072	97	DSZ	112	22	INV	152	00	0	192	00	0	232	85	+
033	22	22	073	00	00	113	74	SM*	153	00	0	193	00	0	233	01	1
034	22	INV	074	00	00	114	06	06	154	00	0	194	00	0	234	01	1
035	77	GE	075	91	91	115	69	OP	155	00	0	195	00	0	235	93	.
036	00	00	076	24	CE	116	36	36	156	00	0	196	00	0	236	02	2
037	72	72	077	01	1	117	32	X:T	157	00	0	197	00	0	237	02	2
038	29	CP	078	00	0	118	74	SM*	158	00	0	198	00	0	238	54	)
039	00	0	079	48	EXC	119	06	06	159	00	0	199	00	0	239	86	STF

<u>FINDING PI</u> - L. Leeds. V10N3P4 described a method for finding pi which involved dividing 2143 by 22 and taking the square root two times. The result was correct to nine digits. Larry used his Model 100 to search for a fraction which would yield more correct digits with the fourth root technique, but did not find any. For a single square root technique he found three fractions which would give thirteen correct digits on the TI-59; 3044467/308469, 17007401/1723210, and 26140802/2648617. For a simple division he found three fractions which would give thirteen digits of pi on a TI-59; 4272943/1360120, 5419351/1725033, and 61905677/19705189. Of course, those are all much harder to remember than the old standby 355/113 which yields seven correct digits.

How does one mechanize a search for fractions which will work? One method which was used by Larry is to use a decimal to fraction converter, and compare the result to a preselected error. Another method is to simply test the decimal equivalent of fractions against the value of pi loaded to the accuracy of the computer. If the value of the fraction is greater than pi then the denominator is increased by one and the new fraction is tested. If the value of the fraction is less than pi then the numerator is increased by one and the new fraction is tested. The previous best solution may be saved for comparison with the newly generated result so that the program only prints improved solutions. Sample programs written for the CC-40 and HX-1000 are listed below.

```
100 PRINT "Decimal to Fraction":PAUS
E 2
110 INPUT "Allowable Error ? ";E
120 INPUT "Decimal Number ? ";N
150 A=1:B=1:C=1:D=1
160 IF 1>=N THEN 180
170 D=0:GOTO 190
180 A=0
190 F=A+C:G=B+D
200 P=F/G:T=ABS(P-N)
220 IF E>=T THEN 270
240 IF N>=P THEN 260
250 C=F:D=G:GOTO 190
260 A=F:B=G:GOTO 190
270 PRINT F;"/";G
280 PAUSE
290 GOTO 110
```

```
100 PRINT "Search for PI Fractions"
110 PAUSE 2
120 N=PI
130 OPEN #1,"10.s=0",OUTPUT
140 E=1
150 A=1:B=1:C=1:D=1
160 IF 1>=N THEN 180
170 D=0:GOTO 190
180 A=0
190 F=A+C:G=B+D
200 P=F/G:T=ABS(P-N)
220 IF E>=T THEN 270
240 IF N>=P THEN 260
250 C=F:D=G:GOTO 190
260 A=F:B=G:GOTO 190
270 PRINT #1,F;"/";G
280 E=.1*E
290 IF E>1.E-13 THEN 150
300 CLOSE #1
310 END
```

```
100 PRINT "PI-Finder":PAUSE 2
110 OPEN #1,"10.s=0".OUTPUT
120 P=3.14159265359
130 M=1:N=1
140 D=ABS(P-M/N)
150 IF ABS(P-M/N)<D THEN 180
160 IF M/N>P THEN N=N+1 ELSE M=M+1
170 GOTO 150
180 PRINT #1,STR$(M)&"/"&STR$(N);TAB
(22);
190 PRINT #1,USING 900,M/N
200 GOTO 140
900 IMAGE #.#############


2/1                   2.0000000000000
3/1                   3.0000000000000
13/4                  3.2500000000000
16/5                  3.2000000000000
19/6                  3.1666666666670
22/7                  3.1428571428570
```

<u>MORE ROBERT PRINS PROGRAMS</u> - The 86-2 edition of Programbiten contains two new programs by R. Prins: (1) Roots of a Polynomial using the Lin-Bairstow Method, and (2) a Histogram Generator. The instructions for both programs are in English. If you would like copies send a SASE plus an additional stamp to cover copying costs.

<u>FOR NEXT YEAR</u> - In 1986 we will continue to emphasize the TI-59 and the PC-100. We will also cover the TI-66 and PC-200, and the CC-40 and its peripherals, and will extend coverage to include the TI-74 when it becomes available. Coverage of older programmables such as the SR-52 and TI-57 will be minimized due to lack of interest. As in 1985 there will be four issues of about 24 to 28 pages each. I do not expect that I will be able to catch up in 1986 any more than I was able to in 1985, so at least one "1986" issue will probably not be distributed until early 1987. A subscription form is attached. If you decide not to continue I would appreciate a note to that effect.