TI PPC NOTES

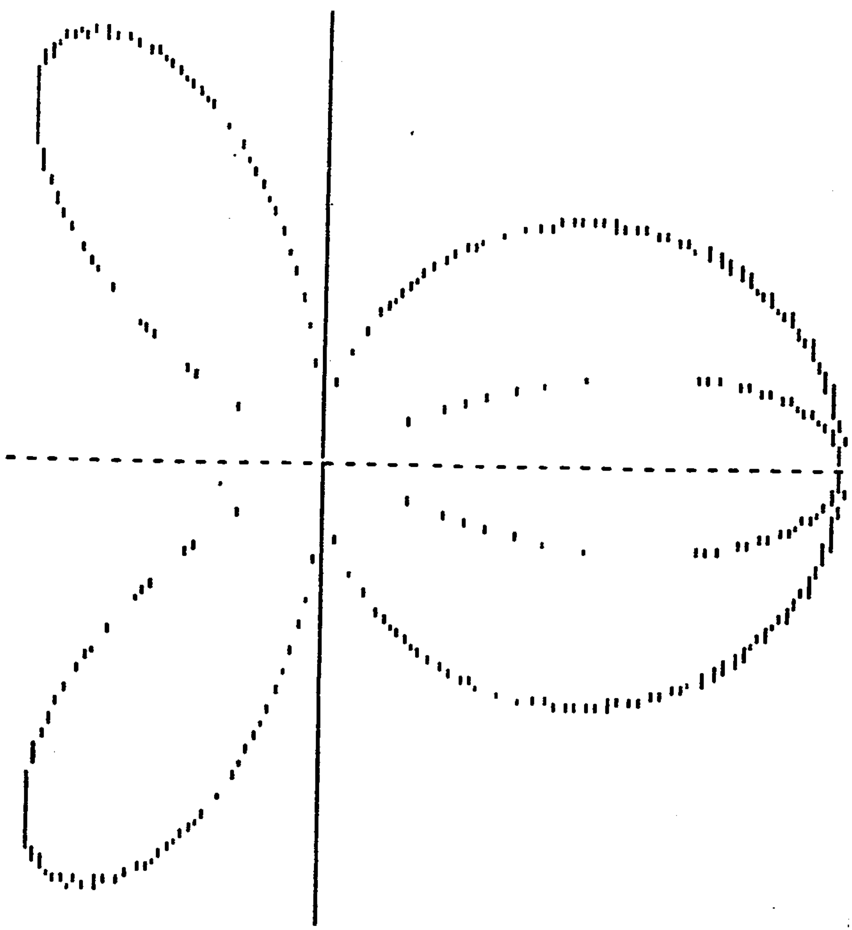
NEWSLETTER OF THE TI PROGRAMMABLE CALCULATOR CLUB

P.O. Box 1421, Largo, FL 34294

Volume 11, Number 1 First Quarter 1986

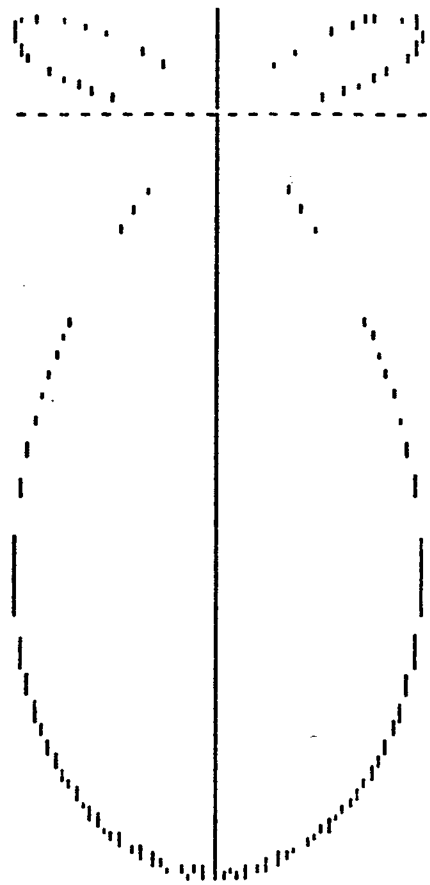
Fast Graphics Mode Polar Plotter - see page 18

Example 2



2. 1.

Example 4



1.

Sin wt & (1 - cos wt) on the fx-7000G

V10N4P7 presented a least squares program for decomposing input velocity data into sin wt and (1 - cos wt) components. The second part of that program was longer than needed since I used subscripts to maintain similarity to an existing Model 100 program. Changing to one letter variables reduces the program by 69 steps, where we note that C[1] is four steps, but the equivalent D is only one step. Use of the implied multiplication feature eliminates an additional 11 steps. The revised program is at the right.

```
Rad
0→D~H
I-1→J:Lbl 2
2πJK÷84→C
Sin C→A
1-Cos C→B
D+A^2→D
E+B^2→E
F+AB→F
G+AK[J]→G
H+BK[J]→H
Dsz J:Goto 2
DE-F^2→J
(GE-HF)÷J→A
(DH-FG)÷J→B
"ACC = ":A▲
"RATE = ":B▲
```

---

800 DIGITS SQUARE ROOT - V10N4P22-25 presented Robert Prins' program for finding 800 digit square roots. The instructions for the extension program on V10N4P25 failed to note that changing the R/S at step 084 to a Nop allows the program to print out the entire additional 400 digits without interruption. For those who may be interested, the printout for 800 digits of the square root of 3 is:

1732050807.	1531234326.	6679580638.	9106768231.
5688772935.	6903322886.	1835366611.	1992883756.
2744634150.	6506722546.	843173780.	4114142201.
5872366942.	6892183797.	8943783161.	6742752102.
8052538103.	1227047131.	208830552.	3729942708.
8062805580.	6603678615.	4901670023.	3105989845.
6979451933.	8801904998.	5207111442.	9475987664.
169088000.	6537379859.	8869599095.	2888977961.
3708114618.	3894676503.	6365797087.	4783795839.
6757248575.	4750657605.	1684980728.	228854852.
6756261414.	756618348.	9949329648.	9035760338.
1540670302.	1296061009.	4283020786.	5280806438.
9969945094.	4760218719.	4086039887.	1972344661.
9989524788.	325083145.	3869753758.	596897228.
1165551209.	8295239598.	2317317831.	7286526415.
4373648528.	3299778982.	3959929830.	3822664698.
932319023.	4508288714.	783870287.	4200211954.
558206797.	4638329173.	7053913369.	8415527844.
4820101084.	4722416398.	5633121037.	1181286534.
6749232650.	4587855397.	726401924.	5070351915.

TI-59 MATERIAL FOR SALE - Former member William Vogel has donated his TI-59 equipment and documentation to the club. Since most of the material duplicates material I already have I am offering it for sale for nominal prices:

TI-59, Master Library Module and Documentation	$ 50.00
Leisure Library Module and Documentation	15.00
Applied Statistics Module and Documentation	15.00
Math Utilities Module and Documentation	15.00
Three roll pack of old style printer paper	6.00

The TI-59 has some keys which have lost the original feel.

MAILBAG - As with previous years the comments received with the 1986 subscription forms provided a wide range of opinions as to what should or should not be included in TI PPC Notes. For example, one subscriber wrote:

> "Expanding the coverage of PPC Notes to cover the Casio fx-7000G is a good idea." E.T.

while another wrote:

> "Don't let the theme drift too far away from the TI products. The trend is very apparent and worrisome." R.R.

My intent for 1986 is to continue to provide strong coverage for the TI-59, TI-66 and CC-40. I will also provide limited coverage for non-TI devices such as the fx-7000G, the HP-15, etc., and will pick up coverage on any new TI devices which may appear such as the TI-74. Other reader comments included:

"I have not seen this excellent book mentioned in the 'Notes': Perspective Drawings by Programmable Calculator - A Method with Graphic Aids by David Yue. Van Nostrand Reinhold, Co. 1984. 232 pages. Cloth bound. It is profusely illustrated with much math, and many programs for both HP and TI-59." P.M.

"... Believe some short BASIC programs of general interest might be worthwhile; e.g., moving averages, sample statistics, 't' test, curve fitting, roots of equations, etc." R.B.

"Would like to see TI-59 programs converted into IBM compatible programs; also programs on AM radio antenna design." F. S.

"More reference to the TI-66." J.M.

"... I continue to think that magnetic cards give the TI-59 an enduring niche in on-site technical calculation, and I still use mine to complement two microcomputers. ..." R.E.

"It has been nearly 6 months since any CHUU bulletins have been sent out so you are not doing so badly. Info on CASIO was appreciated." R.K.

"... Tickled to hear of disk for CC-40. I already ordered one." L.K.

"... I enjoy the hardware and software availability articles." T.C.

"... Can any of the subscribers come up with a program for the PC which translates any TI-59 program to BASIC?" M.S.

"Considering the age of the TI-59 it would be useful to learn more about common machine failures, service hints, repair experiences, etc. R.S.

"If the TI-74 is for real, things should pick up. Now, more than ever, ways to convert TI-59 to BASIC seem to be desireable." R.S.

"More on Casio's 602 and 7000G is worthwhile. More on the TI-74 is mandatory. Tutorials are worthy things. ..." A.L.

"You're doing a great job with TI PPC Notes ..." J.F.

COMBINATORIAL ANALYSIS ON THE TI-66 - P. Hanson

In the discussion of the TI-55III in V10N4P13 I stated that I couldn't imagine why anyone would spend forty dollars for a TI-55II or TI-55III when with shopping one can obtain a TI-66 for about the same price. It turns out that if the user is doing probability problems that involve a lot of factorials, permutations, and combinations, then the built-in functions on the TI-55's provide a real advantage in speed. Consider one of the time-honored tests of speed on the TI-59, the calculation of 69!. Using ML-16 the TI-59 finds the value in about 14 seconds, but as explained in V5N8P4, only because the library programs run in the equivalent of fast mode. If ML-16 is down-loaded into user memory then the calculation requires about 28 seconds. By comparison, my TI-55II finds 69! in just five seconds. The TI-55II has similar speed advantages over the ML-16 on the TI-59 for the calculation of permutations and combinations. There are other hand-helds which are even faster. My HP-11 finds 69! in about a second. The clear winner so far for speed in a hand-held is the Casio fx-7000G where the answer for 69! seems to appear instantaneously when the EXE key is pressed.

What if I want to program a TI-66 to perform factorials, permutations and combinations? One solution would be to enter the 132 steps of the down-loaded version of the TI-59's ML-16 routine. A better solution would be to use the 70 step program listed at the right. The instructions are the same as for the use of ML-16 on the TI-59:

1. Enter n and press A.
2. Enter r where $0 \leq r \leq n$ and press B. This step is not needed for factorials.
3. Press C to find n factorial (n!).
4. Press D to find the number of possible permutations (nPr).
5. Press E to find the number of possible combinations (nCr).

where n and r must be re-entered for each calculation.

Equivalent programs for the CC-40 and the fx-7000G appear on page 5. For the CC-40 enter RUN 100 to calculate factorials, RUN 200 to calculate permutations, and RUN 300 to calculate combinations. Appropriate prompts and annotation of results are provided.

```
000  76 LBL
001  11  A
002  42 STO
003  00  00
004  91 R/S
005  76 LBL
006  12  B
007  42 STO
008  01  01
009  91 R/S
010  76 LBL
011  13  C
012  43 RCL
013  00  00
014  65  ×
015  97 DSZ
016  00  00
017  13  C
018  01  1
019  95  =
020  91 R/S
021  76 LBL
022  14  D
023  43 RCL
024  00  00
025  65  ×
026  69 OP
027  30  30
028  97 DSZ
029  01  01
030  14  D
031  01  1
032  95  =
033  91 R/S
034  76 LBL
035  15  E
036  43 RCL
037  01  01
038  32 X:T
039  01  1
040  42 STO
041  02  02
042  43 RCL
043  00  00
044  75  -
045  43 RCL
046  01  01
047  95  =
048  77  GE
049  30 TAN
050  42 STO
051  01  01
052  76 LBL
053  30 TAN
054  43 RCL
055  00  00
056  55  ÷
057  43 RCL
058  02  02
059  65  ×
060  69 OP
061  22  22
062  69 OP
063  30  30
064  97 DSZ
065  01  01
066  30 TAN
067  01  1
068  95  =
069  91 R/S
```

Combinatorial Analysis - (cont)

For the fx-7000G the upper program is for factorials, the middle program is for permutations, and the lower program is for combinations. Again, appropriate prompts and annotation are provided. Note that the programmed factorial program for the fx-7000G runs noticeably slower than the built in factorial function, but still is done in about a second. The CC-40 can calculate 84! without overflow due to a larger data range. By comparison, machines such as the Radio Shack Model 100 with a smaller data range can only find 48! with the methods illustrated here.

```
100 INPUT "N = ? ":N
110 M=1
120 FOR I=1 TO N
130 M=M*I
140 NEXT I
150 PRINT "N! = ":M
160 PAUSE
170 GOTO 100
200 INPUT "N = ? ":N
210 INPUT "R = ? ":R
230 M=1
240 FOR I=(N-R+1)TO N
250 M=M*I
260 NEXT I
270 PRINT "nPr = ":M
280 PAUSE
290 GOTO 200
300 INPUT "N = ? ":N
310 INPUT "R = ? ":R
320 M=1: J=1
330 IF N>2*R THEN 350
340 R=N-R
350 FOR I=(N-R+1)TO N
360 M=(M*I)/J
370 J=J+1
380 NEXT I
390 PRINT "nCr = ":M
400 PAUSE:GOTO 300
```

Execution times in seconds for three sample problems are listed below. The ML-16 results with the TI-59 are for the program running from the module. "TI-59 program" and "TI-66 program" means the program on page 4 of this issue.

Machine	Factorial (n=69)	Permutations (n=100 r=50)	Combinations (n=328 r=164)
TI-66 program	42	43	246
TI-59 ML-16	14	43	165
TI-59 program	27	29	167
TI-55II	5	3	24
HP-11	1	8	35
CC-40	1	1	8
fx-7000G	1	1	7

```
"N="?→A
1→B
Lbl 1
AB→B
Dsz A:Goto 1
"N! = ":B

"N="?→N
"R="?→R
1→A
Lbl 1
NA→A
N-1→N
Dsz R:Goto 1
"nPR=":A

"N="?→N
"R="?→R
1→B~C:R→A
N≥2R⇒Goto 1
N-R→A
Lbl 1
(CN)/B→C
N-1→N:B+1→B
Dsz A:Goto 1
"nCR=":C
```

where it is clear that the TI-55II has substantial advantages in speed over the best programs that I have been able to write for the TI-59 or TI-66 . For permutations and combinations the program on page 4 would have substantial execution time advantages over the ML-16 routine on the TI-59 if it was run in fast mode.

There is another advantage of the program on page 4 over ML-16. If you use ML-16 to find the number of combinations of 20 items taken 12 at a time you will see 125970 in the display, but the display register value is really 125969.9999995 . For 20 items taken 8 at a time the display will be 125970 but the display register will contain 125969.9999997 . You will find a similar inconsistency with the TI-55II.

Combinatorial Analysis - (cont)

Now we know that the answers for the two problems should be equal and should be integers. If you solve the same problems with the program on page 4 you will get exactly 125970 in either case. The problem with the ML-16 program is that it does not anticipate the round-off errors which will occur with the TI-59. Consider the formula for calculating combinations from page 103 of Spiegel's Theory and Problems of Statistics from the Schaum's Outline Series:

$$nCr = \frac{n(n-1)\cdots(n-r+1)}{r!} = \frac{n!}{r!(n-r)!}$$

One method for calculating combinations would be to simply calculate and combine the factorials as defined in the right hand expression. That would limit n to the largest factorial which does not overflow, 69! in the case of the TI-59. The use of the left hand expression will increase the range of input values. The range can be increased even more if the numerator and denominator are not calculated separately and then combined, but rather the terms from numerator and denominator are used alternately. For the case of 20 items taken 8 at a time:

$$nCr = \frac{20\cdot19\cdot18\cdot17\cdot16\cdot15\cdot14\cdot13}{8\cdot7\cdot6\cdot5\cdot4\cdot3\cdot2\cdot1}$$

we calculate 20 / 8 x 19 / 7 ····· 14 / 2 x 13 = 125969.9999997

which is the method used in ML-16. For the case of 20 items taken 12 at a time:

$$nCr = \frac{20\cdot19\cdot18\cdot17\cdot16\cdot15\cdot14\cdot13\cdot12\cdot11\cdot10\cdot9}{12\cdot11\cdot10\cdot9\cdot8\cdot7\cdot6\cdot5\cdot4\cdot3\cdot2\cdot1}$$ and we

calculate 20 / 12 x 19 / 11 ·······10 / 2 x 9 = 125969.9999995

where it is not surprising that the when the first solution (20C8) required about seven seconds, the second solution (20C12) required about eleven seconds. As either solution progresses decimal parts including round-off errors are generated whenever the intermediate solution is not exactly divisible by the next divisor.

One way to avoid the round-off errors is to reverse the order of the divisors; that is, for the 20 units 8 at a time problem the sequence of operations would be:

nCr = 20 / 1 x 19 / 2 x 18 / 3 ···· 13 / 8 = 125970 exactly.

If you think about this you will realize that a factor equal to the next divisor will have been accumulated in the intermediate solution before that divisor is used. The result is that the program deals only with integers, and round-off is not a problem. This technique was used in the programs on pages 4 and 5. There is a small penalty in range; for example, the program on page 4 overflows for the case of 330 units taken 165 at a time, while ML-16 does not overflow until 338 units taken 169 at a time.

Combinatorial Analysis - (cont)

There is another improvement over ML-16 in the program on page 4 for the calculation of combinations. On page 6 we noted that while the solutions for 20 items taken 8 at a time and for 20 items taken 12 at a time are the same, the execution times using ML-16 on the TI-59 are different. That situation becomes more severe as n gets larger; for example, 100 items taken 1 at a time yields the obvious 100 combinations in about a second, but 100 items taken 99 at a time also yields an answer of 100 combinations, but only after 98 seconds of run time. The solution to reducing execution time for the combination solution to a minimum is to find the minimum value of r and (n-r) and use that value as "r" in the solution. That technique was implemented in all the programs for finding combinations on pages 4 and 5.

MORE MEMBERSHIP LISTING - In response to several requests V10N4P9 listed the names and addresses for eight members who gave permission to publish that information. Eight additional members have agreed to publish similar information:

Dr. D. M. Graham
2149 Scarboro Avenue
Vancouver/B.C.
CANADA V5P 2L2
(604) 325-0253

Robert K. Leaman
215 W. Castlewood
Selma AL 36701

Robert Ericson
32 Ferncrest Blvd.
North Providence RI 02911

Myer Boland
66-E Overlook Way
Englishtown NJ 07726

Gordon Wilson
Box 967
Christiansted
St. Croix VI 00820

Robert Spaulding
550 Elinor Drive
Fullerton CA 92635

Joe Cox
P.O. Box 2210
Atlanta GA 30301

Dale Reddick
751 N. Indian Creek Drive
Apt. 117
Clarkston GA 30021

MORE ON PRINTER PAPER - Robert Lucas writes: "I verify Hewlett Ladd's observation on printer paper, but after 3 winter months, the characters are still faintly readable. Also, the paper cannot be printed upon by rerunning through the printer. Apparently, the ultra-violet of the sun has destroyed the underlying 'ink' which probably does not contain carbon. Heat from a match also does not create any effect. In another test, the new paper is still clearly readable after 2 years when tacked on my office wall, no direct sunlight, and exposed to reflected light and fluorescent lighting. Finally, 'Scotch Brand Magic' transluscent tape, the kind you can write on, lifts the paper surface coating away from the 'ink', turning the paper blank after several months, but the paper is still responsive to marking by pressure after the 2 years. It seems the old paper (my supply was grey looking, now all used up) contains carbon, which of course will not fade or bleach out."

MORE ON THE fx-7000G Ran# FUNCTION - P. Hanson.

```
1   Mcl:1000→Y
2   "DELAY="?→W
3   Lbl 1
4   W→Z
5   Lbl 2
6   Dsz Z:Goto 2
7   Ran#→X
8   V+X→V
9   1+A[10X]→A[10X]
10  Dsz Y:Goto 1
11  V÷1000
```

V10N4P13 noted that the mean value of a sequence of numbers obtained from the Ran# function of the fx-7000G was dependent upon the length of the program. The listing at the right is for one of the programs I used to investigate the function. I have added line numbers to aid in the explanation even though the machine does not use line numbers. Line 2 allows the user to select a variable delay to examine the effect of program cycle time on the random number output. Lines 4 through 6 generate the delay. Line 8 accumulates the sum of the random numbers generated. Line 9 sorts the random numbers using an "address sort" of the type illustrated in lines 800 through 895 of the CC-40 program on V10N3P15. The technique involves multiplying the value of the generated numbers by ten to obtain numbers with integer portions ranging from zero through 9. Those numbers are then sorted into one of ten "bins" by indirect addressing. Note that as with other machines the user does not need to change the indirect addresses to integers--the machine truncates the address for him. A more complete discussion of the technique appears in V10N1P24, which also notes that the technique does not work with a machine like the CC-40 which rounds rather than truncates the subscript. Some sample results for 1000 random numbers and various delay periods appear below. The columns headed by A through J show the contents of those memories, where memory A contains the number of random numbers less than 0.1, memory B the number of random numbers greater than or equal to 0.1 and less than 0.2, etc. Five solutions for the delay of 35 are included to illustrate the variations which might be expected in the distribution from test to test. Note that the Ran# function does not require a seed, and yields numbers which are different with each test.

Delay	Mean	A	B	C	D	E	F	G	H	I	J
1	0.448	56	98	146	168	161	131	109	63	27	41
5	0.478	53	63	137	145	183	110	116	81	42	70
10	0.527	98	64	65	115	97	114	130	121	99	97
15	0.507	97	83	84	93	126	110	94	94	106	113
20	0.520	120	93	100	66	53	84	102	161	128	93
25	0.515	99	109	95	93	83	75	89	132	121	104
30	0.544	91	86	94	64	107	76	75	131	134	142
35	0.383	406	172	68	18	9	1	6	25	36	259
	0.376	355	209	78	39	9	18	4	15	28	245
	0.383	370	191	74	29	14	6	10	24	35	247
	0.372	365	200	77	35	14	20	15	13	18	243
	0.375	369	200	89	29	9	8	4	15	27	250
40	0.428	8	28	226	244	218	223	42	6	4	1
45	0.529	23	11	49	161	214	210	195	87	37	13
50	0.531	64	73	95	104	129	92	116	133	106	88

The fx-7000G Ran# Function - (cont)

```
1  Range 0,10,1,0,400,100
2  Defm 10
3  Cls:Mcl
4  1000→A
5  "DELAY="?→B
6  Lbl 1
7  B→C
8  Lbl 2
9  Dsz C:Goto 2
10 10Ran#+1→X
11 1+Z[X]→Z[X]
12 Dsz A:Goto 1
13 Prog 1

1  Graph Y=
```

The program on page 8 served two purposes: examining the effect of program length on the characteristics of the Ran# output, and demonstrating that the sorting technique which relies on the machine to truncate subscripts works with the fx-7000G. However, that program isn't a very good demonstration of the fx-7000G.

My first improved program appears at the upper right. Line 1 sets the range values for a histogram. Line 2 expands the memory to include ten additional memories to be used by the histogram function. Line 5 enters the delay value. Lines 7 through 9 provide the variable delay function. Lines 10 and 11 provide the sorting into Z[1] through Z[10] for use by the histogram routine. Since this program uses the Dsz command it cannot run in the SD2 mode needed for histogram plotting (see page 110 of the manual). Line 13 provides transfer to another program where the SD2 mode has been set for a single line program where the Graph Y= command in SD2 mode provides a histogram plot.

```
1  Range 0,1,.1,0,400,100
2  Defm 10
3  Cls:Scl
4  "DELAY="?→Z
5  Lbl 1
6  0→Y
7  Lbl 2
8  1+Y→Y
9  Y≠Z⇒Goto 2
10 Ran#DT
11 W≠1000⇒Goto 1
12 Graph Y=◢
13 "x̄=":x̄◢
14 1→X
15 Lbl 3
16 Z[X]◢
17 1+X→X
18 Goto 3
```

A second iteration makes even better use of the capabilities of the fx-7000G. The listing appears at the right. The program runs in the SD2 mode which must be set before the first instruction is entered (see page 109 of the manual). Line 1 sets the size of the cells for the histogram accumulation. By using a cell width of 0.1 there will be no need to rely on a user generated sort routine. Lines 7 through 9 provide the delay loop, but since the program is in SD2 mode the Dsz instruction cannot be used. The DT function in line 10 provides sorting and the calculation of sums needed for statistics solutions. Line 12 displays the histogram. Line 13 displays the mean value as found by the built-in statistics functions. Lines 14 through 18 provide display of the contents of the ten cells of the histogram. When the EXE key is pressed one more time after the contents of the last histogram cell have been read out the display reads "Mem ERROR Step P1-93" which indicates that contents of all the cells have been displayed.

CLEARING THE fx-7000G STATISTICS REGISTERS - The Scl (shift AC) function clears the registers used in statistics calculations in a manner similar to that provided by the 2nd-Pgm-01-SBR-CLR sequence on the TI-59. The major difference is that the registers which are cleared depend upon the calculation mode:

COMP	Scl does not work.
SD1	Clears memories U, V, and W.
SD2	Clears memories U, V, W, and Z(1) through Z(n), where n was set by the Defm mode.
LR1 or LR2	Clear memories P, Q, R, U, V, and W.

FAILURES, SERVICE HINTS, REPAIRS, ETC. - One of the comments received with the 1986 subscription forms was that it would be useful to learn more about common machine failures, repair techniques, and the like. Numerous articles on those subjects have appeared in TI PPC Notes, but have been scattered through many issues. I have probably been exposed to more malfunctions than the average member. My employer, Honeywell, participated in the TI Professional Productivity Program (see the November 1978 issue of PPX Exchange), and a large number of employees were provided with TI-59's. I gradually attained a reputation as a TI-59 expert and was consulted on a wide variety of malfunctions. One of the worst was a calculator which would develop an anomaly in the sine calculation, but only after warming up for about thirty minutes. The problem was particularly hard to identify since the algorithm did not go completely bad; the sine of 30 degrees would be returned as something slightly different from 0.5. General commments on malfunction diagnosis were presented on V9N4P10. For the benefit of newcomers the various articles which have appeared are summarized below:

Calculator Diagnostics

Some diagnostic capability is provided with the TI-59. Personal Programming discusses the various capabilities on pages VII-9 and in Appendix A. Page 43 of the Programmable TI-58/59 Service Manual describes an additional read/write test which stores 1/9 (all ones) into each memory, sums -1/9 into each memory register, checks the contents of each register for zero, and prints out the contents and register number for each failure. The test takes about three minutes.

As users gained experience with their machines they recognized the need for more exhaustive diagnostics. PPX 908119 by Bill Skillman allowed the user to load the same data into different banks and compare the contents as read by the program. V5N2P9 described a diagnostic developed by Maurice Swinnen which involved placing a code 41 (SST) at each location. If you then press CLR RST SST the program counter will move sequentially through each step, stopping at step 959. A successful test takes about ten seconds. If there is a bad data register such that something other than code 41 is read then the calculator will stop, and pressing BST LRN will reveal the bad location; however, V5N8P16 reported that if somehow a code 31 (LRN) was erroneously returned then the program will not stop. There is an easy way to enter a string of 41's described V5N6P4. V5N6P3 reported a calculator problem which passed all of the diagnostic tests recommended by TI including the one from the TI-58/59 Service Manual, and even passed the Magnetic Card Comparator Test (PPX 908119). The SST test stopped when it read 04 at step 793 instead of 41; however, V7N3P10 reported a memory malfunction which was not diagnosed by the SST program, but was diagnosed by the test program from the TI-58/59 Service Manual.

Those malfunctions illustrate one of curiosities of reading from memory--sometimes one bit pattern will be read successfully, while another pattern will not. With some calculators the memory will be read correctly some of the time, and incorrectly other times. To provide a variation in bit patterns I modified the program from the TI-58/59 Service Manual to exercise nine different patterns (PPX 908175). The execution time is long, about sixteen minutes, but I have yet to encounter a memory register malfunction which was not diagnosed by that program. Maurice's SST program provides the means for diagnosing memory problems in program steps 000 through 159.

Failures, Service Hints, Repairs, etc. (cont)

Print Head Cleaning

The PC-100 manuals described a print head cleaning technique for use in clearing foreign particles which may have collected on the print heads. The cleaning cards which were supplied with early packs of printer paper stated that cleaning should be performed before installing each new roll of paper. Page VI-12 of Personal Programming provided a program which would exercise the heads; however, users found that alternate routines would provide improved cleaning by minimizing the delay between printing. It was believed that this produced the maximum possible heat to burn off the tiny shreds of paper sticking to the heads. Routines were published in V5N1P10 and in V5N3P3. I prefer Thomas Wysmuller's routine from V5N3P3 which is reproduced at the right. To use the program press A and see I's printed. Press R/S and then B and see II's printed. Press R/S and then C and see square O's printed. Press R/S and then D and see 8's printed.

```
000  69 OP
001  05  05
002  81 RST
003  76 LBL
004  14  D
005  01  1
006  01  1
007  76 LBL
008  15  E
009  65  X
010  01  1
011  00  0
012  01  1
013  00  0
014  01  1
015  00  0
016  01  1
017  00  0
018  01  1
019  95  =
020  69 OP
021  01  01
022  69 OP
023  02  02
024  69 OP
025  03  03
026  69 OP
027  04  04
028  81 RST
029  76 LBL
030  11  A
031  02  2
032  04  4
033  15  E
034  76 LBL
035  12  B
036  07  7
037  04  4
038  15  E
039  76 LBL
040  13  C
041  03  3
042  02  2
043  15  E
```

V9N2P13 reported that I had tried the various published print head cleaning techniques on one printer without success. Then I cut a piece of crocus cloth to the width of the printer paper, passed it through the printer using the ADV key, and cleared the print head problems. Maurice Swinnen reported that others have had success in print head cleaning using a strip of the material used for polishing defects in Plexiglas, and that HP sells a cleaning card for use with their calculators.

Batteries

V6N8P14 described Bob Fruit's experience running his calculator from a six volt lantern battery in an emergency situation. V7N6P10 discussed the nature of failures in the Nickel-Cadmium batteries used in the battery pack, and provided a circuit which could be used in clearing the internal shorts. The discussion also points out that calculations will be typically be correct and the display somewhat dim if two of the three cells are operational; but with only two cells operational reading and writing of magnetic cards will be erratic.

Keyboard Problems

An article in V7N3P4/5/6 describes methods for taking apart the calculator and cleaning the keys, and includes the suggestion for reversing the foam under the keys to restore the original keyboard "feel".

Contact Problems

A "Mailbag" item in V7N7/8P2 described successful repair of a malfunctioning printer by the old technique of scrubbing the contacts with a pencil eraser. I have had similar success with a malfunctioning memory module.

Failures, Service Hints, Repairs, etc. (cont)

Improper Paper Advance with the PC-100

Some PC-100's develop an inability to properly advance the printer paper. You can often provide proper paper advance by lifting the top cover and pressing on the metal support which holds the print head against the paper, but that's tiresome. You can make a temporary repair by cutting a wedge from an eraser, the soft white ones seem to work best, and pressing the wedge between the metal support and the printer case until proper paper advance occurs. Of course, you will have to do that all over again each time you install a new roll of paper. This temporary repair can become semi-permanent. There are two PC-100's at Honeywell which have been operating satisfactorily in this mode for years.

Magnetic Card Reading Problems

From the very beginning TI-59 users seem to have encountered read/write problems with the magnetic card reader. I was one of the fortunate ones who didn't. If I had encountered inconsistent read/write capability for the first few years it would have been practically impossible to develop the old "load-and-go" method for fast mode entry.

Some problems seem to be with contaminated read/write mechanisms. Other problems seem to be with contaminated cards. TI provided two card reader cleaning cards with each TI-59. The head-cleaning card contains an abrasive and can be used to remove buildup of oxide or foreign material on the read/write head. TI cautions that this card should be used sparingly. I have found very little use for this card; only in one instance did use of the card definitely improve read/write characteristics.

The drive roller cleaning card seems to be no more than a piece of cardboard. Page VII-8 of Personal Programming recommends use of the card "about every 500 reads, or when a card begins to slip or move at a non-uniform pace through the calculator". My experience has been that the use of the card doesn't seem to help; furthermore, the card deteriorates rapidly to a condition where it is simply unuseable.

V8N5P11 reported correspondence between George Thomson and I in early 1983 on the subject of card reading and writing. I had inadvertently discovered that saliva seemed to be the best solvent for cleaning cards to temporarily improve reading and writing. George tried the same technique, using what he dubbed "Hanson's Universal Solvent", and was able to obtain more successful read/write operation. In June of 1983 I obtained a sample of the CCL-144 cleaning strip. I tried it in my TI-59 and was immediately rewarded with trouble free read/write operation. Others have had similar success with the CCL-144. In V9N2P13 Paul Sperry reported that he had identified the solvent used to saturate the CCL-144 as alcohol. He also reported that a brochure from Texwipe indicated that a solution of 91% isopropyl alcohol and 9% ionized water was useful for cleaning magnetic components. Paul and others have suggested that the cleaning strips may be reused by providing your own solvent.

The CCL-144 cleaning strips are still available at twelve dollars for a box of ten from CMPI, Inc., 7308 Aspen Lane #123, Minneapolis MN 55428 (That is a new address). I will also supply sample cleaning strips at a price of two dollars each.

Failures, Service Hints, Repairs, etc. (cont)

Adjusting Magnetic Card Reading Speed

Even with properly cleaned read heads and drive rollers some magnetic cards written by one calculator simply cannot be read by another calculator. The November 1977 issue of PPX Exchange stated:

> "Cards written on one TI-59 may not read on another machine. This incompatibility is attributed to the difference in machine signatures, which is the result of increased magnetic card density. (The density of the TI-59 card is twice that of the SR-52.)"

My experience has been that the cards from the great majority of TI-59's can be read by other machines. One proof of this is the successful magnetic card service provided by TI PPC Notes. There is an occasional machine which will not read cards from most other machines, and which I have generally found can also not write cards which are readable by other machines. I tend to believe that such a machine is a "maverick". V5N2P3 described a technique for adjusting the reading speed devised by Peter Poloczek which was intended to allow adjustment of the card drive to provide compatibility with other TI-59's. I have not tried that procedure since I haven't had a calculator that wasn't compatible with the great majority of other units. I have heavily used three different units and use the cards interchangeably.

References:

1. The Programmable TI58/59 Service Manual was published by TI and was available for $11.95 plus $1.50 for handling and postage.

2. PPX 908119, "Magnetic Card Comparator", by Bill Skillman, 239 program steps, 6 pages.

3. PPX 908175, "Memory Malfunction Diagnostic", by Palmer Hanson, 160 program steps, 13 pages. Includes listings for both the "598-TEST-1" program from the TI-58/59 Service Manual and for the extended diagnostic program.

4. "Memory Diagnostic Using SST Commands", by Palmer Hanson, 240 steps, 8 pages. Includes instructions for special techniques for easily making a card with all 41's.

I will loan my copy of the Service manual to club members. Send two dollars to cover postage. I will send the manual first class and expect you to return it the same way. I will also provide copies of the documentation for the three diagnostic programs for two dollars each. Magnetic cards for the programs are available for the usual price of one dollar per card plus a stamped and self addressed envelope.

DO YOU BELIEVE THIS? The article "Hunting for the Ultimate Tip" on page 49 of the June 2 issue of Newsweek discusses a new programmable printer barely larger than a telephone and notes that it

> "... doesn't require possession of a personal computer--a key selling tool at a time when so many people think it takes an engineering degree to use one. Major brokerage houses are realizing that the number of people who invest at home via a personal computer is not growing as fast as expected. ..."

ANATOMY OF A PROGRAM FOR e - Hewlett Ladd. I enjoyed reading the two recent articles in TI PPC Notes containing programs programs for calculating e to many decimal places (V9N4P24/25 and V9N5P4). The last article was a particularly fine example of how experienced programmers adapt familiar algorithms to calculator operations with such disarming simplicity, and set me to wondering once more how they are able to do it. Despite the fact that I had copied and run earlier programs for many digits of n!, pi, etc., I never fully grasped the principles of extended precision. Bjorn's neat little program (V9N5P4) provided the spark I needed which has prompted me to write down the various stages I went through before I completely understood its finer points. I also wanted to get a program to print out a preselected number of digits, rather thhan just 980 of them. Having at last done both, herewith a personal chronicle of e to Programbiten's e as viewed by an amateur. The familiar infinite series

e = 1 + 1/1! + 1/2! + 1/3! + ··· 1/n!

or with the terms in the reverse order

e = 1/n! + 1/(n-1)! + 1/(n-2)! + ··· 1/1! + 1

may also be described in the form of a continued fraction expansion as

e = (1/n + 1) x 1/(n-1) + 1) x 1/(n-2) + 1 x ···1) + 1

which greatly simplifies the task of handling otherwise unwieldly factorials. A nineteen step program for the TI-58/59/66 which will yield ten digit accuracy using this less familiar, continued fraction version

1/12 + 1) x 1/11 + 1) x ···)1 + 1 = 2.718281828

appears at the upper right. Enter 12 and press A to obtain the ten digit printout. This works fine for 10 digit accuracy, but the use of the 1/x function will pose a problem for extended precision, so it would be better to substitute a divide for the 1/x in a stepping-stone development of an extended precision program. The modified program is at the center right. Again, enter 12 and press A. The answer will appear in the display after about eight seconds.

The next step in the development demonstrates the use of expanded partitioning, and the use of a numerator of 1,000,000,000 instead of 1 during the accumulation. This will be needed in the extended precision routines. The listing is at the lower right. The step at 035 gets the decimal point in the right place for the printout. The instructions are changed to enter (n+1) instead of n. Enter 13 and press A to obtain the solution in about 13 seconds.

```
000  76 LBL
001  11  A
002  42 STO
003  00  00
004  43 RCL
005  00  00
006  35 1/X
007  54  )
008  85  +
009  01  1
010  54  )
011  65  x
012  97 DSZ
013  00  00
014  00  00
015  04  04
016  01  1
017  95  =
018  99 PRT
019  91 R/S
```

```
000  76 LBL
001  11  A
002  42 STO
003  01  01
004  01  1
005  42 STO
006  00  00
007  53  (
008  24 CE
009  55  ÷
010  43 RCL
011  01  01
012  85  +
013  43 RCL
014  00  00
015  54  )
016  97 DSZ
017  01  01
018  00  00
019  07  07
020  99 PRT
021  91 R/S
```

```
000  76 LBL
001  11  A
002  32 X:T
003  47 CMS
004  01  1
005  00  0
006  69 OP
007  17  17
008  32 X:T
009  42 STO
010  99  99
011  01  1
012  52 EE
013  09  9
014  22 INV
015  52 EE
016  42 STO
017  00  00
018  22 INV
019  97 DSZ
020  99  99
021  00  00
022  35  35
023  53  (
024  24 CE
025  55  ÷
026  43 RCL
027  99  99
028  85  +
029  43 RCL
030  00  00
031  54  )
032  61 GTO
033  00  00
034  18  18
035  55  ÷
036  43 RCL
037  00  00
038  95  =
039  99 PRT
040  91 R/S
```

Anatomy of a Program for e - (cont)

So far only two registers have been used, R00 and R99, and the results have been left in the display register. Somehow, we must extend the accuracy of the computation into blocks of say ten digits. Before doing so, we will continue with the ten most significant digits (MSD's) of e, but will store them in R01 using R98 as a pointer, leaving 97 data registers free for later storage of up to 970 digits.

The program appears at the right. Step 012 allows the user to select more than one block of digits for the solution. Steps 030 through 047 are similar to the previous sequence except that more than one data register can be used. Steps 056 through 063 provide printout of all the data registers used. The user instructions remain essentially the same: enter 13 and press A; enter 1 for one block of ten digits and press R/S for the solution and printout. A solution requires about 22 seconds. The EXC 00, ST* 00 and Dsz 0 at steps 028, 038, and 040 will later handle the storage of extended precision results from as many as 97 preselected data registers. For example, if you were to enter a 3 for three blocks of ten digits instead of a 1 with this program you would get a three block output, but since the computation has been carried to only ten digit accuracy the two extra blocks (R03, R02, and R01 in that order) would be meaningless. Not much progress, you might say, but in fact the program is at last ready to accept extended precision with the addition of a few more strategically placed steps.

```
000  76 LBL     032  85  +
001  11  A      033  73 RC*
002  32 X:T     034  00  00
003  47 CMS     035  55  -
004  01  1      036  43 RCL
005  00  0      037  99  99
006  69 OP      038  54  )
007  17  17     039  72 ST*
008  32 X:T     040  00  00
009  42 STO     041  97 DSZ
010  99  99     042  00  00
011  91 R/S     043  00  00
012  42 STO     044  30  30
013  98  98     045  61 GTO
014  01  1      046  00  00
015  52 EE      047  14  14
016  09  9      048  22 INV
017  22 INV     049  64 PD*
018  52 EE      050  98  98
019  74 SM*     051  43 RCL
020  98  98     052  98  98
021  22 INV     053  42 STO
022  97 DSZ     054  00  00
023  99  99     055  98 ADV
024  00  00     056  73 RC*
025  48  48     057  00  00
026  43 RCL     058  99 PRT
027  98  98     059  97 DSZ
028  48 EXC     060  00  00
029  00  00     061  00  00
030  53  (      062  56  56
031  24 CE      063  91 R/S
```

But, first we must define an extended precision divide for the 1/n calculation. Bjorn's Programbiten version reported by Robert Prins started off with 449 for n, which was compatible with a 980 digit solution. Without a programmable calculator we might calculate in schoolboy fashion:

```
                     q1          q2          q3          q4

                  2227171:4922048997:7728285077:9510022271: ...
           449  / 1000000000:0000000000:0000000000:0000000000: ...

n x q1        - 999999779
r                    221 0000000000
n x q2              -220 9999999653
r                               347 0000000000
n x q3                         -346 9999999573
r                                          427 0000000000
n x q4                                    -426 9999999679
r                                                     321
```

By marking off the dividend into blocks of ten digits the block remainder r will always contain no more than three digits since the divisor will contain no more than three digits. Each succeeding trial dividend will therfore always contain thirteen or fewer digits, or e EE 10 plus the next ten digits of dividend.

Anatomy of a Program for e - (cont)

It will help to note that a program sequence such as r (EE 10 . . . without a CE immediately after the parenthesis operates like r (CE EE . . . It converts the r before the parenthesis to r EE 10 within the parentheses. If r is in EE format, which it will be after the first divide, the EE 10 must be preceded by INV EE as in the sequence

r (INV EE EE 10 . . .

With this in mind we imitate the schoolboy arithmetic in the program which appears at the right. Try it with 449 and press A; then enter 4 and press R/S to obtain a five line printout

2227171.
4922048997.
7728285077.
9510022271.
-321.

which includes the quotients and remainder from the schoolboy example on page 15.

000	76	LBL	035	24	CE
001	11	A	036	55	÷
002	32	X:T	037	43	RCL
003	01	1	038	99	99
004	00	0	039	54	)
005	69	OP	040	59	INT
006	17	17	041	72	ST*
007	32	X:T	042	00	00
008	42	STO	043	65	×
009	99	99	044	43	RCL
010	91	R/S	045	99	99
011	42	STO	046	54	)
012	98	98	047	97	DSZ
013	01	1	048	00	00
014	52	EE	049	00	00
015	09	9	050	24	24
016	22	INV	051	32	X:T
017	52	EE	052	22	INV
018	74	SM*	053	52	EE
019	98	98	054	43	RCL
020	43	RCL	055	98	98
021	98	98	056	42	STO
022	48	EXC	057	00	00
023	00	00	058	73	RC*
024	53	(	059	00	00
025	22	INV	060	99	PRT
026	52	EE	061	97	DSZ
027	52	EE	062	00	00
028	01	1	063	00	00
029	00	0	064	58	58
030	85	+	065	32	X:T
031	73	RC*	066	94	+/-
032	00	00	067	99	PRT
033	75	-	068	91	R/S
034	53	(			

Bjorn's program neatly repeated this routine with 448, 447, etc. We can accomplish the same by adding a few steps from the program at the right to the last one for e on page 15.

The rest of the program from page 15 is left intact except for making the necessary changes in direct addresses. If a fast mode routine is added the program for e with an option to run with less than 970 digits becomes:

000	91	R/S	020	52	EE	040	54	)	060	00	00				140	00	0
001	25	CLR	021	52	EE	041	97	DSZ	061	00	00				141	76	LBL
002	01	1	022	01	1	042	00	00	062	56	56				142	11	A
003	52	EE	023	00	0	043	00	00	063	00	0			0	143	32	X:T
004	09	9	024	85	+	044	18	18	064	81	RST			0	144	47	CMS
005	22	INV	025	73	RC*	045	61	GTO	065	00	0		J0	0	145	01	1
006	52	EE	026	00	00	046	00	00	066	00	0		00	0	146	00	0
007	74	SM*	027	75	-	047	02	02	067	00	0	7	00	0	147	69	OP
008	98	98	028	53	(	048	22	INV	068	00	0	128	00	0	148	17	17
009	22	INV	029	24	CE	049	64	PD*	069	00	0	129	00	0	149	32	X:T
010	97	DSZ	030	55	÷	050	98	98	070	00	0	130	00	0	150	42	STO
011	99	99	031	43	RCL	051	43	RCL	071	00	0	131	00	0	151	99	99
012	00	00	032	99	99	052	98	98	072	00	0	132	00	0	152	91	R/S
013	48	48	033	54	)	053	42	STO	073	00	0	133	00	0	153	42	STO
014	43	RCL	034	59	INT	054	00	00	074	00		134	00	0	154	98	98
015	98	98	035	72	ST*	055	98	ADV	075	0		135	00	0	155	04	4
016	48	EXC	036	00	00	056	73	RC*	076			136	00	0	156	05	5
017	00	00	037	65	×	057	00	00	077			137	00	0	157	30	TAN
018	53	(	038	43	RCL	058	99	PRT	07			138	00	0	158	33	X^2
019	22	INV	039	99	99	059	97	DSZ				139	00	0	159	86	STF

As a test you can enter 29 (n+1 for 28 terms) and press A, and then enter 3 for 30 digits (3 banks of 30 digits) and press R/S to see a flashing "1." in the display. Press 7 and then EE. In about 68 seconds you will get a 3 bank printout of e correct to 30 digits.

2.718281828
4590452353.
6028747135.

Anatomy of a Program for e - (cont)

The major deficiency of the program so far is the requirement that the user determine the number of terms needed to provide the desired number of correct digits. Through the use of one of the approximations for n!, say Stirling's formula or one of the logarithmic calculations such as in the May 1979 issue of PPX Exchange, one can make a table of the exponent of 1/n! versus n. A least squares fit yields the coefficients for a second degree polynomial approximation. A little fine tuning yields a concise polynomial which ensures that enough terms are always used to obtain the desired accuracy. The penalty for the tuning is that in some cases a few more terms will be used than are actually needed, extending the execution time. The worst case is for a single block of digits, where the formula selects 24 terms when only 13 are needed, more than doubling the execution time. But then you wouldn't use this routine to find ten digits of e anyway. A front-end calculation was also added to make sure that all of the digits in the last block will be correct. The resulting program is:

000	91	R/S	027	75	-	054	00	00	081	00	0	108	00	0	135	09	9
001	25	CLR	028	53	(	055	98	ADV	082	00	0	109	00	0	136	85	+
002	01	1	029	24	CE	056	73	RC*	083	00	0	110	00	0	137	05	5
003	52	EE	030	55	÷	057	00	00	084	00	0	111	00	0	138	93	.
004	09	9	031	43	RCL	058	99	PRT	085	00	0	112	00	0	139	04	4
005	22	INV	032	99	99	059	97	DSZ	086	00	0	113	00	0	140	65	×
006	52	EE	033	54	)	060	00	00	087	00	0	114	00	0	141	43	RCL
007	74	SM*	034	59	INT	061	00	00	088	00	0	115	76	LBL	142	98	98
008	98	98	035	72	ST*	062	56	56	089	00	0	116	11	A	143	75	-
009	22	INV	036	00	00	063	00	0	090	00	0	117	85	+	144	93	.
010	97	DSZ	037	65	×	064	81	RST	091	00	0	118	09	9	145	00	0
011	99	99	038	43	RCL	065	00	0	092	00	0	119	95	=	146	01	1
012	00	00	039	99	99	066	00	0	093	00	0	120	55	÷	147	65	×
013	48	48	040	54	)	067	00	0	094	00	0	121	01	1	148	43	RCL
014	43	RCL	041	97	DSZ	068	00	0	095	00	0	122	00	0	149	98	98
015	98	98	042	00	00	069	00	0	096	00	0	123	95	=	150	33	X²
016	48	EXC	043	00	00	070	00	0	097	00	0	124	59	INT	151	95	=
017	00	00	044	18	18	071	00	0	098	00	0	125	32	X:T	152	59	INT
018	53	(	045	61	GTO	072	00	0	099	00	0	126	01	1	153	42	STO
019	22	INV	046	00	00	073	00	0	100	00	0	127	00	0	154	99	99
020	52	EE	047	02	02	074	00	0	101	00	0	128	69	OP	155	04	4
021	52	EE	048	22	INV	075	00	0	102	00	0	129	17	17	156	05	5
022	01	1	049	64	PD*	076	00	0	103	00	0	130	47	CMS	157	30	TAN
023	00	0	050	98	98	077	00	0	104	00	0	131	32	X:T	158	33	X²
024	85	+	051	43	RCL	078	00	0	105	00	0	132	42	STO	159	86	STF
025	73	RC*	052	98	98	079	00	0	106	00	0	133	98	98			
026	00	00	053	42	STO	080	00	0	107	00	0	134	01	1			

The user enters the number of digits desired, presses A, and sees a flashing "1." in the display. He then presses 7 and then EE and waits for the printout, where all the printed digits will be correct.

ANOTHER TI-59 QUIRK - Robert Prins found a TI-59 quirk which does not seem to have been reported in TI PPC Notes while scanning old issues of Programbiten. On page 5 of the 82-1 issue Markus S. Markusson noted that an SST (code 41) followed by DMs, P/R, etc., acts as an R/S, but gave no further explanation. My tests show that the firmware functions P/R, DMs, , and x yield that result, where the calculator stops at the location folowing the firmware function. The inverses of those functions and the statistics functions Op 11 through Op 15 do not cause the stopping. No use for this quirk has been identified. I presume that Mr. Markusson is the same individual who provided the very fast 13 digit register listing program published in the May/June 1982 issue of PPX Exchange. Newcomers may wonder at all this interest in quirks which have no apparent practical use. The best answer is that the very useful fast mode and high resolution graphics techniques were once only "quirks".

<u>FAST GRAPHICS MODE POLAR PLOTTER</u> - Robert Prins. V10N2P12 presented a program for plotting functions defined by polar coordinates. At that time Robert reported that he was working on a fast mode, high resolution version of the program. Here it is. The program is split up into two parts, one for the input of data and one for the actual printing/plotting. The program listings appear on page 21. Here are the instructions:

1. Enter the data-entry program

At this point you can choose one of these two options:

a. Input of all variables, X-min, X-max, Y-min, Y-max, number of tapes and number of rotations. Press A for this option and answer all prompts with the entry of the value of the requested variable, followed by an R/S.

b. Let the program set X/Y-min/max to +/- 1.044 and only enter number of tapes and number of rotations. Press B for this option and answer all prompts with the entry of the value of the requested variable, followed by an R/S. This option is very useful when you know your functions are limited to the square with the above coordinates, as a lot of functions are.

Make sure you only enter positive integers for the number of tapes and the number of rotations!!

2. Enter the print/plot program.

3. Initialise the print/plot part by pressing the following sequence of keys, not paying attention to any flashing display.

Make sure the Master Library is in the calculator.

Press	*Display*	*Comment*
10 OP 17	159.99	
CLR	0	
GTO 016	0.	
PGM 19 SBR 045	0.	Flashing zero with decimal point.
P/R	0	Decimal point has disappeared.
LRN	016 55	
INS	016 55	This INS creates the h25 needed for Graphics Mode.
SST [16x]	032 65	Make sure you don't SST past step 032!
INS	032 65	This second INS creates the h12 needed for Fast Mode.
LRN RST CLR	0	
6 OP 17	479.59	Any other partitioning that gives access to R01 through R20 is also allowed.

Although the above process doesn't seem to do anything it alters steps 016-023 and 032-039 and inserts two steps at step 024 and 040, causing the original steps 039-159 to shift down two steps, so that steps 158 and 159 disappear.

After this initialisation steps 014-040 look like this on a PC-100:

014 69 OP	021 15 15	028 35 1/X	035 43 RCL
015 05 05	022 60 DEG	029 22 INV	036 14 14
	023 00 0	030 58 FIX	037 61 GTO
017 92 RTN	024 93 .	031 86 STF	038 01 01
018 43 RCL	025 01 1	032 12 12	039 99 99
019 02 02	026 34 √X	033 68 NOP	040 56 DEL
020 42 STO	027 33 X↑2	034 29 CP	

Fast Graphics Mode Polar Plotter - (cont)

4. Now you must enter your functions, starting at step 280. Remember that the calculator is in DEG-mode when the program arrives at step 280! In your functions you must use RCL 07 as Theta. To let the calculator know the end of your function you must end it with: X:T xxx GTO 102 or GTO 109. In this sequence xxx is the address of the first step of the next function. To designate the last function you must end it with: X:T 71 GTO 102/109.
   To see if you need GTO 102 or GTO 109 you can use this table, which is valid for functions of the type R=Sin or Cos(M*Theta/N).

M	N	GTO
Odd	Even	102
Even	Odd	109
Odd	Odd	102

For functions of the above type you should enter the number of rotations equal to N.

If either M or N is odd you also have to program an IxI before the X:T. [This is due to the fact that the PC-100 can only print forward] When you want to plot other types of functions you have to decide yourself if you have to end them in either GTO 102 or GTO 109 and also if they need an IxI.

Here are some examples of functions, the print-out is included on a separate sheet:

```
280  43 RCL
281  07  07
282  65  x
283  02  2
284  95  =
285  38 SIN
286  50 IxI
287  32 X:T
288  02  2
289  09  9
290  04  4
291  61 GTO
292  01  01
293  09  09
294  01  1
295  32 X:T
296  07  7
297  01  1
298  61 GTO
299  01  01
300  09  09
```

Example 1

```
280  43 RCL
281  07  07
282  65  x
283  03  3
284  95  =
285  38 SIN
286  32 X:T
287  02  2
288  09  9
289  03  3
290  61 GTO
291  01  01
292  02  02
293  43 RCL
294  07  07
295  38 SIN
296  32 X:T
297  07  7
298  01  1
299  61 GTO
300  01  01
301  09  09
```

Example 2

```
280  43 RCL
281  07  07
282  35 1/X
283  65  x
284  01  1
285  08  8
286  00  0
287  55  /
288  89 PI
289  95  =
290  32 X:T
291  07  7
292  01  1
293  61 GTO
294  01  01
295  09  09
```

Example 3

```
280  43 RCL
281  07  07
282  65  x
283  03  3
284  95  =
285  39 COS
286  65  x
287  43 RCL
288  07  07
289  39 COS
290  95  =
291  32 X:T
292  07  7
293  01  1
294  61 GTO
295  01  01
296  09  09
```

Example 4

Example 1 was made using the B option of the data-entry part, the other three using the A option.

```
TAPES
             1.
ROTATIONS
             1.
```

```
X-MIN
          -0.95
X-MAX
           0.95
Y-MIN
           -0.6
Y-MAX
           1.02
TAPES
             2.
ROTATIONS
             1.
```

```
X-MIN
          -0.35
X-MAX
           0.18
Y-MIN
          -0.32
Y-MAX
            0.7
TAPES
             2.
ROTATIONS
             3.
```

```
X-MIN
         -0.125
X-MAX
             1.
Y-MIN
          -0.25
Y-MAX
           0.26
TAPES
             1.
ROTATIONS
             1.
```

Fast Graphics Mode Polar Plotter - (cont)

When you enter your functions you have to observe the usual rules for Fast and Graphics Mode programs.

5. After having entered your functions you can press A to start the program.

The time for the program to finish the printing ranges from quite to extremely long, but I personally think the results are worth it!

Finally, I would like to end this description with a list of registers and their use in the program.

R01 X-min
R02 Y-min
R03 dx,dy
R04 Number of lines
R05 Epsilon [In the program eps is given the value dx/Sqrt(2), but sometimes it may be desirable to make eps a little larger]
R06 Number of rotations
R07 Calculated Theta
R08 Calculated R
R09 Pointer to printregister
R10 Pointer to printcode
R11 Symbol counter [There are 3 possible symbols for every position]
R12 Position counter [There are 5 possible positions in every printregister]
R13 Op counter [There are 4 printregisters, OP 1 through OP 4]
R14 Working X
R15 Working Y
R16 Lines counter
R17 Tapes counter
R18 Rotations counter
R19 Register in which the printcode for an OP is assembled
R20 Register to store the return-addresses of the user-entered functions

The method used to print/plot the functions is quite simple:

1. A grid is placed over the X/Y-plane.

2. For every cross-point on this grid with coordinates (X,Y) the polar representation is calculated, that is R=Sqrt(X**2+Y**2) and Theta=Arctan(Y/X).

3. The calculated Theta is entered in the function that is defined by the user to generate an R'.

4. This R' is compared to the real R for the (X,Y)-coordinate and at this point there are two possibilities:

   a. The difference between R and R' is less than epsilon.

      In this case a line is printed at that particular (X,Y)-coordinate. If the above mentioned line is part of the L (sub-)position within an OP-position the program continues with the (X,Y)-coordinate belonging to the J (sub-)position. If that point would also give rise to a printout then the printcode for that OP-position, which was 27 (for L), is changed into 41 (for U) to accomadate both points, thereby increasing the resolution of Graphics Mode even more.

Fast Graphics Mode Polar Plotter - (cont)

b. The difference between R and R' is more than epsilon.

In this case the program goes on with the next function or, if there is no next function, it it will add 180 or 360 degrees to Theta and go on with the first function. By selecting GTO 102 or GTO 109 the user can select between 360 or 180 degrees respectively. When all rotations for the point under consideration are completed the program continues with the next (X,Y)-coordinate.

Listing of the data entry program:

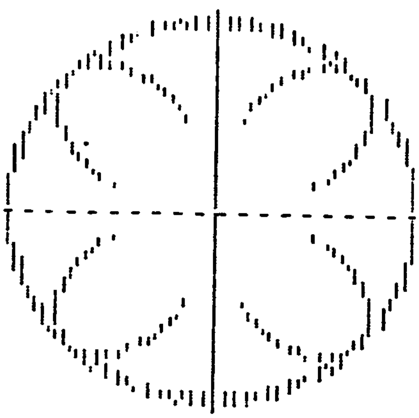
```
000  76 LBL   060  05  5    120  42 STO   180  44 SUM
001  10 E'    061  03  3    121  02  02   181  04  04
002  69 OP    062  02  2    122  22 INV   182  69 OP
003  01  01   063  03  3    123  44 SUM   183  24  24
004  69 OP    064  07  7    124  08  08   184  69 OP
005  05  05   065  01  1    125  04  4    185  00  00
006  00  0    066  03  3    126  05  5    186  69 OP
007  91 R/S   067  03  3    127  02  2    187  05  05
008  99 PRT   068  07  7    128  00  0    188  00  0
009  92 RTN   069  10 E'    129  03  3    189  92 RTN
010  76 LBL   070  42 STO   130  00  0    190  76 LBL
011  19 D'    071  06  06   131  01  1    191  12  B
012  42 STO   072  44 SUM   132  03  3    192  71 SBR
013  09  09   073  06  06   133  04  4    193  01  01
014  53  (    074  69 OP    134  04  4    194  84  84
015  73 RC*   075  36  36   135  10 E'    195  47 CMS
016  09  09   076  92 RTN   136  44 SUM   196  18 C'
017  55  /    077  76 LBL   137  08  08   197  53  (
018  43 RCL   078  11  A    138  18 C'    198  43 RCL
019  03  03   079  71 SBR   139  53  (    199  17  17
020  72 ST*   080  01  01   140  43 RCL   200  42 STO
021  09  09   081  84  84   141  08  08   201  04  04
022  85  +    082  47 CMS   142  55  /    202  65  x
023  22 INV   083  04  4    143  53  (    203  04  4
024  59 INT   084  04  4    144  06  6    204  00  0
025  69 OP    085  02  2    145  00  0    205  54  )
026  10  10   086  00  0    146  65  x    206  35 1/X
027  54  )    087  03  3    147  43 RCL   207  42 STO
028  59 INT   088  00  0    148  17  17   208  05  05
029  64 PD*   089  02  2    149  75  -    209  01  1
030  09  09   090  04  4    150  01  1    210  93  .
031  92 RTN   091  03  3    151  54  )    211  00  0
032  76 LBL   092  01  1    152  52 EE    212  04  4
033  18 C'    093  10 E'    153  54  )    213  04  4
034  03  3    094  42 STO   154  58 FIX   214  94 +/-
035  07  7    095  01  01   155  02  02   215  42 STO
036  01  1    096  04  4    156  52 EE    216  01  01
037  03  3    097  04  4    157  22 INV   217  42 STO
038  03  3    098  02  2    158  57 ENG   218  02  02
039  03  3    099  00  0    159  22 INV   219  53  (
040  01  1    100  03  3    160  58 FIX   220  93  .
041  07  7    101  00  0    161  53  (    221  00  0
042  03  3    102  01  1    162  42 STO   222  03  3
043  06  6    103  03  3    163  03  03   223  04  4
044  10 E'    104  04  4    164  55  /    224  08  8
045  42 STO   105  04  4    165  02  2    225  55  /
046  17  17   106  10 E'    166  34 √x    226  43 RCL
047  02  2    107  42 STO   167  54  )    227  17  17
048  04  4    108  07  07   168  42 STO   228  54  )
049  03  3    109  04  4    169  05  05   229  42 STO
050  02  2    110  05  5    170  01  1    230  03  03
051  03  3    111  02  2    171  19 D'    231  06  6
052  01  1    112  00  0    172  50 IxI   232  00  0
053  03  3    113  03  3    173  42 STO   233  49 PRD
054  06  6    114  00  0    174  04  04   234  04  04
055  00  0    115  02  2    175  02  2    235  61 GTO
056  00  0    116  04  4    176  19 D'    236  01  01
057  69 OP    117  03  3    177  07  7    237  82  82
058  02  02   118  01  1    178  19 D'
059  03  3    119  10 E'    179  50 IxI
```

Listing of the print/plot program:

```
000  92 RTN   056  99 PRT   112  32 X:T   168  00  0    224  43 RCL
001  76 LBL   057  43 RCL   113  75  -    169  48 EXC   225  06  06
002  11  A    058  15  15   114  43 RCL   170  19  19   226  42 STO
003  61 GTO   059  42 STO   115  08  08   171  84 OP*   227  18  18
004  00  00   060  02  02   116  54  )    172  09  09   228  53  (
005  41  41   061  98 ADV   117  50 IxI   173  69 OP    229  53  (
006  76 LBL   062  97 DSZ   118  77  GE   174  29  29   230  43 RCL
007  12  B    063  17  17   119  40 IND   175  97 DSZ   231  15  15
008  61 GTO   064  00  00   120  20  20   176  13  13   232  55  /
009  00  00   065  41  41   121  83 GO*   177  02  02   233  32 X:T
010  18  18   066  29 CP    122  10  10   178  08  08   234  43 RCL
011  76 LBL   067  00  0    123  02  2    179  43 RCL   235  14  14
012  13  C    068  92 RTN   124  07  7    180  03  03   236  54  )
013  25 CLR   069  01  1    125  44 SUM   181  44 SUM   237  22 INV
014  69 OP    070  08  8    126  19  19   182  14  14   238  30 TAN
015  05  05   071  00  0    127  09  9    183  92 RTN   239  85  +
016  74 SM*   072  44 SUM   128  44 SUM   184  02  2    240  43 RCL
017  90  90   073  07  07   129  10  10   185  93  .    241  14  14
018  33 X^2   074  22 INV   130  02  2    186  02  2    242  69 OP
019  64 PD*   075  87 IFF   131  42 STO   187  35 1/X   243  10  10
020  11  11   076  04  04   132  11  11   188  82 HIR   244  22 INV
021  84 OP*   077  00  00   133  43 RCL   189  05  05   245  39 COS
022  20  20   078  82  82   134  03  03   190  82 HIR   246  85  +
023  29 CP    079  22 INV   135  44 SUM   191  06  06   247  69 OP
024  01  1    080  86 STF   136  15  15   192  82 HIR   248  10  10
025  34 √x    081  04  04   137  61 GTO   193  07  07   249  65  x
026  33 X^2   082  97 DSZ   138  00  00   194  82 HIR   250  53  (
027  35 1/X   083  18  18   139  93  93   195  08  08   251  46 INS
028  22 INV   084  02  02   140  61 GTO   196  61 GTO   252  75  -
029  58 FIX   085  80  80   141  01  01   197  01  01   253  01  1
030  86 STF   086  03  3    142  25  25   198  79  79   254  54  )
031  71  71   087  44 SUM   143  04  4    199  67 X=T   255  65  x
032  35 1/X   088  10  10   144  07  7    200  01  01   256  01  1
033  40 IND   089  43 RCL   145  93  .    201  84  84   257  08  8
034  04  4    090  03  03   146  02  2    202  01  1    258  00  0
035  04  4    091  44 SUM   147  05  5    203  42 STO   259  54  )
036  53  (    092  15  15   148  93  .    204  09  09   260  42 STO
037  59 INT   093  97 DSZ   149  01  1    205  04  4    261  07  07
038  56 DEL   094  11  11   150  04  4    206  42 STO   262  53  (
039  43 RCL   095  02  02   151  59 INT   207  13  13   263  43 RCL
040  01  01   096  24  24   152  44 SUM   208  05  5    264  14  14
041  42 STO   097  61 GTO   153  19  19   209  42 STO   265  33 X^2
042  14  14   098  01  01   154  43 RCL   210  12  12   266  85  +
043  43 RCL   099  64  64   155  03  03   211  03  3    267  43 RCL
044  04  04   100  42 STO   156  49 PRD   212  42 STO   268  15  15
045  42 STO   101  20  20   157  11  11   213  11  11   269  33 X^2
046  16  16   102  22 INV   158  56 DEL   214  01  1    270  54  )
047  12  B    103  87 IFF   159  56 DEL   215  00  0    271  34 √x
048  13  C    104  04  04   160  43 RCL   216  00  0    272  42 STO
049  97 DSZ   105  40 IND   161  11  11   217  49 PRD   273  08  08
050  16  16   106  20  20   162  44 SUM   218  19  19   274  00  0
051  00  00   107  42 STO   163  15  15   219  01  1    275  67 X=T
052  49  49   108  20  20   164  97 DSZ   220  04  4    276  40 IND
053  98 ADV   109  43 RCL   165  12  12   221  02  2    277  10  10
054  43 RCL   110  05  05   166  02  02   222  42 STO   278  86 STF
055  17  17   111  53  (    167  11  11   223  10  10   279  04  04
```

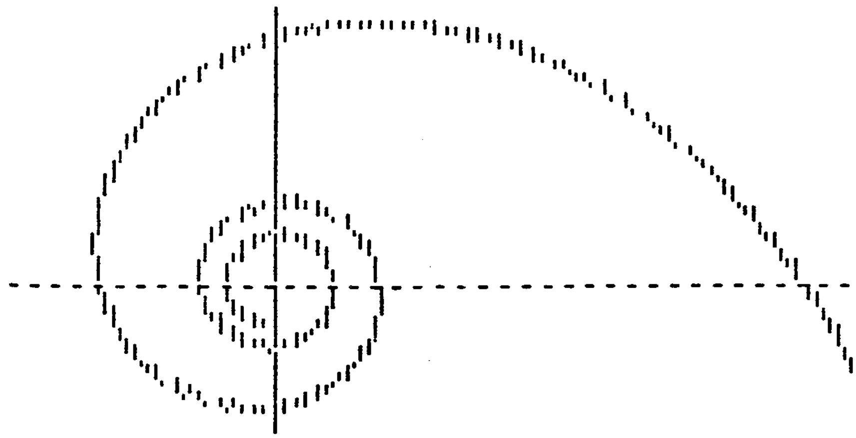
Editor's Note: Sample plots appear on the front page and on page 22. The plots are numbered to correspond to the example problems defined on page 19. Unfortunately, I have so far been unable to duplicate all of the plots with my own TI-59/PC-100. I do obtain exactly the same plot for Example 1. For Example 3 I obtain two spirals, one inside the other, as illustrated on page 22. I will continue investigating the differences and report the results in the next issue. In the meantime, happy plotting!

Fast Graphics Mode Polar Plotter - (cont)



1.

Example 1



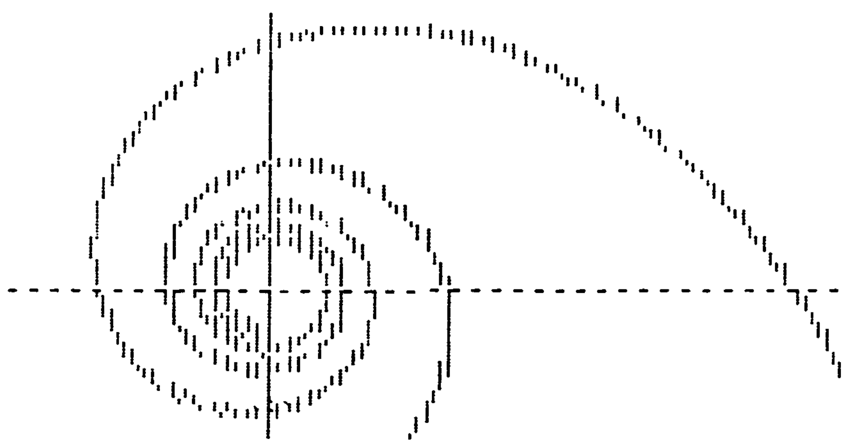
2. 1.

Example 3

The upper example 3 curve was provided by Robert Prins with the material he submitted with his program.

The lower example 3 curve is the one the editor obtained using the data entry and function on page 19. Note the dual spirals.

The editor obtained the same plot for example 1, and has not yet tried examples 2 and 4.



2. 1.

MORE TI-66 CAPABILITY - Dave Leising has been active in the search for additional capability for the TI-66. He found that if a program is being executed which includes a PAUSE instruction, then holding down any of the keys LRN, BST, SST, OP, X=T, GTO, SBR, LBL, (,), CE, or CLR will cause the pause to be ignored so that the program runs at full speed. You may test that by inserting a pause at step 007 of Robert Prins' factorial program on page 25, that is, just before the Dsz. Then, enter a 6, press A and the values 6, 30, 120, 360 and 720 will be flashed in sequence and the execution time will be stretched to about twelve seconds. Again, enter a 6 and press A, but then press and hold the LBL key. There will be no pauses, and the execution time will be returned to four seconds.

Dave has also determined that, for all the merged instruction codes, the "long form" also works. Thus

INV SBR is equivalent to RTN

GTO IND is equivalent to GO* , etc.

SPECIALITY PAKETTE AVAILABILITY - The specialty pakettes provided documentation for groups of 5 to 11 TI-59 programs in a given subject area. The programs were selected from those in PPX Exchange, but were retyped and reprinted for improved legibility and bound in a notebook format. The price for a pakette was ten dollars, a bargain relative to the single copy price for PPX programs of five dollars each. Eighteen pakettes were issued.

Page 2 of this issue noted that former member William Vogel had donated his documentation to the club. Four pakettes were included. I will loan these pakettes to club members. Send three dollars for each pakette to cover postage. I will send the pakette by first class mail and expect you to return it the same way. The pakettes which are available are:

9B0002	Statistical Testing	9B0007	Printer Utility
9B0005	Blackbody	9B0009	Programming Aids

Members are invited to list other pakettes which they would make available on loan in coming issues.

PROGRAM SUPPORT - Page 77 of the June 12, 1986 issue of EDN carried an advertisement for the HP-41 which states that "... The HP-41 has over 2,500 software programs to choose from. More than any other calculator. ..." They can make that statement only because TI chose to discontinue support of PPX. In June 1982 the H Update to the PPX Software Catalog listed 3640 programs, more than a thousand more than the number in the HP advertisement.

FOR BI-LINGUALS - The 1985 issues of the NASA STAR Index listed two sets of programs for the HP-41CV. The documentation for both sets is available from the National Technical Information Service (NTIS).

N85-35192# Naval Postgraduate School, Monterey, Calif.
DEVELOPMENT OF NATOPS PERFORMANCE SOFTWARE FOR THE SH-3D AND SH-3H HELICOPTERS M.S. Thesis
J. T. Curtis Mar. 1985 107 p
(AD-A156140) Avail: NTIS HC A06/MF A01 CSCL 01/2

This thesis generates closed form equations for significant and frequently used Naval Air Training and Operating Procedure Standardization (NATOPS) performance charts for the SH-3D and SH-3H helicopters. These equations are developed into interactive software for the Hewlett-Packard HP-41CV hand-held programmable calculator. With this software installed in the calculator the user is able to calculate numerous NATOPS performance parameters (expeditiously, with reduced risk of error) both prior to and in flight.
Author (GRA)

N85-23106# National Inst. for Aeronautics and Systems Technology, Pretoria (South Africa). Aeroelasticity Div.
FIBRE COMPOSITE LAMINATED PLATE ANALYSIS USING A CARD PROGRAMMABLE CALCULATOR (HP41C/CV)
A. P. N. Sutherland Oct. 1983 155 p refs
(NIAST-83/88; ISBN-0-7988-3020-4) Avail: NTIS HC A08/MF A01

The HP 41C/CV programs which perform the stiffness and strength calculations for composite laminates are presented. Two sets of programs are given, each to be used interactively: (1) symmetric laminates or sandwich beams comprising one type of ply material only; and (2) general laminates comprising up to three different ply materials. The programs include nonmechanical effects. The strength ratios (reserve factors) calculated are for first ply failure (FPF) and are based on mechanical strain. Behavior beyond FPF is not included. Classical lamination theory (CLT) is used.
E.A.K.

THE TI-74 BASICALC - Maurice Swinnen reports that the title page of a manual for the Mechatronic Quick Disk (see V10N4P4) indicates that the device can be used with both the CC-40 and the TI-74 BASICALC. This seems to confirm the notation that appeared in the Executive Photo Catalog (see V10N4P3), but as yet there is nothing from TI's Consumer Press Relations organization.

DSZ 15 ON THE TI-66 - Robert Prins. V10N4P13 suggested that it is possible to Dsz on registers higher than R09 with the TI-66. To do so it is necessary to implant some hexadecimal code in memory. As with the TI-59 the implanting process involves non-normal keystroke sequences. For a Dsz 15 you must implant a 0F code which can be done with the following sequence:

1. Press 2nd Part 64 to set the partitioning to 0.63, that is, all data registers and no program steps.

2. Press 1 EE 5 0 / EE . 4 +/- 1 +/- $\sqrt{x}$ =

   After the square root key is pressed the display will be "1. 4E" . After the equals key is pressed the display will be "1. 0 " where the blank after the zero is equal to hexadecimal F.

3. Press STO 63, STO 62, STO 61, and STO 60. Press INV EE to go out of the EE mode. Recall the contents of the registers to the display and see "100000". If the PC-200 is connected press 60 INV 2nd List you and see "100000." printed for each register.

4. Press 2 Op 17, or any other set any other partitioning which leaves at least sixteen data registers (R00 through R15). The data placed in R60 through R63 will have been converted to instructions. Listing with the PC100 will show ones at steps 000, 008, 016, and 024, and blanks at steps 007, 015, 022, and 031. Zeroes will be printed for all of the other steps in the range from steps 000 through 031. Go to LRN and SST through steps 000 throug 031. At locations 007, 015, 023 and 031 where the printout showed blanks, the display shows the mnemonic "**". It turns out that this double asterisk is not uniquely associated with the hexcode 0F. We will cover more on that in future issues.

5. To insert a demonstration program around the 0F codes go out of LRN, press RST, and return to LRN mode. Then:

   a. Press LBL A STO 15 RCL 00 . The display will be "005 00".

   b. Press SST seven times. The display will be :012 0".

   c. Press 2nd Del eight times. The display will be "004 RCL".

   d. Press SST. The display will be "005 **".

   e. Press x 2nd DSZ 0 A . The display will be "009 A".

   f. Press SST seven times. The display will be "016 0".

   g. Press 2nd Del nine times. The display will be "007 DSZ".

   h. Press SST one time. The display will be "008 **".

   i. Press GTO 0 0 4. The display will be "011 04".

   j. Press BST two times. The display will be "009 GTO".

   k. Press 2nd Del. The display will be "008 **".

Dsz 15 on the TI-66 - (cont)

l. Press SST two times. The display will be "010 04".

m. Press 1 = R/S. The display will be "013 R/S".

6. Go out of LRN, press RST, and go back to LRN. SST through steps 000 through 013 and you should see the program

LBL A STO 15 RCL ** x DSZ ** 00 04 1 = R/S

which should be a program for calculating factorials if RCL ** recalls the contents of data register 15 (OF) and DSZ ** dsz's the contents of data register 15. Press 6 and then A. In about four seconds 720 (6!) will appear in the display.

Of course, there will be many more things that the TI-66 user will be able to do with hexadecimal codes. For example, we could have used a STO ** at steps 002/003 to store the initial value for the factorial calculation. Two extra OF codes were implanted in steps 1 through 4 above for the user to experiment with. In future issues we will show how to generate other hexadecimal codes, and give other examples of how to use them in programs.

SOCIAL SECURITY NUMBER PUZZLE

In V10N4P12 Maurice Swinnen proposed the puzzle of finding a nine digit number for which the first two digits on the left are divisible by two, the first three numbers are divisible by three, etc., up to the entire number is divisible by nine. The number does not contain zeroes, and no digit is repeated. Maurice proposed a very slow computer solution. Members were asked to write a faster solution.

The correct answer is 381654729. Larry Leeds and Robert Prins obtained the solution without the use of a computer program. Maurice Swinnen submitted a program from a friend written for a Radio Shack computer. I converted it for use on the CC-40 (listing at the right); it obtains the solution in about six minutes. George Thomson submitted a longer program which obtains the solution in only 32 seconds on his Panasonic Senior Partner running in ZBASIC.

```
100 S(5)=5
2000 FOR A=2 TO 8 STEP 2
2020 FOR B=2 TO 8 STEP 2
2030 IF B=A THEN 2140
2040 FOR C=2 TO 8 STEP 2
2050 IF C=A OR C=B THEN 2120
2060 FOR D=2 TO 8 STEP 2
2070 IF D=A OR D=B OR D=C THEN 2100
2080 S(2)=A:S(4)=B:S(6)=C:S(8)=D
2090 GOTO 3000
2100 NEXT D
2120 NEXT C
2140 NEXT B
2160 NEXT A
3000 FOR W=1 TO 9 STEP 2
3010 IF W=5 THEN 3560
3020 FOR X=1 TO 9 STEP 2
3030 IF X=W OR X=5 THEN 3540
3040 FOR Y=1 TO 9 STEP 2
3050 IF Y=W OR Y=X OR Y=5 THEN 3520
3060 FOR Z=1 TO 9 STEP 2
3070 IF Z=W OR Z=X OR Z=Y OR Z=5 THE
N 3500
3075 J=J+1:IF J>576 THEN END
3080 S(1)=W:S(3)=X:S(7)=Y:S(9)=Z
3090 S$="":FOR Q=1 TO 9:S$=S$&STR$(S
(Q))
3100 IF Q<3 OR Q=5 THEN 3400
3120 SS=VAL(S$)
3140 IF (SS/Q-INT(SS/Q))>.01 THEN 35
00
3400 NEXT Q
3420 PRINT "The number is ":SS
3450 PAUSE
3500 NEXT Z
3520 NEXT Y
3540 NEXT X
3560 NEXT W
3600 GOTO 2100
```

All of the solutions relied on the idea that the fifth digit must be a five, and the even digits must be even, and proceed to shuffle and test the digits according to some algorithm. More on the methods in the next issue.

CLOSING OF SERVICE CENTERS - Two members have written to tell me that the TI Customer Service Centers are closing and that in the future calculators in need of service should be sent to Texas Instruments Incorporated, 2305 North University Avenue, Lubbock, Texas 79415. The information number 1-800-TI-CARES is still available for current repair information.

WANTED - Real Estate Module or other modules with documentation for the TI-59. Also, will take the best offer over $75.00 for either my PC-100A or TI-59 with cards, several modules and accessories. Call Bob at (718)-436-8463.

GENERAL LEDGER PACKAGE FOR THE TI-59 - Louis Krumpelman. This program system for the TI-59 requires a PC-100 and the Math/Utilities module. The features include 99 accounts, a separate check register, an entry tape that leaves an audit trail, an interim trial balance which verifies the data base, double entry accounting, etc. Five separate programs are involved. A thorough 61 page spiral bound program system description is available. Louis will provide the system description for $8.00 in the USA. Outside the USA the price in $14.00 in US funds. If you want the magnetic cards send an additional $2.00 AND 6 magnetic cards. Write to Louis Krumpelman, P.O. Box 698, Richmond KY 40475-0698. If you include a #10 SASE with your order he will send a notice of any changes or corrections which occur in the next 9-12 months.

NO TEA PLEASE - V10N4P5 defined this puzzle by Charlie Williamson. He forwarded his 29 step solution. Other members who have submitted solutions include Hewlett Ladd, Larry Leeds, Don Graham and Robert Prins. Don and Robert submitted very similar 27 step solutions:

Don's Solution:

```
LBL E ( RCL 0 - RCL 1 ) OP 10 + ( RCL 0 - RCL 2 ) OP 10 = / 2 = INT RTN
```

Robert's Solution:

```
LBL A ( ( RCL 1 - RCL 0 ) OP 10 + ( RCL 1 - RCL 2 ) OP 10 - OP 10 ) R/S
```

TABLE OF CONTENTS

Magnetic card service for the programs in this issue and for 1983, 1984 and 1985 is available for the price of one dollar per card plus a SASE.