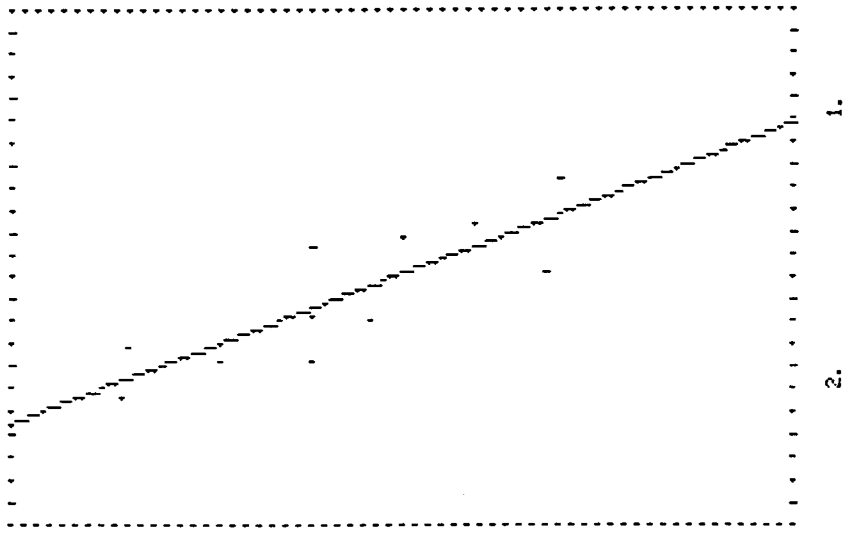
TI PPC NOTES

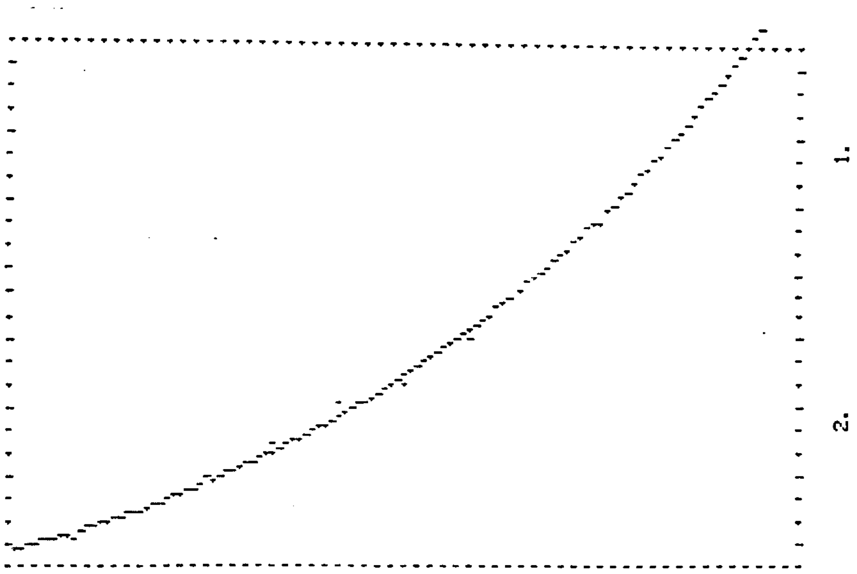
NEWSLETTER OF THE TI PROGRAMMABLE CALCULATOR CLUB
P.O. Box 1421, Largo, FL 34294

Volume 9, Number 4 July/August 1984

Plot 60 Plus - see page 14

Linear Regression with High Resolution Graphics - example on page 16.





Exponential Curve Fit - example from page 5-11 of the Applied Statistics Manual

The newsletter is not copyrighted and may be reproduced for personal use. When material is used elsewhere we ask as a matter of courtesy that TI PPC Notes be mentioned. The use of material in this newsletter is entirely at the user's risk. No responsibility as to the accuracy and the consequences due to the lack of it will be borne by either the club or the editor.

ERRATA:

TI-66/PC-200 Calendar - V9N2P9 - Dave Leising writes that the time of execution for his calendar program is one hour thirty-seven minutes, not the one hour forty-six seconds as listed in V9N2P9. Dave also had run Robert Prins' improved calendar program which reduces the execution time to one hour twenty-four minutes.

---

13 DIGIT REGISTER LIST PROGRAMS - V9N3P14/15 - Several readers noted that the comments for the Durbin and Snow programs on V9N3P15 did not seem to match the printouts. That is a result of an error in pasting up the proofs. The editor goofed and used the wrong printouts. The correct printouts to go with the comments are shown below. I did not reproduce the correct printouts for the Worthington/Regelman program since the comments on V9N3P15 were consistent with those printouts.

```
    INV-List             Durbin                  Snow               Markusson
                      (V5N9/10P17)          (V5N9/10P17)         (PPX Exchange)

  3.141592654     00 | 3.141592653590   00 |                     | 00 3.141592653590 00
  1.111111111     01 | 1.111111111111   01 | 1.111111111111   01 | 01 1.111111111111 00
  2.222222222     02 | 2.222222222222   02 | 2.222222222222   02 | 02 2.222222222222 00
  3.333333333     03 | 3.333333333333   03 | 3.333333333333   03 | 03 3.333333333333 00
  4.444444444     04 | 4.444444444444   04 | 4.444444444444   04 | 04 4.444444444444 00
  5.555555556     05 | 5.555555555555   05 | 5.555555555555   05 | 05 5.555555555555 00
  6.666666667     06 | 6.666666666666   06 | 6.666666666666   06 | 06 6.666666666666 00
  7.777777778     07 | 7.777777777777   07 | 7.777777777777   07 | 07 7.777777777777 00
  8.888888889     08 | 8.888888888888   08 | 8.888888888888   08 | 08 8.888888888888 00
          10.     09 | 9.999999999999   09 | 9.99999999999    09 | 09 9.999999999999 00
           1.     10 | .9999999999999   10 | 0.999999999999   10 | 10 9.999999999999-01
  3.141592654     11 | 3.141592653590   11 | 3.14159265359    11 | 11 3.141592653590 00
       1.2345 12  12 | 123450000000.    12 | 123450000000.    12 | 12 1.234500000000 12
       1.2345 13  13 |      1.2345 13   13 | 1234500000000    .1 | 13 1.234500000000 13
       1.2345 14  14 |      1.2345 14   14 | 1234500000000    1. | 14 1.234500000000 14
       1.2345 15  15 |      1.2345 15   15 | 1234500000000    15 | 15 1.234500000000 15
           1.     16 | 1.000000000000   16 |  1.              16 | 16 1.000000000000 00
  2.718281828     17 | 2.718281828459   17 | 2.718281828459   17 | 17 2.718281828459 00
    4.3170165 63  18 |   4.3170165 63   18 | 4317016463003O   18 | 18 4.317016463003 63
           0.     19 |          0.      19 |                     | 19 0.000000000000 00
           0.     20 |          0.      20 |                     | 20 0.000000000000 00
           0.     21 |          0.      21 |                     | 21 0.000000000000 00
   123456789.     22 | 123456789.0000   22 | 123456789.       22 | 22 1.234567890000 08
  .0000000001     23 |        1.23-10   23 | 0.000000000123   23 | 23 1.230000000000-10
    9.8765432 11  24 | 987654321987.0   24 | 987654321987.    24 | 24 9.876543219870 11
  .4971498727     25 | .4971498726941   25 | 0.497149872694   25 | 25 4.971498726941-01
           1.     26 | .9999999999950   26 | 0.999999999995   26 | 26 9.999999999950-01
  0.000012345     27 | .0000123450000   27 | 0.000012345      27 | 27 1.234500000000-05
           0.     28 |          0.      28 |                     | 28 0.000000000000 00
           0.     29 |          0.      29 |                     | 29 0.000000000000 00
```

Durbin - R18 shows that only eight digits are printed when the EE mode applies. Trailing zeroes are seen in R00, R16, R22, R24, R26, and R27.

Snow - R00 cannot be printed. R09 shows a missing 13th digit when the first digit is a nine. R10 and R25 show a missing 13th digit when the printout is of the form).XXX... . R13 and R14 show spillover of the decimal point into the register number for some exponentials. R15 is incorrect. R18 shows 14 digits, but no indication that the number is much larger. No printouts for R19, R20, R21, R28 and R29 which contain 0.

ANOTHER TEST OF PRECISION - V9N2P11 described Fred Gruenberber's test of precision which appeared in "Computer Recreations" in the April 1984 issue of Scientific American. The test problem was to enter the number 1.0000001 and press the square key 27 times. This is equivalent to raising the initial number to the 134,217,728th power. A table was presented for an assortment of calculators and computers. None of the machines listed got even 7 digits correct. For nearly every machine there are several ways to solve the problem. For example, in BASIC language the problem may be solved as:

```
     Method A                 Method B                 Method C
     --------                 --------                 --------
10 A = 1.0000001         10 A = 1.0000001         10 A = 1.0000001
20 FOR I = 1 TO 27       20 FOR I = 1 TO 27       20 B = 134217728
30 A = A^2               30 A = A*A               30 A = A^B
40 NEXT I                40 NEXT I                40 PRINT A
50 PRINT A               50 PRINT A               50 END
60 END                   60 END
```

On most calculators the equivalent of methods A and B will yield identical results, but the y^x function which is equivalent to method C will return a different, and usually more accurate, result. On many computers the results vary widely with method; for example, the BASIC on the Commodore 64 and Apple II+. V9N2P11 presented results from the three methods from a representative set of machines. V9N3P3 reported an answer correct to thirty digits from Laurence Leeds. Additional results were received from Myer Boland and Laurance Leeds for the TI-99/4A, and from Howard Yamaguchi for the Casio PB-700 and the Sinclair Spectrum. Dejan Ristanovic confirmed that the HP-41 yields the same answers as the HP-11, but wondered why 1.0000001 log 134217728 * 10^x yields 674530.4708 . The HP-11 yields the same value for that sequence, but yields 674530.4715 if natural logarithm functions are used. Tests with the TI-59 show that the use of natural logarithms yields the same solution as the y^x function, but the use of base ten logarithms yields a substantially degraded result. That only confirms what we might have already suspected based on the revealed firmware. Results received to date are:

Machine	Method A	Method B	Method C
Exact	674530.47074108..	674530.47074108..	674530.47074108..
Model 100	674529.41305068	674529.41305068	674530.47074049
TI-66	674520.6067381	674520.6067381	674530.4707400
HP-11, etc.	674494.0561	674494.0561	674530.4707
Casio PB-700	674475.396	674475.3961	674530.4707
R/S EC-4009	674294.1172	674294.1172	674530.47055
CC-40	674530.31804225	674530.31804225	674621.4634954
TI-MBA	674492.75112	674492.75112	674530.92340
TI-99/4A	674530.31804225	674530.31804225	674530.9234107
BA-55	674432.82060	674432.82060	674530.92317
TI-57	674432.8204	674432.8204	674530.9232
TI-59	674520.6052712	674520.6052712	674530.9234109
TI-55II/57LCD	674432.8206	674432.8206	660003.2248
TI-35	674363.69092	674363.69092	673336.17392
Sharp EL-326S		671189.63	
Sinclair Spec.	710493.46	685090.6	685222.8
Color Computer	713658.879	643571.305	665348.188
C-64/Apple	728339.418	22723.9709	665348.189

AN IMPROVED MARKUSSON 13 DIGIT PRINTER - Gene Friel. The Markusson thirteen digit printer from the May/June 1982 issue of PPX Exchange can only control the beginning register. The program then lists register contents from the beginning register to the end of the partition. This revision permits control of the ending register as well. The instructions for registers 00 through 89 are:

1. Enter the card containing the program on bank 1.
2. Press GTO 217 LRN. Enter the two digits defining the last register to be printed in program steps 217 and 218. A register less than ten must be indicated by 0X. Press LRN and CLR.
3. Enter the number of the first register to be printed into the display and press C. The register contents will be printed at a rate of about one register per 15 seconds. If you inadvertently enter an ending register number smaller than the starting register number, then only the contents of the starting register are printed.

Program Listing:

```
000  76 LBL     040  16 A'      080  90  90     120  08   8     160  10 E'      200  82 HIR
001  16 A'      041  16 A'      081  50 I×I     121  77  GE     161  82 HIR     201  38  38
002  55  ÷      042  16 A'      082  82 HIR     122  01  01     162  06  06     202  25 CLR
003  01  1      043  16 A'      083  04  04     123  26  26     163  10 E'      203  09   9
004  00  0      044  65  ×      084  29 CP      124  02   2     164  82 HIR     204  09   9
005  75  -      045  93  .      085  67  EQ     125  85   +     165  07  07     205  35 1/X
006  22 INV     046  01  1      086  01  01     126  73 RC*     166  25 CLR     206  82 HIR
007  59 INT     047  65  ×      087  16  16     127  90  90     167  82 HIR     207  36  36
008  65  ×      048  77  GE     088  55   ÷     128  29 CP      168  18  18     208  82 HIR
009  77  GE     049  00  00     089  28 LOG     129  77  GE     169  52 EE      209  37  37
010  00  00     050  55  55     090  77  GE     130  01  01     170  02   2     210  82 HIR
011  16  16     051  93  .      091  00  00     131  36  36     171  71 SBR     211  38  38
012  93  .      052  01  1      092  94  94     132  02   2     172  00  00     212  69 OP
013  09  9      053  95  =      093  52 EE      133  52 EE      173  43  43     213  05  05
014  95  =      054  92 RTN     094  59 INT     134  03   3     174  82 HIR     214  82 HIR
015  92 RTN     055  93  .      095  82 HIR     135  85   +     175  08  08     215  13  13
016  93  .      056  01  1      096  04  04     136  43 RCL     176  82 HIR     216  32 X:T
017  09  9      057  85  +      097  22 INV     137  91  91     177  14  14     217  00   0
018  85  +      058  93  .      098  28 LOG     138  95   =     178  50 I×I     218  04   4
019  93  .      059  00  0      099  52 EE      139  82 HIR     179  71 SBR     219  77  GE
020  00  0      060  02  2      100  55   ÷     140  05  05     180  00  00     220  00  00
021  02  2      061  95  =      101  28 LOG     141  93   .     181  43  43     221  75  75
022  95  =      062  92 RTN     102  77  GE     142  08   8     182  75   -     222  25 CLR
023  92 RTN     063  76 LBL     103  01  01     143  09   9     183  82 HIR     223  92 RTN
024  76 LBL     064  13  C      104  07  07     144  32 X:T     184  14  14     224  00   0
025  10 E'      065  32 X:T     105  01   1     145  25 CLR     185  29 CP      225  01   1
026  01  1      066  01  1      106  94 +/-     146  43 RCL     186  77  GE     226  14   D
027  52 EE      067  00  0      107  59 INT     147  90  90     187  01  01     227  00   0
028  05  5      068  69 OP      108  82 HIR     148  16 A'      188  93  93     228  10 E'
029  82 HIR     069  17  17     109  34  34     149  52 EE      189  01   1     229  10 E'
030  48  48     070  32 X:T     110  22 INV     150  06   6     190  09   9     230  00   0
031  82 HIR     071  82 HIR     111  28 LOG     151  82 HIR     191  94 +/-     231  10 E'
032  18  18     072  03  03     112  52 EE      152  35  35     192  76 LBL     232  00   0
033  59 INT     073  42 STO     113  75   -     153  01   1     193  01   1     233  00   0
034  82 HIR     074  90  90     114  22 INV     154  44 SUM     194  95   =     234  00   0
035  58  58     075  01  1      115  59 INT     155  90  90     195  22 INV     235  00   0
036  77  GE     076  00  0      116  82 HIR     156  82 HIR     196  52 EE      236  00   0
037  00  00     077  69 OP      117  08  08     157  33  33     197  52 EE      237  00   0
038  40  40     078  17  17     118  85   +     158  69 OP      198  06   6     238  00   0
039  92 RTN     079  73 RC*     119  32 X:T     159  17  17     199  94 +/-     239  00   0
```

If you want to print out data from registers 90 through 99 with the correct register notation record the register contests from bank 1 on a magnetic card, and then:

1. Enter the card containing the program below into bank 1.
2. Force the recorded data into bank 3 by **entering** -3 into the display and entering the data card.

Improved Markusson 13 Digit Printer - (cont)

3. Modify the program as follows:
   a. Replace the 90 at step 147 with a zero.
   b. Replace the 9 9 1/x at steps 203-205 with SBR 240.
   c. Enter the sequence 9 9 1/x Op 20 RTN in steps 240 through 245.
4. Enter the value 3 Q at program steps 217 and 218 where Q is the ones digit of the last register to be printed.
5. Store the vallue 11P in register 00 where P is the ones digit of the first register to be printed.
6. Enter 3P in the display and press C.

---

44 FACTORIAL FOR THE TI-57 - Robert Prins.

It took me a year to recover from the shock I got after seeing Reginald van Genechten's "Exact Factorials for the TI-57" program in V8N3P12. Here is my response, which will calculate 44! To run the program:

1. Enter the program.
2. Enter n and press RST R/S. (You do not need to press INV C.t. to initialize.)
3. When calculations are complete a 0. will be in the display. In a manner similar to the van Genechten program the results are read out from data registers 1 through 7, with the highest digits in R7 and the lowest digits in R1. For example, for 44!

RCL 7	2658271
RCL 6	57478844
RCL 5	87680436
RCL 4	25811014
RCL 3	61589031
RCL 2	96385280
RCL 1	0

4. For a new problem, go to step 2. Again, you do not need to clear the data registers.

Editor's Note: The penalty for the increased range is a severe increase in execution time. van Genechten's program would find 34! in just 2 minutes 30 seconds. This program requires 5 minutes 50 seconds for 34!, and requires 7 minutes 30 seconds for 44!.

```
00  66          X=T
01  01            1
02  32 0        STO 0
03  08            8
04  84          +/-
05 -18     INV  LOG
06  32 1        STO 1
07  86 0        LBL 0
08  43            (
09  07            7
10  38 0        EXC 0
11  75            +
12  61 2        SBR 2
13  86 1        LBL 1
14  34 1        SUM 1
15  33 1        RCL 1
16 -49     INV  INT
17  38 1        EXC 1
18  49          INT
19  42           EE
20  08            8
21  84          +/-
22 -42     INV   EE
23  38 1        EXC 1
24  38 7        EXC 7
25  38 6        EXC 6
26  38 5        EXC 5
27  38 4        EXC 4
28  38 3        EXC 3
29  38 2        EXC 2
30  38 1        EXC 1
31  56          DSZ
32  51 1        GTO 1
33  00            0
34  44            )
35  32 0        STO 0
36  56          DSZ
37  51 0        GTO 0
38  08            8
39 -18     INV  LOG
40  86 2        LBL 2
41  39 1        PRD 1
42  39 2        PRD 2
43  39 3        PRD 3
44  39 4        PRD 4
45  39 5        PRD 5
46  39 6        PRD 6
47  39 7        PRD 7
48  00            0
49 -61     INV  SBR
```

A CALL FOR PROGRAMS IN BASIC - Member Russell Byer would like to correspond with readers who have knowledge of BASIC programs in the following areas: (1) solution for non-linear systems of equations; (2) solution for ordinary differential equations; and, (3) solution for 2 and 3 dimensional partial differential equations. Write to Russell N. Byer, 1503½ South F. Street, Elwood, Indiana 46036 .

AN MBA QUIRK - P. Hanson. In V9N2P15 George Thomson reviewed the non-commutative multiply quirk of the TI-59 as described in W. Kahan's paper "Mathematics Written in the Sand ..." from the Proceedings of the Statistical Computing Section, American Statistical Association, 1983. Although the paper has a distinctly pro-HP bias it is still a rich resource. In discussing financial calculators, Kahan poses the following problem of the "yield from a risky investment":

> "For an investment of -PV = $35,000,000 now, investors are promised n = 100 equal monthly installments of an amount PMT yet to be agreed upon, but between $640,000 and $1,000,000 , plus a final payment at the 100th month of FV = $100,000,000 . How does the yield i , reckoned in percent per month, vary with PMT ?
>
> Tabulated in the first column below are selected values of PMT, with the corresponding yield in the second column as displayed on any of the hp-92, -37E, -38C or -12C after about a dozen seconds of calculation. The third column shows what the TI MBA displayed.

PMT	true i %	i on the MBA
$ 640,000	2.314053	2.314053
650,000	2.335758	-1 -97 Blinking
660,000	2.357528	2.357528
800,000	2.669065	2.669065 after a long time.
1,000,000	3.135506	-2106.949 Blinking

> The blinking tiny number is a symptom of roundoff troubles. The other anomalies could be caused by an unfortunate choice of iterative method for the equation to be solved."

A knee-jerk sort of response for defenders of calculators manufactured by TI might be that at least the MBA indicates the existence of a faulty result by the flashing display. Unfortunately, when I tested other problems with the same PV, FV, and n, but different PMT's in the close vicinity of $640,000 I found that the MBA can yield incorrect results without an accompanying flashing display:

PMT	i on the MBA
$ 649,999.95	2.335757589
649,999.96	-0.000007892
649,999.97	+0.000002975
649,999.98	+0.000001983
649,999.99	+0.000000992
650,000.00	-1.-97 Blinking
650,000.01	+0.000000992
650,000.02	+0.000001983
650,000.03	+0.000002975
650,000.04	-0.000007892
650,000.05	2.335757806

Note: The solutions for PMT between $649,999.96 and $650,000.04 take only a few seconds. The solutions at $649,999.95 and $650,000.05 take about twenty seconds.

As Professor Kahan might say, egregious! For some reason he chose not to include the results which can be obtained with the BA-55. My tests show that the BA-55 obtains the same results as the HP calculators, but with a substantially reduced execution time of only five seconds.

I plan to continue to work my way through "Mathematics Written in Sand ..." and will report the results in future issues.

FOURTEEN DIGITS OF PI FROM THE 99/4 AND CC-40 - Myer Boland

"Finding Pi in BASIC" in V8N3P26 reported that both the TI-99/4A and the CC-40 returned the twelve digits 3.1415 92653 59 in response to the BASIC instruction P = 4*ATN(1) . Myer Boland reports that one can recover fourteen digits with the equation P = 4000*ATN(1) on the TI-99/4A, and I verify the same result with the CC-40:

Pi x 1000 exact = 3141.5 92653 58979 3 ...
4000*ATN(1) = 3141.5 92653 5898

Unfortunately, at least on the CC-40, if one tries to convert to the value of pi, not 1000xpi, by dividing the result by 1000, the end result reverts to the twelve digit value 3.1415 92653 59 . This is one more illustration of the kind of results which occur with BASIC, but which we would not expect with the typical calculator.

THE ROBERT PRINS BRAINTEASER - V9N2P13 and V9N3P12 described a brainteaser proposed by Robert Prins. The object was to obtain a flashing one in the display after starting from turn-on, but using a very restricted set of commands from the keyboard. Since no solutions have arrived, even with the hint in V9N3P12, here is the solution:

```
Turnon  R/S  LRN  GTO  CLR  INV  PRD  CLR  RCL  CLR
```

Now, Robert poses what he calls a "very difficult puzzle". Is it possible for the TI-59 to stop with a RTN while the subroutine register is not empty, and closely related to this question, is it possible to press a user defined key, SBR N, or SBR nnn from the keyboard without clearing the subroutine return register? As with the flashing one brainteaser, Robert has not provided the answer as yet, and I have been unsuccessful in obtaining an answer to date. Oh well -- I scored very poorly on the Ristanovic Supertest in V7N9 as well.

PC-200 AVAILABILITY - V9N3P22 reported that TI expected to have the PC-200 available later this year. Page 22 of the latest Educalc catalog (Issue # 22) indicates that the PC-200 will be available in the first quarter of 1985.

HARDWARE AVAILABILITY - The University of Michigan bookstore has a limited amount of TI-59 related hardware available. The available items and the unit prices are:

7	PC-100C Printers	$146.25
1	Business Decisions Module	$24.00
2	RPN Modules	$24.00
2	Leisure Library Modules	$24.00
2	Marine Navigation Modules	$24.00

Those prices do not include shipping. If you are interested, call 517-355-3454, and ask for Jody Mitchelson. I thank Dave Leising for telling me about this hardware.

BOB FRUIT'S BENCHMARK TEST - Many members have expressed interest in the results of these tests which compare the capability of various machines. In V8N4P4 Bob Fruit proposed that a savings accumulation problem might be a good vehicle for comparing numerical precision. The specific problem he proposed was a dollar contributed to a savings account every month, with interest compounded every month for thirty years. The appropriate equation is

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

An annual interest rate of ten per cent was assumed. The effective monthly interest rate is then 10%/12. Bob provided answers for various computers and calculators. V8N5P26 reported exact solutions from Laurance Leeds, George Thomson, and Carl Rabe. Solutions for additional machines were presented in V8N6P12. Since then Sterling Hartman provided solutions for Casio, HP, Sanyo and Atari, and Dave Leising reported a double precision solution on the Macintosh. The results reported to date are:

Exact Solution	2260.48792 47960 86067 64793
Radio Shack Model III (S.P.)	2260.29
Atari 800 (usoft BASIC S.P.)	2260.48
Atari 800 (usoft BASIC D.P.)	2260.46240 23437 5
Sanyo MBC 555 (S.P.)	2260.48
Sanyo MBC 555 (D.P.)	2260.48095 70312 5
Atari 600XL, 800, 1200XL	2260.48555 2
Apple II Plus	2260.48828 4
Commodore 64 or VIC-20	2260.48828 4
HP-11C or HP-41	2260.48764 1
TI-55-II, TI-57-LCD	2260.48772 43
Radio Shack Color Computer	2260.48801
TI-57	2260.48799 67
TI BA-55	2260.48782 43
Casio FX-450	2260.48988 6
TI-30	2260.4879
TI MBA	2260.48790.9
HP-9830	2260.48791 8
HP-9845	2260.48791 914
HP-85	2260.49792 196
Casio FX-700P	2260.48792 22
TI CC-40	2260.48792 41984
TI-99/4A	2260.48792 43288
TI-66	2260.48792 451
TI-58/59	2260.48792 4713
Radio Shack Model 100	2260.48792 47471
Macintosh (D.P.)	2260.48792 47472
IBM PC (Double Precision)	2260.48792 47960 93

For this evaluation we did not use some of the more exotic methods of calculation described in V8N4P4. Rather, we used straightforward evaluation of the formula using the raise-to-a-power function. Several members suggested that such a restriction discriminated unfairly against the HP-11 and HP-41 which could use a ln(1+x) technique to obtain greater precision. I do not have an HP-41. I could find no mention of the technique in the handbook which comes with the HP-11.

Bob Fruit's Benchmark Test - (cont)

I did find a technique described on page 181 of the HP-15C Advanced Functions Handbook. (I don't have an HP-15C either, but the Advanced Functions Handbook is an important reference to Kahan's paper "Mathematics Written in Sand ..."). The technique circumvents the loss of precision which occurs when 1 is added to x when x < 1. The (1+x)^n function in the equation above is replaced with e^(n*ln(1+x)) where a more accurate value for ln(1+x) is found in the following manner:

1. Find u = the rounded value of (1+x)

2. Calculate ln(1+x) = ln(u)*x/(u-1)

If u = 1, then ln(1+x) = x . Of course, if one uses this technique on the HP-11, it would be only fair to permit use of the same technique with other machines as well. Representative results are:

Exact Solution	2260.48792 47960 86067 64793
BA-55/TI-55-II/TI-57LCD	2260.48776 64
TI-57	2260.48802 48
Color Computer	2260.48803
HP-11	2260.48792 4
CC-40	2260.48792 44876
TI-59	2260.48792 4991
TI-66	2260.48792 4793
Model 100	2260.48792 47954

Results from several devices such as the Color Computer show little change. The results from the HP-11, the TI-66, and the Model 100 are much improved. The results from the TI-59 are degraded. Egregious, some might say, but for this problem even the degraded solution from the TI-59 has a relative error four times smaller than that for the best solution from the HP-11. For other problems where x is much smaller the TI-59 does not do so well.

Page 182 of the HP-15C Advanced Functions Handbook cautions that the ln(1+x) technique may be invalid on machines which calculate functions such as ln(u) with small absolute error, but large relative error. In a future issue we will examine the ln(x) function for various machines as the argument approaches one. In any case it would be well to proceed with caution when one uses the y^x function of the TI-59 when y is near 1. Of course that is precisely what the caution on page C-2 of Personal Programming says.

CALCULATOR REPAIR OR REPLACEMENT - Several members have asked about repair or replacement of their defective TI-59. The easiest method is to take the defective unit to one of the exchange centers located around the country. You turn in the defective unit and about $65.00 and receive a refurbished calculator. The exact price seems to vary from center to center. You can get the telephone number and address of the repair center in your area by calling 800-858-1802. You can also send your calculator to Texas Instruments, Inc., 2305 North University, Lubbock, Texas 79415. The current exchange price at that facility is $60.50.

MALFUNCTION DIAGNOSIS - The number of members reporting malfunctions of calculators and printers has been increasing. Perhaps that is not surprising. Much of our hardware is over five years old and may be showing wear; for example, A. Krufka was experiencing keyboard problems with his TI-59. When he took the calculator apart he found that the sheet of foam under the keys had round holes wherever the problem keys were located. He rearranged the foam so that the holes would not be under any of the round pins on the back of the keys. He reassembled the calculator and the keyboard problems were gone.

A valuable reference for working on your calculator is the Programmable TI58/59 Service Manual. The 62 page manual includes disassembly instructions, troubleshooting techniques, block diagrams, schematics, waveforms, diagnostic programs and parts lists. The manual contains some interesting insights. Consider this excerpt from page 23:

> "Check resonator: Replace if yellow, uncoated resonator is found. If rectangular resonator is used, perform "drop test" by: starting diagnostic program, dropping calculator approximately 6 inches onto a hard surface while program is still running. If calculator goes into pre-load or gets wrong answer to diagnostic program, replace resonator."

I wouldn't do that test intentionally with my TI-59. A friend did, starting with a drop of about an inch, and gradually increasing the drop to over six inches. The calculator performed flawlessly throughout.

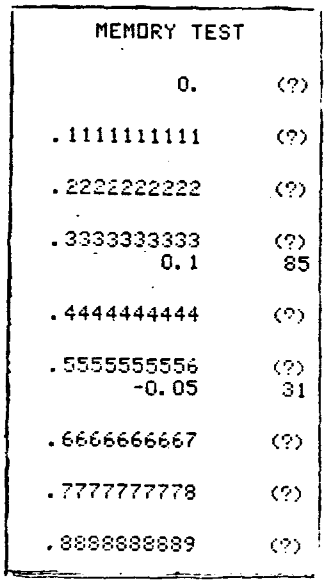
Page 43 of the manual contains a memory malfunction diagnostic program which stores 1/9 (all ones) to memory registers 01 through 99, sums -1/9 into the memory registers, checks for zero in each register, and prints out the register contents and register number for registers which fail. When I found that some defective memories would pass this test I wrote an extension which would exercise the memory registers with values other than 1/9. The modified program uses the test values 0, 1/9, 2/9, etc., through 8/9. To date I have not found a calculator with a defective memory register which will pass this test. To run the program, simply load the program listed on the next page, press A, and wait about sixteen minutes. The use of each test value is indicated by a printout of the value with a question mark in parentheses at the right. Test failures are indicated by printing the register contents and register numbers. A sample printout appears to the right of the program listing. The indicated failures at R85 with a test value of 1/3 and at R31 with a test value of 5/9 were induced by stopping the calculator at the appropriate time and summing an additional value into the register before the test for zero contents.

V3N9P6 of 52 Notes indicated that the service manual was available from TI for $11.95 each plus $1.50 handling and postage. That was September 1978. You may not be able to obtain a copy of the service manual now. I will loan my copy to club members. In the U.S. send two dollars to cover postage and handling. I will send the manual by first class mail and expect it to be returned the same way.

Malfunction Diagnosis - (cont)

Program Listing - Memory Malfunction Diagnostic

```
000  76 LBL    040  09  9     080  00  00    120  59 INT
001  11  A     041  09  9     081  29 CP     121  65  ×
002  69 OP     042  42 STO    082  67  EQ    122  01  1
003  00  00    043  00  00    083  01  01    123  00  0
004  03  3     044  98 ADV    084  39  39    124  54  )
005  08  8     045  05  5     085  32 X:T    125  85  +
006  69 OP     046  05  5     086  53  (     126  01  1
007  01  01    047  07  7     087  53  (     127  85  +
008  01  1     048  01  1     088  53  (     128  28 LOG
009  07  7     049  05  5     089  43 RCL    129  59 INT
010  03  3     050  06  6     090  00  00    130  65  ×
011  00  0     051  69 OP     091  55  ÷     131  02  2
012  03  3     052  04  04    092  01  1     132  54  )
013  02  2     053  32 X:T    093  00  0     133  95  =
014  03  3     054  69 OP     094  54  )     134  69 OP
015  05  5     055  06  06    095  59 INT    135  04  04
016  04  4     056  72 ST*    096  85  +     136  32 X:T
017  05  5     057  00  00    097  01  1     137  69 OP
018  69 OP     058  97 DSZ    098  85  +     138  06  06
019  02  02    059  00  00    099  28 LOG    139  97 DSZ
020  03  3     060  00  00    100  59 INT    140  00  00
021  07  7     061  56  56    101  65  ×     141  00  00
022  01  1     062  32 X:T    102  02  2     142  79  79
023  07  7     063  09  9     103  54  )     143  01  1
024  03  3     064  09  9     104  65  ×     144  82 HIR
025  06  6     065  42 STO    105  01  1     145  37  37
026  03  3     066  00  00    106  00  0     146  32 X:T
027  07  7     067  32 X:T    107  00  0     147  82 HIR
028  69 OP     068  94 +/-    108  54  )     148  17  17
029  03  03    069  74 SM*    109  85  +     149  55  ÷
030  69 OP     070  00  00    110  53  (     150  09  9
031  05  05    071  97 DSZ    111  53  (     151  95  =
032  01  1     072  00  00    112  53  (     152  22 INV
033  00  0     073  00  00    113  43 RCL    153  77  GE
034  69 OP     074  69  69    114  00  00    154  00  00
035  17  17    075  09  9     115  55  ÷     155  38  38
036  29 CP     076  09  9     116  01  1     156  06  6
037  37 P/R    077  42 STO    117  00  0     157  69 OP
038  47 CMS    078  00  00    118  54  )     158  17  17
039  32 X:T    079  73 RC*    119  22 INV    159  91 R/S
```



```
        MEMORY TEST

               0.        (?)

     .1111111111         (?)

     .2222222222         (?)

     .3333333333         (?)
             0.1          85

     .4444444444         (?)

     .5555555556         (?)
           -0.05          31

     .6666666667         (?)

     .7777777778         (?)

     .8888888889         (?)
```

MATTESON'S PSEUDO RANDOM NUMBER GENERATOR - George Wm. Thomson

The editor noted that Lem Matteson's game Minefield III included a compact pseudo random number generator for the TI-59 (V9N3P24-25). The seed is stored in R09.

Part I: RCL 09 Lnx |x| INV INT STO 09 (the new seed)

Part II: x 4 INV Lnx INV INT x 10 = INT (output, Pseudo R.N., 0-9)

Investigation shows that this is a very good pseudo random number generator. To test it for the range from zero to just below one I wrote a short fast mode program with a sum and sum of squares at the end of part I and similar steps after the INV INT in part II. At the end of the run I calculated the mean and the "n-1" standard deviation estimate for both of the quantities. The theoretical valuesfor both the mean and standard deviation of this rectangular distribution are 0.5 . For two series of 5000 values, one with the starting seed equal to pi-squared and the other with starting seed equal to 12345 the results were very good, although the mean for the seeds was somewhat low. Part II provided an excellent correction.

	Pi-squared		12345	
	seed	R.N.	seed	R.N.
mean	0.44446	0.49721	0.44207	0.50360
std deviation	0.49696	0.50004	0.49668	0.50003

Congratulations, Lem. You did it again.

NEWCOMER'S CORNER - The Use of Editing Commands in Programs

V9N3P10 commented that the function as a dummy operator which would not clear an existing error state was the only known use for INS (code 46) in a program. Several members wrote that there was another use for INS, and uses for the other editing commands as well, in a program. Let's look at some history.

The May 1978 issue of PPX Exchange stated:

> "Mr. Peter K. Buckley shares the following: The keys 2nd, LRN, SST, and BST can be used as common labels but must be written into the program. For example, to use SST as a label, check the keycode (i.e., 41) and key in (in LRN mode) Lbl STO STO 41 ... and delete the two STO instructions."

But the September 1978 issue of PPX Exchange corrected that statement by reporting:

> "It has been brought to our attention that SST, BST, LRN, and 2nd are not fully functional labels. ..."

Earlier, in November 1977 V2N11P4 of 52 Notes had reported:

> "As the manual says: LRN, Ins, Del, SST, BST and Ind are not valid labels; HIR (code 82) is the only pseudo that is. ... "

Let's consider at IND (code 40) first. IND can be used as a dummy operator to avoid setting an existing error condition (V5N6P3) in the same manner as INS (code 46). A program sequence which will list as Lbl IND can be generated, but there seems to be no way to branch to that location using a common label format. The code 40 is always interpreted to indicate a following indirect address; either in accordance with the table on page V-68 of Personal Programming, or with the observation that a program sequence such as GTO IND is interpreted as GO* (V5N8P2).

Although the editing commands do not work as labels several other uses in programs have been identified. V3N2P6 of 52 Notes reported:

> "Rusty Wright discovered that the SR-52 LBL LBL tricks can be made to work on the new machines if each Lbl Lbl sequence is preceded by the SST pseudo (code 41). Jared's flag reversal routine for the SR-52 can be written ...Ifflg 0 SST Lbl Lbl INV Stflg 0... for the new machines."

Tests will show that the flag reversal routine works equally well with 2nd (code 21) or 2nd-2nd (code 26) in place of the SST (code 41). But with Ins (code 46), BST (code 51) or Del (code 56) in place of the SST the routine will not work. Consider another demonstration routine starting at step 000

```
1 2 3 4 5 Lbl Nop 6 7 8 9 R/S Lbl A GTO SST Nop
```

Pressing A yields "6789" in the display since the calculator executes a GTO Nop, ignoring the SST. Insert two zeroes before the last Nop, press A and see "123456789" in the display. This time the calculator executed a GTO 000. We see that Codes 21, 26 and 41 act like true "no-op's". I have not found examples of this use of 2nd, 2nd-2nd and SST in published programs.

Editing Commands in Programs - (cont)

There are also uses for Ins, BST, or Del in programs. V3N9P5 of 52 Notes reported:

> "Maurice Swinnen passes along a tip from the Jan-Apr issue of DISPLAY (in German, due to S. Seitz) suggesting a sequence of this form: Dsz mn ab where register mn is to be decremented, and ab is the code for one of the Ins, Del, or BST pseudos. These pseudos are always skipped during program execution, effectively turning the Dsz mn into Op 3mn, except that zero is the lower decrement limit."

As with the normal use of the Dsz command mn may be any memory register other than 40. Of course, the Op 3n sequence provides decrementing in only two program steps for memory registers 00 through 09, and the 1 INV SUM mn sequence provides decrementing in four program steps for memory registers from 10 to 99. Repeated calls to either sequence will eventually drive the register contents through zero. If, for some reason, the program cannot tolerate negative values in the register then additional steps would be needed to test the contents before decrementing. In that case the Dsz mn ab sequence would provide the desired effect in only three steps. Examples of this technique appear at steps 450-452 of the Worthington/Regelman Inverse List Print All program (V5N9P16) and at steps 049-051 of the TI-59 Test program in the May/June 1982 issue of PPX Exchange.

Another use for editing commands in programs is the branching from the keyboard technique originally devised by Martin Neef (V5N7P11), and refined by Dejan Ristanovic (V6N9/10P31 and V8N2P2). Dejan's sequence is Pgm 09 SBR 058 Pgm 09 Bst with the Master Library module in place. The Pgm 09 SBR 058 sequence simply exercises the RTN at location 058 of ML-09 and control returns to user memory. The program counter has been set up for a return to location 059 with a Pgm 09 R/S sequence (see V8N6P4 for examples). Dejan's technique uses the sequence Pgm 09 BST instead. Program execution continues on without jumping to the library routine, but the calculator has been "armed". An R/S from the keyboard does not stop calculations; rather, a operator-controlled transfer to step 059 of ML-09 occurs. The library program sequence starting at step 059 is

```
     Lbl D  E  Pgm 00 A'  where subroutine E is

     Lbl E  ( RCL 01 + RCL 05 x RCL 03 ) RTN
```

Subroutine E is performed and the sequence Pgm 00 A' calls a subroutine A' in user memory. Subroutine A' sets up the desired response to the interrupt. A RST is typically included in subroutine A' to clear the subroutine return registers. Examples of the technique appear in steps 096-103 of the demonstration program on V6N9/10P31 and at steps 288-293 and 385-392 of the Supertest TI-59 scoring program on V7N9P11. The demonstration program on 6N9/10P31 does not provide a Lbl A' routine in user memory; as a result, the program "hangs up" in library memory after an interrupt such that a RST R/S sequence from the keyboard is needed to continue on. V5N7P11 reports that 2nd (code 21) and SST (code 41) can be used in place of BST, but there may be side effects. V5N7P11 also contains a list restrictions generated by Richard Snow.

PLOT 60 PLUS - Barry W. Widman. This program is an enhanced version of Michael Sperber's Plot 60 program which appeared in V6N4-5P5. In addition to plotting functions, this program also

- plots data (up to 19 points).
- provides vertical and horizontal boundaries (every other point horizontally and every fifth point vertically).
- provides an option to print functions in an "every-other-point" mode in addition to the "every-point" mode.

Data can be entered using the ST-04 program from the Applied Statistics memory module, or entered directly by storing the x,y data pairs in R32 through R59 (14 datapairs maximum) and storing the number of the register following the last y value in R31.

If data is not being plotted, or if the data consists of nine or fewer points, the calculator may be repartitioned to 719.29 or 559.49 respectively, to permit more room for defining functions to be plotted.

If functions to be plotted do not extend past step 399, the calculator may be repartitioned to 399.69 to permit more data points to be plotted. (up to 19 data points maximum).

The basic version of Plot 60 from V6N4-5P5 has been modified as follows:

- A Nop instruction has been deleted from step 027 and inserted at step 001. I feel that this gives the resulting plot a more pleasing appearance.
- The data which had been stored in registers R00 through R06 have been moved to registers R20 through R26. This allows the ST-04 program to accumulate data in the statistics registers (R00-R06) as well in the raw data registers (R32-R59). As illustrated in the example used to generate the upper plot on page 1, this permits the "best-fit" line to be plotted with the Op 14 command. This change also provides the potential to use MU-08 from the Math/Utilities module for storing and retrieving larger amounts of data for plotting.
- The plotting parameter entry routine (Lbl E) has been expanded to store the top and bottom lines of the graph in R17 and R18 respectively. R17 corresponds to Y min and R18 corresponds with the uppermost tick mark. Note that the program will plot four points above the uppermost tick mark should data or functions exceed the R18 value.
- Because of the above changes, the location for defining functions to be plotted is changed from step 224 to 252. However, in order to provide the capability to plot data, boundaries and "even-point" plots the following additional memory assignments were made:

  Locations 252-271: test for first and last print line.

  Locations 272-306: print twelve vertical tick marks.

  Locations 307-357: plots data from R32 through R59.

  Locations 358-373: "even-point" plotting.

  Locations 374- : define functions to be plotted. The address in steps 372 and 373 specifies the beginning of the "all-point" functions. "Even-point" functions are defined between step 375 and this address. At least one "all-point" function is required. Subroutine D must be specified after the last function, and the entire routine must end with GTO 252.

Plot 60 Plus - (cont)

Program Listing:

Before Initialization

```
000   92  RTN
001   68  NOP
002   76  LBL
003   15   E
004   42  STO
005   26   26
006   92  RTN
007   61  GTO
008   01   01
009   83   83
010   76  LBL
011   11   A
012   61  GTO
013   00   00
014   33   33
015   76  LBL
016   14   D
017   11   A
018   61  GTO
019   01   01
020   35   35
021   25  CLR
022   69  OP
023   05   05
024   74  SM*
025   80   80
026   02   2
027   68  NOP
028   68  NOP
029   68  NOP
030   68  NOP
031   68  NOP
032   22  INV
033   58  FIX
034   75   -
035   43  RCL
036   21   21
037   95   =
038   55   ÷
039   01   1
040   32  X:T
041   43  RCL
042   22   22
043   65   ×
044   77   GE
045   01   01
046   30   30
047   29  CP
048   22  INV
049   77   GE
050   01   01
051   30   30
052   04   4
053   85   +
054   09   9
055   75   -
056   59  INT
057   42  STO
058   13   13
059   95   =
060   65   ×
061   05   5
062   42  STO
063   07   07
064   85   +
065   59  INT
066   94  +/-
067   44  SUM
068   07   07
069   95   =
070   65   ×
071   03   3
072   95   =
073   52  EE
074   59  INT
075   67   EQ
076   00   00
077   87   87
078   32  X:T
079   02   2
```

```
080   22  INV
081   67   EQ
082   00   00
083   90   90
084   94  +/-
085   85   +
086   02   2
087   07   7
088   95   =
089   65   ×
090   93   .
091   00   0
092   01   1
093   95   =
094   44  SUM
095   13   13
096   73  RC*
097   13   13
098   55   ÷
099   43  RCL
100   07   07
101   22  INV
102   28  LOG
103   33  X²
104   52  EE
105   22  INV
106   57  ENG
107   42  STO
108   07   07
109   95   =
110   88  DMS
111   82  HIR
112   18   18
113   59  INT
114   29  CP
115   22  INV
116   67   EQ
117   01   01
118   30   30
119   43  RCL
120   07   07
121   65   ×
122   43  RCL
123   13   13
124   22  INV
125   59  INT
126   95   =
127   74  SM*
128   13   13
129   25  CLR
130   29  CP
131   43  RCL
132   25   25
133   92  RTN
134   43  RCL
135   09   09
136   69  OP
137   01   01
138   43  RCL
139   10   10
140   69  OP
141   02   02
142   43  RCL
143   11   11
144   69  OP
145   03   03
146   43  RCL
147   12   12
148   69  OP
149   04   04
150   71  SBR
151   00   00
152   20   20
153   71  SBR
154   02   02
155   41   41
156   43  RCL
157   24   24
158   44  SUM
159   25   25
```

```
160   25  CLR
161   97  DSZ
162   20   20
163   01   01
164   30   30
165   98  ADV
166   25  CLR
167   43  RCL
168   08   08
169   99  PRT
170   43  RCL
171   22   22
172   44  SUM
173   21   21
174   98  ADV
175   71  SBR
176   02   02
177   32   32
178   97  DSZ
179   08   08
180   00   00
181   00   00
182   81  RST
183   42  STO
184   21   21
185   42  STO
186   22   22
187   42  STO
188   17   17
189   92  RTN
190   22  INV
191   44  SUM
192   22   22
193   92  RTN
194   42  STO
195   23   23
196   92  RTN
197   42  STO
198   24   24
199   92  RTN
200   42  STO
201   08   08
202   94  +/-
203   22  INV
204   49  PRD
205   22   22
206   06   6
207   00   0
208   65   ×
209   43  RCL
210   08   08
211   75   -
212   05   5
213   95   =
214   55   ÷
215   53   (
216   46  INS
217   85   +
218   04   4
219   54   )
220   65   ×
221   43  RCL
222   08   08
223   65   ×
224   43  RCL
225   22   22
226   85   +
227   43  RCL
228   21   21
229   95   =
230   42  STO
231   18   18
232   43  RCL
233   26   26
234   42  STO
235   20   20
236   43  RCL
237   23   23
238   42  STO
239   25   25
```

```
240   32  X:T
241   00   0
242   42  STO
243   09   09
244   42  STO
245   10   10
246   42  STO
247   11   11
248   42  STO
249   12   12
250   32  X:T
251   92  RTN
252   32  X:T
253   43  RCL
254   23   23
255   67   EQ
256   02   02
257   72   72
258   85   +
259   43  RCL
260   24   24
261   65   ×
262   53   (
263   43  RCL
264   26   26
265   75   -
266   01   1
267   95   =
268   22  INV
269   67   EQ
270   03   03
271   07   07
272   43  RCL
273   22   22
274   55   ÷
275   01   1
276   01   1
277   93   .
278   08   8
279   95   =
280   42  STO
281   16   16
282   65   ×
283   01   1
284   02   2
285   85   +
286   43  RCL
287   21   21
288   95   =
289   42  STO
290   15   15
291   01   1
292   02   2
293   42  STO
294   14   14
295   43  RCL
296   16   16
297   22  INV
298   44  SUM
299   15   15
300   43  RCL
301   15   15
302   11   A
303   97  DSZ
304   14   14
305   02   02
306   95   95
307   22  INV
308   87  IFF
309   01   01
310   03   03
311   58   58
312   03   3
313   02   2
314   48  EXC
315   31   31
316   42  STO
317   30   30
318   73  RC*
319   31   31
```

```
320   32  X:T
321   01   1
322   44  SUM
323   31   31
324   43  RCL
325   25   25
326   85   +
327   43  RCL
328   24   24
329   55   ÷
330   02   2
331   95   =
332   22  INV
333   77   GE
334   03   03
335   46   46
336   75   -
337   43  RCL
338   24   24
339   95   =
340   77   GE
341   03   03
342   46   46
343   73  RC*
344   31   31
345   11   A
346   01   1
347   44  SUM
348   31   31
349   43  RCL
350   30   30
351   32  X:T
352   43  RCL
353   31   31
354   22  INV
355   67   EQ
356   03   03
357   18   18
358   43  RCL
359   26   26
360   75   -
361   43  RCL
362   20   20
363   95   =
364   55   ÷
365   02   2
366   95   =
367   22  INV
368   59  INT
369   29  CP
370   22  INV
371   67   EQ
372   03   03
373   80   80
374   43  RCL
375   17   17
376   11   A
377   43  RCL
378   18   18
379   11   A
380   43  RCL
381   25   25
382   69  OP
383   14   14
384   14   D
385   61  GTO
386   02   02
387   52   52
```

After

```
000   92  RTN
001   68  NOP
002   76  LBL
003   15   E
004   42  STO
005   26   26
006   92  RTN
007   61  GTO
008   01   01
009   83   83
010   76  LBL
011   11   A
012   61  GTO
013   00   00
014   33   33
015   76  LBL
016   14   D
017   11   A
018   61  GTO
019   01   01
020   35   35
021   25  CLR
022   69  OP
023   05   05
           0
025   92  RTN
026   42  STO
027   17   17
028   68  NOP
029   69  OP
030   68   68
031   97  DSZ
032   68   68
033   22  INV
034   58  FIX
035   75   75
036   43  RCL
037   21   21
038   95   =
039   55   ÷
040   01   1
041   32  X:T
042   43  RCL
043   22   22
044   65   ×
045   77   GE
046   01   01
047   30   30
048   29  CP
049   22  INV
050   77   GE
051   01   01
052   30   30
053   04   4
054   85   +
055   09
056   7
057
05
```

Plot 60 Plus - (cont)

Instructions:

1. Enter the data to be plotted, if any, using ST-04, data stored on magnetic cards, or a user defined program. Remember that the x,y pairs are stored in R32 and the following registers, and that the number of the register following the last y value must be in R31. The ST-04 routine does this automatically, and in addition accumulates the sums needed for linear regression. The following program in user memory will accomplish the same results as ST-04:

```
000  76 LBL   007  02   2   014  31  31   021  76 LBL   028  44 SUM
001  10 E'    008  42 STO   015  99 PRT   022  12  B    029  31  31
002  36 PGM   009  31  31   016  32 X:T   023  72 ST*   030  43 RCL
003  01  01   010  91 R/S   017  01   1   024  31  31   031  03  03
004  71 SBR   011  76 LBL   018  44 SUM   025  99 PRT   032  91 R/S
005  25 CLR   012  11  A    019  31  31   026  78 Σ+
006  03   3   013  72 ST*   020  91 R/S   027  01   1
```

Press E to initialize and clear the statistics registers. Enter the first x value and press A. Enter the corresponding y value and press B. A "1" will be returned to the display. Repeat the entry of x and y values as many times as required. Each time, the number of data pairs entered is returned to the display.

2. Enter the magnetic cards (banks 1 and 2) for the program before initialization from page 15. As it stands, this program will plot the regression line and the data using the Op 14 command. If you want to mechanize other functions enter them at this time. If you add even-point functions, be sure to adjust the address at 372/373 to include the first address of the all-point functions. You may use equal signs in your functions. You may enter as many functions as program space permits, but the subroutine call A must follow each function, except the last function, which must be followed by a call to subroutine D. The X value is available from R25 if you need it in one or more of your functions. End the entire routine with GTO 252.

3. The following initialization routine generates the hexadecimal code h25 at step 024 for high resolution graphics. If you used ST-04 for data entry, remember to press RST to leave the library routine.

GTO 024 10 Op 17 CLR <see below> P/R LRN INS LRN RST CLR 6 Op 17 CLR

The keystrokes to be inserted at <see below> depend upon which Solid State module is installed:

Master Library	Pgm 19 SBR 045
Applied Statistics	Pgm 14 SBR 024
Math/Utilities	Pgm 06 SBR 077
Real Estate/Investment	Pgm 10 SBR 039

When initialization is complete steps 24 through 32 will have been changed as indicated in the "After" column on page 15. The commands previously in steps 32 through 158 will have been moved down one step. The 25 in step 159 will have been deleted, and the CLR which was at step 160 will have been converted to the second part of a Sum 25 command. Although a listing shows no line number and a zero at what should have been step 24, if you press GTO 024 LRN you will see a 024 25 in the display.

3. If you want to plot data press Stflg 1.

4. Enter the parameters that define the dimensions of the plot:

   a. Enter the number of points to be plotted and press E. See the number returned to the display.

Plot 60 Plus - (cont)

b. Enter the minimum y value (Ymin) and press R/S. See Ymin returned to the display.

c. Enter the maximum y value (Ymax) and press R/S. See Ymax returned to the display.

d. Enter the x starting point (Xo) and press R/S. See Xo returned to the display.

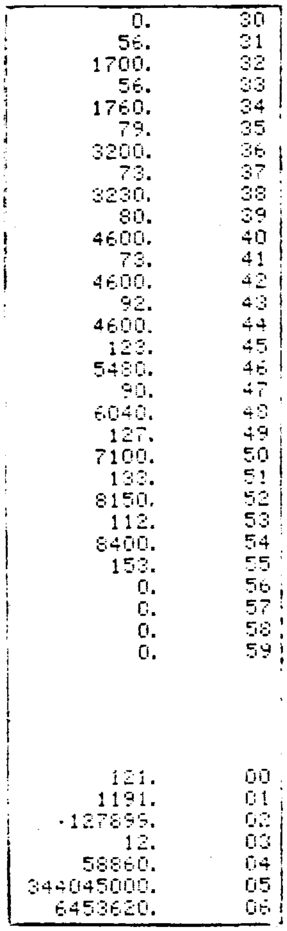
e. Enter the increment in x (delta-x) and press R/S. See delta-x returned to the display.

f. Enter the number of tapes to be used (n) and press R/S. See Xo returned to the display.

g. Press R/S to start plotting.

Sample Problem:

The sample problem uses the program on page 15 to plot both the input data and the linear regression line for a set of twelve data pairs. The data may be entered with ST-04 or with the short user program on page 16. After completion of the data entry the INV List will show the values at the right. Note that there are 3 data pairs where the x value is the same (4600). This will demonstrate the ability of the Plot 60 Plus program to plot multiple y values for a given x value. Although the table at the right is arranged with x values in ascending order, this is not required. Plot 60 Plus will faithfully plot x,y data pairs which have been entered with no attention to the order of the x values.



```
       0.     30
      56.     31
    1700.     32
      56.     33
    1760.     34
      79.     35
    3200.     36
      73.     37
    3230.     38
      80.     39
    4600.     40
      73.     41
    4600.     42
      92.     43
    4600.     44
     123.     45
    5480.     46
      90.     47
    6040.     48
     127.     49
    7100.     50
     133.     51
    8150.     52
     112.     53
    8400.     54
     153.     55
       0.     56
       0.     57
       0.     58
       0.     59



     121.     00
    1191.     01
  -127695.    02
      12.     03
   58860.     04
344045000.    05
  6453620.    06
```

The plotting parameters used to obtain the upper plot on page 1 were entered as follows (step 4 above):

121 E 0 R/S 238 R/S 0 R/S 100 R/S 2 R/S R/S

The rationale for selecting those parameters follows. For the most pleasing plot appearance the number of x axis points to be plotted should be between about 50 to 80 times the number of tapes. Also, the increment in x and the increment in y should be equal to some integer (preferably 1, 2, or 5) times a factor of 10. In this example the x values range between 1700 and 8400, and we wish to start the plot at 0,0 and extrapolate the linear regression line to 12,000. For the two tape plot, select delta-x = 100, then the number of points = (Xmax - Xmin)/(delta-x) + 1, which for this case is 121. With this program the X axis (bottom line) actually shows 61 tick marks since it is produced as an "even-point" function. That is, only points 0, 2, 4, ... , 120 are plotted. The first point that is plotted is considered to be the 0 point.

For the Y axis the number of increments that can be plotted is fixed by the number of tapes selected, that is 60 times the number of tapes. Selecting a delta-y of 2, yields a range of Y values of 238. Since we wish to start the plot at 0,0, then Ymin = 0, and Ymax = 238, which comfortably fits the range of the expected y values.

Since there are five intervals between each vertical tick mark, the vertical tick marks can be labeled in increments of ten. Similarly, for the horizontal axis which plots tick marks for every other x increment, the tick marks correspond to values of 0, 200, 400, ... , 12000.

Plot 60 Plus - (cont)

Editor's Note: The use of the data storage capability of the Applied Statistics module provides an inherent versatility for this program. The ST-12 program, Bivariate Data Transforms, provides built-in transforms for converting input data with exponential, power, and logarithmic characteristics for analysis by linear regression. The storage locations are the same as for ST-04, and the necessary summations for the linear regression are obtained as part of the data entry. However, the program stores the transformed data, not the input data. This means that the use of Plot 60 Plus as is would yield plots in the transformed coordinates, not in the input coordinates. Users frequently want plots in the input coordinate system.

The program listing at the right illustrates the changes required to Plot 60 Plus for use with the exponential curve fit option, and for plotting in input coordinates. Steps before 307 are the same as on page 15, and the user instructions are identical, except that ST-12 is used instead of ST-04. Steps 345/346 convert the stored ln(y) values back to input values for plotting. The insertion of those steps requires several changes in absolute addresses. Steps 382 through 391 convert the solution in transformed coordinates back to input coordinates for plotting the best fit exponential line.

```
307  22 INV     351  43 RCL
308  87 IFF     352  30  30
309  01  01     353  32 X:T
310  03  03     354  43 RCL
311  60  60     355  31  31
312  03  3      356  22 INV
313  02  2      357  67  EQ
314  48 EXC     358  03  03
315  31  31     359  18  18
316  42 STO     360  43 RCL
317  30  30     361  26  26
318  73 RC*     362  75  -
319  31  31     363  43 RCL
320  32 X:T     364  20  20
321  01  1      365  95  =
322  44 SUM     366  55  ÷
323  31  31     367  02  2
324  43 RCL     368  95  =
325  25  25     369  22 INV
326  85  +      370  59 INT
327  43 RCL     371  29 CP
328  24  24     372  22 INV
329  55  ÷      373  67  EQ
330  02  2      374  03  03
331  95  =      375  82  82
332  22 INV     376  43 RCL
333  77  GE     377  17  17
334  03  03     378  11  A
335  48  48     379  43 RCL
336  75  -      380  18  18
337  43 RCL     381  11  A
338  24  24     382  69 OP
339  95  =      383  12  12
340  77  GE     384  85  +
341  03  03     385  32 X:T
342  48  48     386  65  X
343  73 RC*     387  43 RCL
344  31  31     388  25  25
345  22 INV     389  95  =
346  23 LNX     390  22 INV
347  11  A      391  23 LNX
348  01  1      392  14  D
349  44 SUM     393  61 GTO
350  31  31     394  02  02
                395  52  52
```

The demonstration problem is taken from page 5-11 of the Applied Statistics manual. Population versus census year is given as:

1890	62,947,714	1920	105,710,620	1950	150,697,361
1900	75,994,575	1930	122,775,046	1960	179,323,175
1910	91,972,266	1940	131,669,275	1970	203,235,298

The data is entered in accordance with the ST-12 instructions. If you do not have the Applied Statistics module you may use the user program from the top of page 16, but with a lnx command inserted immediately after the Lbl B, and before the ST* 31. Then enter the data as defined for that program. The appropriate transformation and summations will be made for equivalence with ST-12. Initialization procedures for Plot 60 Plus are unchanged. The plotting parameters to obtain the lower curve on page 1 were:

111 E 50000000 R/S 290000000 R/S 1880 R/S 1 R/S 2 R/S R/S.

These parameters were specifically chosen to demonstrate the feature that four points can be plotted above the upper tick marks.

Similar alterations to Plot 60 Plus can be defined for compatibility with the power and exponential options of ST-12. It would seem that an interface with the RE-11 Automatic Curve Choice program in the Real Estate/Investment module should also be possible. Readers are invited to sent in additional demonstrations of the interface of this powerful program with other plotting requirements.

1984 FEDERAL INCOME TAX JOINT RETURN - Hewlett Ladd

```
  25000.     T.I.
   3565.      TAX
     25.     TOP%
   14.26     AV.%

   6000.     T.I.
    291.      TAX
     12.     TOP%
    4.85     AV.%
```

This fast mode program accepts "Taxable Income" and returns a printout including the input value, the tax, the top percentage bracket, and the average percentage. The instructions are straightforward. Enter the taxable income and press A. See a flashing "1" in the display. Press 7 and then EE and wait for the printout. Some sample printouts appear at the right.

The constants in R01 through R29 establish that this program finds the income tax for a joint return. Straightforward replacement of the constants can be used to change the calculations to head-of-household or married filing separately. A problem occurs for the single taxpayer schedule since there is one more bracket there. The user must elect to omit the lowest or the highest bracket.

Program Listing:

```
000  91 R/S    040  00  00    080  06  06    120  95  =         0.          00
001  25 CLR    041  59 INT    081  43 RCL    121  42 STO      3400.         01
002  43 RCL    042  95  =     082  97  97    122  31  31      5500.00231    02
003  14  14    043  42 STO    083  69 OP     123  43 RCL      7600.00483    03
004  59 INT    044  31  31    084  04  04    124  14  14     11900.01085    04
005  22 INV    045  01  1     085  43 RCL    125  22 INV     16000.01741    05
006  77  GE    046  05  5     086  32  32    126  59 INT     20200.02497    06
007  01  01    047  44 SUM    087  65  ×     127  52 EE      24600.03465    07
008  14  14    048  00  00    088  01  1     128  05  5      29900.0479     08
009  01  1     049  73 RC*    089  00  0     129  95  =      35200.06274    09
010  42 STO    050  00  00    090  00  0     130  22 INV     45800.09772    10
011  00  00    051  42 STO    091  95  =     131  52 EE      60000.15168    11
012  73 RC*    052  32  32    092  69 OP     132  42 STO     85600.2592     12
013  00  00    053  65  ×     093  06  06    133  30  30    109400.3663     13
014  59 INT    054  43 RCL    094  43 RCL    134  43 RCL    162400.626      14
015  77  GE    055  31  31    095  96  96    135  31  31         0.          15
016  00  00    056  95  =     096  69 OP     136  65  ×          0.11        16
017  23  23    057  44 SUM    097  04  04    137  43 RCL         0.12        17
018  69 OP     058  30  30    098  43 RCL    138  29  29         0.14        18
019  20  20    059  43 RCL    099  33  33    139  42 STO         0.16        19
020  61 GTO    060  99  99    100  65  ×     140  32  32         0.18        20
021  00  00    061  69 OP     101  01  1     141  95  =          0.22        21
022  12  12    062  04  04    102  00  0     142  44 SUM         0.25        22
023  69 OP     063  32 X:T    103  00  0     143  30  30         0.28        23
024  30  30    064  69 OP     104  95  =     144  61 GTO         0.33        24
025  73 RC*    065  06  06    105  58 FIX    145  00  00         0.38        25
026  00  00    066  35 1/X    106  02  02    146  59  59         0.42        26
027  22 INV    067  65  ×     107  69 OP     147  00  0          0.45        27
028  59 INT    068  43 RCL    108  06  06    148  76 LBL         0.49        28
029  52 EE     069  30  30    109  22 INV    149  11  A          0.5         29
030  05  5     070  95  =     110  58 FIX    150  32 X:T         0.          90
031  95  =     071  42 STO    111  98 ADV    151  01  1          0.          91
032  22 INV    072  33  33    112  66 PAU    152  00  0          0.          92
033  52 EE     073  43 RCL    113  81 RST    153  69 OP          0.          93
034  42 STO    074  98  98    114  32 X:T    154  17  17         0.          94
035  30  30    075  69 OP     115  75  -     155  04  4          0.          95
036  32 X:T    076  04  04    116  32 X:T    156  05  5   13424061.          96
037  75  -     077  43 RCL    117  43 RCL    157  30 TAN  37323361.          97
038  32 X:T    078  30  30    118  14  14    158  33 X²     371344.          98
039  73 RC*    079  69 OP     119  59 INT    159  86 STF  37402440.          99
```

BASIC TO TI-59 COMPILER - Page 6508 of Volume 83, Number 26 of

Government Reports announcements announces the availability of AD-A132 172/8, "Design and Implementation of a Basic Cross-Compiler and Virtual Memory Management System ,for the TI-59 Programmable Calculator". The 304 page document was a master's thesis at the Naval Postgraduate School, Monterey, CA. The authors are Mark R. Kindl and James H. W. Inskeep, Jr. A hard copy is $25.00. The abstract appears at the right. I have not obtained a copy. Based on some recent letters there might be more interest in TI-59 to Basic.

The instruction set of the TI-59 Programmable Calculator bears a close similarity to that of an assembler. Though most of the calculator instructions perform primitive data movement and/or sequence control, some can do the work of small high level language procedures. Regardless of this fact, to design and debug TI-59 programs of moderate size can be more difficult than doing the computations themselves. Programming in a higher order language such as BASIC offers many advantages over calculator code. This report presents the design and implementation of a cross-compiler which translates correct BASIC programs into equivalent TI-59 programs. This software package includes a linker which maps calculator instructions to a set of magnetic cards. The cards are then used to implement a manually operated virtual memory system for the calculator. This expands program step capacity, and permits more complex programs to be written in BASIC language for translation into TI-59 instructions. (Author)

YOT RACE - Dave Lane. Captain your own yacht in a 100 mile ocean race. There may be five other boats racing toward the finish buoy, but the wind varies with position, so you'll have to find the best course. Wind shifts cause you, and the others, to renavigate. Sudden storms may completely change the position and order of the boats, but the race must go on. The effectiveness of the wind versus wind angle is included in the dynamics of the boat. So realizing that the boat will cut a faster course when going abeam and can't go directly into the wind, you'll have to choose your tack to keep ahead of your fellow yacht-persons. You can be a multi-millionaire and race all six boats yourself; or six players can race one boat each; or any combination of players and boats. However, the players must control every boat. The calculator simply carries out the players commands.

Rules of the Game:

1. The area of sailing is a square 100 miles on a side. Coordinates are given as N.E. For example, 50.25 indicates 50 miles north and 25 miles east of the origin.

2. The race is 100 miles in a northerly direction. You start the race at an N-value of zero, and at an E-value randomly selected by the program. The N-value of the finish buoy is 100, with the E-value selected by the program. You must cross the finish line between the buoy and a point 10 miles east of the buoy.

3. At each turn, you are given the wind velocity and direction. You simply have to choose the direction to go. The choices are West, Northwest, North, Northeast, and East. You can't go southward.

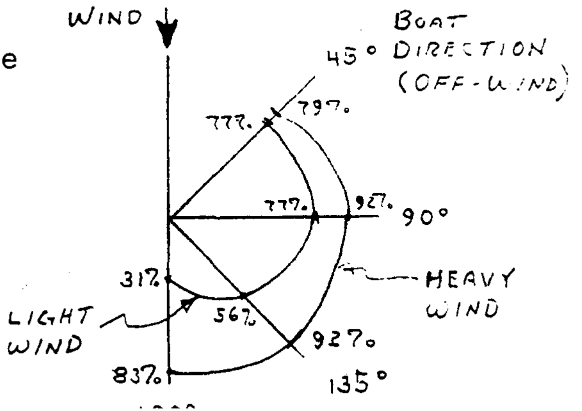
4. The wind speed varies with location, both in the north and east direction. The wind direction can be from any quadrant, but is the same over the entire area. Randomly, the wind speed and direction will shift. All shifts are at the end of a round, where a round is essentially one hour of sailing time. Hence, after all boats have moved with a given wind, the wind may shift. The wind velocity ranges from 2 to 26 knots.

5. In addition to wind shifts, there are random storms which will move the boats in varying amounts. The storms always come from the northwest.

6. Changing your boats direction costs you, in terms of distance traveled. It takes longer to turn in light winds than in heavy winds. So be careful when the wind is light.

7. As each boat crosses the finish line the time and position of crossing is printed. If the E-position is outside of the finish buoy location (+ 10 miles) it is so noted. As each boat finishes it is taken out of the race. The remaining boats continue to race until all have crossed the finish line.

8. The boat speed is dependent on the relative angle of the direction of the boat to the direction of the wind. It is also dependent on wind speed. (The response is base on that given in Scientific American, August 1966). Generally, the fastest boat speed is attained when going at right angles, or slightly more to the wind. It is assumed you have a spinnaker and use it properly at the right times.



Yot Race - (cont)

Program Listing - Banks 1 and 2

```
000  76 LBL    080  69 OP     160  77  GE    240  01  1     320  42 STO    400  05  5
001  34 √X     081  10  10    161  79  x̄     241  75  -     321  28  28    401  01  1
002  43 RCL    082  50 I×I    162  76 LBL    242  53  (     322  76 LBL    402  95  =
003  35  35    083  22 INV    163  77  GE    243  73 RC*    323  39 COS    403  42 STO
004  32 X:T    084  44 SUM    164  01  1     244  05  05    324  58 FIX    404  00  00
005  25 CLR    085  33  33    165  44 SUM    245  75  -     325  03  03    405  73 RC*
006  73 RC*    086  43 RCL    166  26  26    246  43 RCL    326  98 ADV    406  00  00
007  05  05    087  01  01    167  03  3     247  35  35    327  43 RCL    407  42 STO
008  77  GE    088  72 ST*    168  44 SUM    248  72 ST*    328  46  46    408  31  31
009  77  GE    089  06  06    169  04  04    249  05  05    329  69 OP     409  18 C'
010  86 STF    090  75  -     170  44 SUM    250  54  )     330  04  04    410  06  6
011  00  00    091  43 RCL    171  05  05    251  55  ÷     331  69 OP     411  00  0
012  65  ×     092  30  30    172  44 SUM    252  43 RCL    332  05  05    412  95  =
013  07  7     093  95  =     173  06  06    253  33  33    333  43 RCL    413  44 SUM
014  85  +     094  50 I×I    174  97 DSZ    254  55  ÷     334  32  32    414  29  29
015  73 RC*    095  85  +     175  02  02    255  43 RCL    335  42 STO    415  61 GTO
016  04  04    096  03  3     176  34 √X     256  01  01    336  02  02    416  75  -
017  65  ×     097  07  7     177  22 INV    257  38 SIN    337  07  7     417  76 LBL
018  01  1     098  95  =     178  87 IFF    258  95  =     338  42 STO    418  10 E'
019  08  8     099  42 STO    179  00  00    259  49 PRD    339  04  04    419  42 STO
020  75  -     100  00  00    180  30 TAN    260  33  33    340  08  8     420  59  59
021  43 RCL    101  73 RC*    181  18 C'     261  85  +     341  42 STO    421  02  2
022  29  29    102  00  00    182  01  1     262  43 RCL    342  05  05    422  42 STO
023  95  =     103  59 INT    183  00  0     263  03  03    343  76 LBL    423  32  32
024  38 SIN    104  55  ÷     184  95  =     264  95  =     344  38 SIN    424  76 LBL
025  65  ×     105  43 RCL    185  32 X:T    265  58 FIX    345  18 C'     425  30 TAN
026  01  1     106  35  35    186  01  1     266  02  02    346  43 RCL    426  07  7
027  02  2     107  85  +     187  77  GE    267  69 OP     347  28  28    427  44 SUM
028  85  +     108  73 RC*    188  65  ×     268  06  06    348  85  +     428  59  59
029  01  1     109  00  00    189  03  3     269  61 GTO    349  73 RC*    429  22 INV
030  04  4     110  22 INV    190  77  GE    270  35 1/X    350  05  05    430  86 STF
031  95  =     111  59 INT    191  55  ÷     271  76 LBL    351  95  =     431  01  01
032  42 STO    112  65  ×     192  76 LBL    272  79  x̄     352  50 I×I    432  22 INV
033  33  33    113  43 RCL    193  75  -     273  32 X:T    353  72 ST*    433  58 FIX
034  98 ADV    114  33  33    194  22 INV    274  43 RCL    354  05  05    434  98 ADV
035  58 FIX    115  95  =     195  86 STF    275  34  34    355  18 C'     435  03  3
036  00  00    116  49 PRD    196  00  00    276  69 OP     356  02  2     436  69 OP
037  43 RCL    117  33  33    197  69 OP     277  04  04    357  00  0     437  17  17
038  47  47    118  04  4     198  23  23    278  43 RCL    358  85  +     438  47 CMS
039  69 OP     119  05  5     199  01  1     279  25  25    359  43 RCL    439  06  6
040  04  04    120  49 PRD    200  42 STO    280  77  GE    360  27  27    440  69 OP
041  43 RCL    121  01  01    201  26  26    281  78 Σ+     361  95  =     441  17  17
042  26  26    122  43 RCL    202  07  7     282  85  +     362  74 SM*    442  43 RCL
043  69 OP     123  01  01    203  42 STO    283  93  .     363  04  04    443  45  45
044  06  06    124  38 SIN    204  04  04    284  00  0     364  73 RC*    444  69 OP
045  43 RCL    125  65  ×     205  08  8     285  01  1     365  04  04    445  04  04
046  31  31    126  43 RCL    206  42 STO    286  95  =     366  50 I×I    446  18 C'
047  69 OP     127  33  33    207  05  05    287  77  GE    367  55  ÷     447  93  .
048  04  04    128  95  =     208  09  9     288  77  GE    368  43 RCL    448  01  1
049  43 RCL    129  74 SM*    209  42 STO    289  76 LBL    369  36  36    449  85  +
050  33  33    130  05  05    210  06  06    290  78 Σ+     370  85  +     450  43 RCL
051  69 OP     131  73 RC*    211  43 RCL    291  69 OP     371  73 RC*    451  35  35
052  06  06    132  05  05    212  32  32    292  06  06    372  05  05    452  95  =
053  91 R/S    133  77  GE    213  42 STO    293  61 GTO    373  59 INT    453  58 FIX
054  76 LBL    134  45 Y^X    214  02  02    294  77  GE    374  95  =     454  03  03
055  11  A     135  76 LBL    215  61 GTO    295  76 LBL    375  99 PRT    455  69 OP
056  01  1     136  35 1/X    216  34 √X     296  65  ×     376  03  3     456  06  06
057  85  +     137  43 RCL    217  76 LBL    297  87 IFF    377  44 SUM    457  42 STO
058  76 LBL    138  01  01    218  18 C'     298  01  01    378  04  04    458  25  25
059  12  B     139  39 COS    219  07  7     299  55  ÷     379  44 SUM    459  04  4
060  01  1     140  65  ×     220  22 INV    300  01  1     380  05  05    460  00  0
061  85  +     141  43 RCL    221  23 LNX    301  05  5     381  97 DSZ    461  42 STO
062  76 LBL    142  33  33    222  65  ×     302  32 X:T    382  02  02    462  27  27
063  13  C     143  95  =     223  43 RCL    303  43 RCL    383  38 SIN    463  61 GTO
064  01  1     144  74 SM*    224  59  59    304  03  03    384  76 LBL    464  39 COS
065  85  +     145  04  04    225  95  =     305  77  GE    385  55  ÷     465  76 LBL
066  76 LBL    146  73 RC*    226  22 INV    306  55  ÷     386  98 ADV    466  16 A'
067  14  D     147  04  04    227  59 INT    307  98 ADV    387  43 RCL    467  32 X:T
068  01  1     148  50 I×I    228  42 STO    308  43 RCL    388  49  49    468  06  6
069  85  +     149  55  ÷     229  59  59    309  50  50    389  69 OP     469  22 INV
070  76 LBL    150  43 RCL    230  65  ×     310  69 OP     390  04  04    470  77  GE
071  15  E     151  36  36    231  92 RTN    311  04  04    391  69 OP     471  52 EE
072  00  0     152  85  +     232  76 LBL    312  69 OP     392  05  05    472  00  0
073  95  =     153  73 RC*    233  45 Y^X    313  05  05    393  18 C'     473  77  GE
074  42 STO    154  05  05    234  86 STF    314  00  0     394  08  8     474  52 EE
075  01  01    155  59 INT    235  01  01    315  42 STO    395  95  =     475  32 X:T
076  75  -     156  95  =     236  43 RCL    316  27  27    396  59 INT    476  42 STO
077  73 RC*    157  58 FIX    237  48  48    317  02  2     397  42 STO    477  32  32
078  06  06    158  03  03    238  69 OP     318  00  0     398  30  30    478  61 GTO
079  95  =     159  99 PRT    239  04  04    319  94 +/-    399  85  +     479  30 TAN
```

Yot Race - (cont)

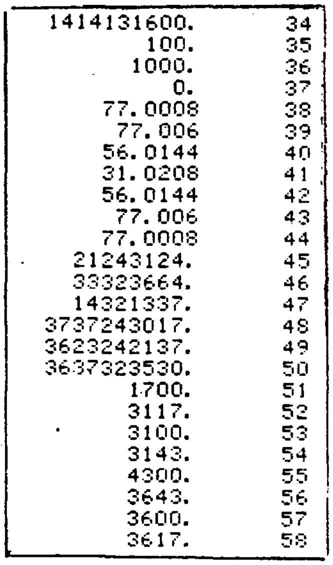
User Instructions:

1. Use the turn-on partitioning. Load in banks 1 and 2 from the listing on page 21, and bank 3 from the listing at the right.

2. Enter a positive number to seed the random number generator.
   - Press E' to start. Two boats will race, this is the default mode.
   - A number such as 100.043 with the annotation "FINI" will be printed to indicate the position of the buoy on the finish line.
   - The starting positions of the boats will be printed.
   - The wind speed and direction will be printed.

3. The first player presses A, B, C, D, or E for West, Northwest, North, Northeast, or east to set the direction for his boat. The new position for the first boat is printed, followed by the wind speed and direction for the next boat.

Bank 3



```
 1414131600.   34
        100.   35
       1000.   36
          0.   37
     77.0008   38
      77.006   39
     56.0144   40
     31.0208   41
     56.0144   42
      77.006   43
     77.0008   44
   21243124.   45
   33323664.   46
   14321337.   47
 3737243017.   48
 3623242137.   49
 3637323530.   50
       1700.   51
       3117.   52
       3100.   53
       3143.   54
       4300.   55
       3643.   56
       3600.   57
       3617.   58
```

4. Repeat step 3 for the remaining boats. The program will continue cycling through the boats in order until all boats pass the finish line.

5. As each boat pzsses the finish line its crossing time in hours and its position when crossing the line are printed. If the finishing position is outside the finishing gate, the closest buoy and "BAD" will be printed, and the boat is removed from the race.

Notes:

1. Wind shifts are indicated by printing "SHIFT".

2. Storms are indicated by printing "STORM". After a storm the new position of each boat is printed.

3. After all boats have crossed the finish line, a new race is automatically started. It is not necessary to press E' again.

4. To have other than a two boat race, enter a number between 1 and 6, and press A'. Do not press E' again.

5. Be sure to enter a different seed number each time you start the game, or else you will get the same sequence of races and race events.

6. In printing the boat location negative numbers are not indicated. The North value is always kept between 0 and 100. The east value can go higher than 100. This will print OK. When the east value goes below zero, the program remembers that it is negative, but it is printed as positive. For example, a true position of north = 50 and east = -12 is printed 50.012 .

7. Remember, the positions are printed to the nearest mile, but the program keeps fractional values as well. So it's a good idea to allow a one mile tolerance when aiming between the end buoys at the finish line.

Sample Game:

A sample two player game with a starting seed of 77777 appears on page 23.

Yot Race - (cont)

```
Press
      100.060      FINI

                   POS=
        0.043
        0.051

                   SHIFT

           1.      BOAT
          21.        NW
D     12.056

           2.      BOAT
          13.        NW
C      9.051

           1.      BOAT
          13.        NW
D     20.063

           2.      BOAT
           3.        NW
D     10.052

           1.      BOAT
          12.        NW
C     28.063

           2.      BOAT
           2.        NW
D     11.053

           1.      BOAT
           3.        NW
C     31.063

           2.      BOAT
           4.        NW
D     13.055

                   STORM

                   POS=
       28.078
        3.059

                   SHIFT

           1.      BOAT
          21.        NW
C     44.078

           2.      BOAT
          13.        NW
C     12.059

           1.      BOAT
           2.        NW
C     46.078

           2.      BOAT
          24.        NW
C     31.059

                   SHIFT

           1.      BOAT
           2.         N
B     46.077
```

```
Press
           2.      BOAT
          12.         N
B     38.053

                   SHIFT

           1.      BOAT
           4.         S
B     48.076

           2.      BOAT
          26.         S
C     58.053

           1.      BOAT
           7.         S
B     52.072

           2.      BOAT
           4.         S
D     60.054

                   SHIFT

           1.      BOAT
          21.        NW
C     68.072

           2.      BOAT
           4.        NW
D     62.056

           1.      BOAT
           2.        NW
C     69.072

           2.      BOAT
           3.        NW
D     64.058

           1.      BOAT
           2.        NW
C     71.072

           2.      BOAT
           9.        NW
D     69.063

           1.      BOAT
           2.        NW
C     73.072

           2.      BOAT
          26.        NW
C     88.063

                   SHIFT

           1.      BOAT
           3.         E
B     74.071

           2.      BOAT
          12.         E
C     99.063

                   SHIFT
```

```
Press
           1.      BOAT
          10.        SE
C     80.071

           2.      BOAT
           5.        SE          Boat 2
C      15.29       TIME          Finish
     100.063

           1.      BOAT
           3.        SE
C     82.071

                   SHIFT

           1.      BOAT
           4.         S
C     83.071

                   SHIFT

           1.      BOAT
           3.        SE
C     86.071

           1.      BOAT
           2.        SE
C     87.071

           1.      BOAT
           2.        SE
C     88.071

           1.      BOAT
           2.        SE
C     89.071

           1.      BOAT
           2.        SE
C     91.071

           1.      BOAT
           3.        SE
C     92.071

                   SHIFT

           1.      BOAT
           2.         W
C     94.071

           1.      BOAT
           3.         W
C     96.071

                   SHIFT

           1.      BOAT
           2.        NW
C     98.071

                   SHIFT

           1.      BOAT
           3.        NW          Boat 1
C      27.91       TIME          Finish
     100.071
     100.070       BAD
```

CALCULATING e TO MANY DIGITS - Patrik Johansson. This program for finding many digits of the base for natural logarithms appeared on page 4/5 of Volume 80-3 of Programbiten, the Swedish newsletter. The program is a straightforward mechanization of the formula

$$e = 1 + 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \ldots \quad \ldots \frac{1}{n!} \quad \ldots$$

The program published in Programbiten used 142 steps, no subroutines, no user-defined or common labels, and was started with a RST R/S sequence. That made it very easy to convert to fast mode using the Stf at the end of partition technique.

The TI-59 program delivers 480 digits, of which the last three are incorrect due to truncation effects. Normal mode requires nearly twelve hours. Fast mode requires just over six hours.

A second program can be run on both the TI-58 and the TI-59, and by disabling the fast mode sequence can be run on the TI-66. This program delivers 180 digits, where the last two are suspect. This program provides a good comparison of execution speeds:

TI-66	3 hours 50 minutes
TI-58C	2 hours 40 minutes
TI-58C Fast Mode	1 hour 29 minutes
TI-59	2 hours 1 minute
TI-59 Fast Mode	1 hour 1 minute

A third program is for the TI-66 only. It takes advantage of the Part command to obtain more digits than can be obtained with the TI-58C. The program runs in 2nd-Part-47 which yields 136 instructions and 47 data registers. The TI-66 program will yield 220 digits, of which the last two are suspect. The run time is about 5½ hours.

Operating Instructions:

1. For the TI-58C/59 programs, load the program and press A. See a flashing 1. Press 7 and then EE, and settle back and wait. The calculator stops with the first 10 digits in the display. Press R/S to recall additional groups of ten digits. A zero in the display indicates the end.

2. For the TI-66, enter the program and press RST R/S to start.

TI-59 Listing:

```
000  47 CMS    027  99  99    054  08  8     081  22 INV    108  55  ÷     135  00  0
001  01  1     028  43 RCL    055  44 SUM    082  74 SM*    109  09  9     136  92 RTN
002  52 EE     029  99  99    056  00  00    083  00  00    110  22 INV    137  76 LBL
003  09  9     030  55  ÷     057  32 X:T    084  32 X:T    111  28 LOG    138  11  A
004  42 STO    031  43 RCL    058  74 SM*    085  22 INV    112  52 EE     139  71 SBR
005  96  96    032  98  98    059  00  00    086  44 SUM    113  22 INV    140  01  01
006  42 STO    033  95  =     060  73 RC*    087  00  00    114  52 EE     141  47  47
007  48  48    034  59 INT    061  00  00    088  32 X:T    115  95  =     142  61 GTO
008  04  4     035  72 ST*    062  55  ÷     089  43 RCL    116  91 R/S    143  01  01
009  08  8     036  00  00    063  01  1     090  00  00    117  04  4     144  06  06
010  42 STO    037  65  ×     064  00  0     091  22 INV    118  08  8     145  00  0
011  00  00    038  43 RCL    065  22 INV    092  67  EQ    119  32 X:T    146  00  0
012  01  1     039  98  98    066  28 LOG    093  00  00    120  09  9     147  01  1
013  44 SUM    040  95  =     067  95  =     094  48  48    121  05  5     148  00  0
014  98  98    041  22 INV    068  59 INT    095  02  2     122  42 STO    149  69 OP
015  25 CLR    042  44 SUM    069  69 OP     096  04  4     123  00  00    150  17  17
016  42 STO    043  99  99    070  20  20    097  05  5     124  73 RC*    151  47 CMS
017  99  99    044  97 DSZ    071  74 SM*    098  32 X:T    125  00  00    152  60 DEG
018  01  1     045  00  00    072  00  00    099  43 RCL    126  91 R/S    153  22 INV
019  00  0     046  00  00    073  69 OP     100  98  98    127  69 OP     154  58 FIX
020  22 INV    047  18  18    074  30  30    101  22 INV    128  30  30    155  04  4
021  28 LOG    048  69 OP     075  65  ×     102  67  EQ    129  43 RCL    156  05  5
022  49 PRD    049  20  20    076  01  1     103  00  00    130  00  00    157  30 TAN
023  99  99    050  73 RC*    077  00  0     104  08  08    131  22 INV    158  33 X²
024  73 RC*    051  00  00    078  22 INV    105  92 RTN    132  67  EQ    159  86 STF
025  00  00    052  32 X:T    079  28 LOG    106  43 RCL    133  01  01
026  44 SUM    053  04  4     080  95  =     107  96  96    134  24  24
```

Calculating e to Many Digits - (cont)

TI-58C Listing: (Note: this will also run on the TI-59 and deliver fewer digits, but in much shorter time.)

```
000  47 CMS    027  39  39    054  08  8     081  22 INV    108  55  ÷     135  00  0
001  01  1     028  43 RCL    055  44 SUM    082  74 SM*    109  09  9     136  92 RTN
002  52 EE     029  39  39    056  00  00    083  00  00    110  22 INV    137  76 LBL
003  09  9     030  55  ÷     057  32 X:T    084  32 X:T    111  28 LOG    138  11  A
004  42 STO    031  43 RCL    058  74 SM*    085  22 INV    112  52 EE     139  71 SBR
005  36  36    032  38  38    059  00  00    086  44 SUM    113  22 INV    140  01  01
006  42 STO    033  95  =     060  73 RC*    087  00  00    114  52 EE     141  47  47
007  18  18    034  59 INT    061  00  00    088  32 X:T    115  95  =     142  61 GTO
008  01  1     035  72 ST*    062  55  ÷     089  43 RCL    116  91 R/S    143  01  01
009  08  8     036  00  00    063  01  1     090  00  00    117  01  1     144  06  06
010  42 STO    037  65  ×     064  00  0     091  22 INV    118  08  8     145  00  0
011  00  00    038  43 RCL    065  22 INV    092  67  EQ    119  32 X:T    146  00  0
012  01  1     039  38  38    066  28 LOG    093  00  00    120  03  3     147  01  1
013  44 SUM    040  95  =     067  95  =     094  48  48    121  05  5     148  00  0
014  38  38    041  22 INV    068  59 INT    095  01  1     122  42 STO    149  69 OP
015  25 CLR    042  44 SUM    069  69 OP     096  01  1     123  00  00    150  17  17
016  42 STO    043  39  39    070  20  20    097  01  1     124  73 RC*    151  47 CMS
017  39  39    044  97 DSZ    071  74 SM*    098  32 X:T    125  00  00    152  60 DEG
018  01  1     045  00  00    072  00  00    099  43 RCL    126  91 R/S    153  22 INV
019  00  0     046  00  00    073  69 OP     100  38  38    127  69 OP     154  58 FIX
020  22 INV    047  18  18    074  30  30    101  22 INV    128  30  30    155  04  4
021  28 LOG    048  69 OP     075  65  ×     102  67  EQ    129  43 RCL    156  05  5
022  49 PRD    049  20  20    076  01  1     103  00  00    130  00  00    157  30 TAN
023  39  39    050  73 RC*    077  00  0     104  08  08    131  22 INV    158  33 X²
024  73 RC*    051  00  00    078  22 INV    105  92 RTN    132  67  EQ    159  86 STF
025  00  00    052  32 X:T    079  28 LOG    106  43 RCL    133  01  01
026  44 SUM    053  01  1     080  95  =     107  36  36    134  24  24
```

TI-66 Listing: (Remember to use 2nd-Part-47)

```
000  47 CMS    023  46  46    046  00  00    069  69 OP     092  67  EQ    115  91 R/S
001  01  1     024  73 RC*    047  18  18    070  20  20    093  00  00    116  02  2
002  52 EE     025  00  00    048  69 OP     071  74 SM*    094  48  48    117  02  2
003  09  9     026  44 SUM    049  20  20    072  00  00    095  01  1     118  32 X:T
004  42 STO    027  46  46    050  73 RC*    073  69 OP     096  03  3     119  04  4
005  44  44    028  43 RCL    051  00  00    074  30  30    097  00  0     120  03  3
006  42 STO    029  46  46    052  32 X:T    075  65  ×     098  32 X:T    121  42 STO
007  22  22    030  55  ÷     053  02  2     076  01  1     099  43 RCL    122  00  00
008  02  2     031  43 RCL    054  02  2     077  00  0     100  45  45    123  73 RC*
009  02  2     032  45  45    055  44 SUM    078  22 INV    101  22 INV    124  00  00
010  42 STO    033  95  =     056  00  00    079  28 LOG    102  67  EQ    125  91 R/S
011  00  00    034  59 INT    057  32 X:T    080  95  =     103  00  00    126  69 OP
012  01  1     035  72 ST*    058  74 SM*    081  22 INV    104  08  08    127  30  30
013  44 SUM    036  00  00    059  00  00    082  74 SM*    105  43 RCL    128  43 RCL
014  45  45    037  65  ×     060  73 RC*    083  00  00    106 ~~44~~ 44  129  00  00
015  25 CLR    038  43 RCL    061  00  00    084  32 X:T    107  55  ÷     130  22 INV
016  42 STO    039  45  45    062  55  ÷     085  22 INV    108  09  9     131  67  EQ
017  46  46    040  95  =     063  01  1     086  44 SUM    109  22 INV    132  01  01
018  01  1     041  22 INV    064  00  0     087  00  00    110  28 LOG    133  23  23
019  00  0     042  44 SUM    065  22 INV    088  32 X:T    111  52 EE     134  00  0
020  22 INV    043  46  46    066  28 LOG    089  43 RCL    112  22 INV    135  92 RTN
021  28 LOG    044  97 DSZ    067  95  =     090  00  00    113  52 EE     136  00  0
022  49 PRD    045  00  00    068  59 INT    091  22 INV    114  95  =     137  00  0
```

MICROSTRIP DESIGN - E. T. Simon. Maurice Swinnen called my attention to this program which is discussed on pages 103-104 of the July 1984 issue of Microwaves & RF. The title is "Calculator Program Simplifies Microstrip Line Computations:. When you read the article you find the equations, and the instructions for running the program, but no program listing. To obtain the program you must send a stamped (40¢ US postage), self-addressed leagal-size envelope to Microstrip/Simon, c/o TinaMarie Pisa, Microwaves & RF, 10 Mulholland Drive, Hasbrouck Heights, NJ 07804.

FROM THE EDITOR:

This issue is later than I had expected. My recovery from back surgery has been slower than I had hoped for, but about as rapid as my doctor had predicted. For the future, I hope to distribute the fifth issue in mid-November, and the sixth issue early in 1985.

The "star" program in this issue is Barry Widman's combination of linear regression, data entry using ST-04, and high resolution graphics. The first page illustrates the versatility of the idea. The built-in ability to accept data pairs in any order, and to properly plot them in order, should be of use to those users who have been trying to plot multi-valued curves such as circles, foliums, and the like. A great program. Peter Poloczek's club in the Federal Republic of Germany has also been publishing some very good high resolution graphics. We will attempt to publish some translations in future issues. In the meantime, if you would like to borrow the original (in German) send a few stamps. As with other such offers I will send the master to you by first class mail, and expect a timely return by the same method.

Tests of precision continue to interest many members. Both the Bob Fruit test and the test from Scientific American turn out to depend on the precision of the Lnx function. Laurance Leeds and Myer Boland have been in the forefront in investigations in this area, and I plan to present some of their results in the next issue. It turns out that as the argument approaches one there is a significant loss in precision, consistent with the caution in Personal Programming. Larry and I have been developing routines which will provide at least a partial remedy for this problem. Curiously enough, it seems that the Lnx routine in the TI-66 is distinctly superior to that of the TI-59 for arguments very near one.

My HX-1000 Plotter/Printer for use with the CC-40 arrived. It performs the self-test beautifully, but does not communicate with my CC-40. The restrictions on activity while recovering from surgery have prevented me from resolving the problem. Meanwhile member Louis Krumpelman reports successful use of the RS-232C interface he purchased from Educalc.

V8N5P24/25 discussed the demise of PPX and suggested that we generate a listing of programs available from various members. The response was enthusiastic, and I now have a list of over 600 programs from PPX. However, several members expressed concern that providing copies for other members might result in problems with TI-s copyright protection. In mid-July I wrote to Consumer Relations requesting permission to set up the informal exchange that I discussed in V8N5. So far there has been no answer--maybe in time for the next issue.

TABLE OF CONTENTS

Magnetic card service and card reader cleaning strips continue to be available at the same rates quoted in earlier issues.

Palmer O Hanson, Jr