TI PPC NOTES

NEWSLETTER OF THE TI PERSONAL PROGRAMMABLE CALCULATOR CLUB

P.O. Box 1421, Largo, FL 34294

Volume 11, Number 4 Fourth Quarter 1986

With this issue we have completed the seventh year of our newsletter. At the end of the year we have 160 members in 33 states and 13 other countries. The big positive news items were the arrival of the TI-74 and TI-95. The big negative news appears on page 25 of this issue--after ten years TI has decided to stop repairing TI-59's and PC-100's. But there are still a lot of busy TI-59's out there. We will continue to support the TI-59, The TI-66 and the CC-40. I particularly invite members to send in information on the availability of used hardware for sale, supplies for sale, and any repair capability which might surface.

We will also continue and expand the coverage of the TI-74, the TI-95, the PC-324 and the associated peripherals. A critical issue is the establishment of portability of tapes made with the CI-7 Cassette Interface, and of programs recorded on the TI-695 RAM Cartridge. (You may remember that there have always been a few TI-59's which seemed to be incompatible with the rest.) To establish compatibility I am suggesting that members who have the CI-7 or TI-695 write to arrange exchange with other members.

A subscription form for the next "year" of four issues is attached. As in the past year I do not expect that I will be able to catch up so that all the issues can be mailed in 1987. Rather I expect that the final "1987" issue will not be mailed until April 1988. If you should decide not to continue our membership for 1987 I would appreciate a note to that effect.

TABLE OF CONTENTS

Magnetic card service will continue to be available for TI-59 programs in this issue, and for programs in the 1983, 1984 and 1985 issues. One dollar per card plus a stamped and self addressed envelope, please.

ERRATA:

Matrix Inversion Benchmark - During a review of the matrix inversion routine in the TI-74 I found that I had truncated the CC-40 results for the "Mathematics Written in Sand" benchmark problem in V10N1P5. The correct result, whether read from array2 in the CC-40 or array1 of the TI-74 is:

-4.99999 99999 61	-5.99999 99999 56	22.99999 99998 19	8.99999 99999 30
-10.99999 99999 17	-12.99999 99999 08	49.99999 99996 20	19.99999 99998 54
-6.99999 99999 47	-7.99999 99999 42	30.99999 99997 59	11.99999 99999 07
-0.99999 99999 925	-0.99999 99999 917	4.99999 99999 66	1.99999 99999 87

The changes reflect the availability of a fourteenth digit for those values which have two digits to the left of the decimal point.

USED HARDWARE FOR SALE

TI-59 Programmable Calculator with Personal Programming manual
PC-100C Printer
Master Library module
Real Estate/Investment module
Business Decisions module
Leisure Library module
Math/Utilities module
(the five modules above: with manuals)
TI-59 Workbook
12 Magnetic cards

The price is two hundred dollars ($200.00) for the complete package. Write to Thomas P. Nyenhuis, Milwaukee Metal Products Co., 8000 West Florist Avenue, Milwaukee WI 53218 or call (414)-463-2090.

MORE HARDWARE FOR SALE

TI-58C Programmable Calculator (used)
PC-100C Printer (new)
Master Library module
Surveying module
AC adapter/charger for the TI-58C (new)
1 spare BP-1A Battery Pack
3 rolls of paper for the PC-100C.

The manuals are in excellent condition. The prices are $125 for the PC-100C and $75.00 for the calculator, for a total of $200.00 . The owner would like to dispose of all items as a package. Write to Thomas R. Edick, 8344 Sandra Avenue, Clay, NY 13041 or call (315)-699-4053.

TI-95 AVAILABILITY - Elek-Tek is now listing TI-95 equipment:

TI-95 Procalc	$139.00
Mathematics Cartridge	33.00
Statistics Cartridge	33.00
PC-324 Printer (Also used with the TI-74)	69.00
CI-7 Cassette Interface (Also used with the TI-74)	22.00
TI-8K Memory Cartridge (Also used with the TI-74)	34.00

Shipping charges are $4.00 for the first item and $1.00 per additional item. Write to Elek-Tek, 6557 N. Lincoln Avenue, Chicago IL 60645, or call 800-621-1269 toll free. You can use Visa or Mastercard.

MORE ON THE ARITHMETIC IN THE TI-95 - V11N3P17 reported that the "Itsy-bit of Paranoia" test had indicated that the TI-95 had a "guard digit" while the TI-59 and TI-66 did not. The idea of a "guard digit" in Paranoia sense does not include the extra three digits (11th through 13th) used in the data registers of the TI-59 and TI-66 to protect the accuracy of the ten digits in the display. Rather, a guard digit is defined as an extra digit beyond the limits of the data registers, which protects the accuracy of all the digits in the data registers. In the case of 13 digit data registers as in the TI-59, TI-66 and TI-95 an additional (14th) digit is required. For the HP-11C which uses ten digit data registers, an eleventh digit is required to provide the guard digit function. One effect of a guard digit in this sense is that arithmetic results are rounded in the data registers, not truncated as with the TI-59 and TI-66. Peter Messer called to suggest that the truncation in the TI-59 was a key to several of the multi-precision routines, and that a round from a guard digit may cause errors in those routines. It appeared appropriate to examine the arithmetic of the TI-95 more carefully.

The discussion of numeric accuracy on page A-8 of the TI-95 User's Guide notes that the command sequence 1 / 3 x 3 - 1 = yields the answer -1.-13 , not the zero we might have expected. The same sequence on the TI-59 and TI-66 yields the answer -1.-12 . Equivalent sequences on other calculators yield -1e-10 on the HP-11C, -1e-14 on the Radio Shack Model 100 , and zero on the Casio fx-7000G. For the TI-95, HP-11C and Model 100 the error is 1 in the least significant digit, while on the TI-59 and TI-66 the error is 1 in the digit immediately above the least significant digit. This example illustrates preservation of accuracy, but does not address the rounding effect.

Consider the sequence 2 / 3 x 3 - 2 = . The TI-95, the HP-11C, the Model 100 and the fx-7000G yield zero. The TI-59 and TI-66 yield -1.-12 . For a similar sequence 1 / 9 x 6 x 3 - 2 = , the TI-95, the HP-11C, the Model 100 and the fx-7000G still yield zero. The TI-59 and the TI-66 still yield -1.-12 . One interesting thing about those sequences is that the contents of display register just before the multiplication by three are different for the two cases for most of the devices:

Device	2 / 3 =	1 / 9 x 6 =
TI-95	.6666666666667	.6666666666666
HP-11C	.6666666667	.6666666666
Model 100	.66666666666667	.66666666666666
fx-7000G	.6666666666666	.6666666666666
TI-59/TI-66	.6666666666666	.6666666666666

After the multiplication by three the devices other than the TI-59 and TI-66 yield the same end result for either intermediate result, a clear indication of the rounding routine. If we manufacture a number with twelve sixes and a seven on the TI-59 or TI-66, say by the sequence 2 / 3 + 1 EE 13 +/- = , then the subsequent multiplication by three and subtraction of two will yield zero in the display register. This is consistent with the truncation which occurs in those devices.

As Peter Messer suggests, we will have to consider the impact of the different arithmetic when converting TI-59 programs for use on the TI-95. There are other arithmetic differences between the TI-59 and TI-95. One is described in the conversion of a program for computing many digits of e on the folowing page.

MANY DIGITS OF e ON THE TI-95

V11N1P17 presented Hewlett Ladd's program for finding many digits of e using a TI-59 and PC-100. The upper listing at the right is the result of a direct conversion of that program for use on the TI-95, with the absolute addressing in the TI-59 program changed to label addressing.

```
0000 BRK GTL AE CLR
0005 LBL AA 1 EE 9 INV
0012 EE ST+ IND 098 INV
0018 DSZ 099 GTL AC
0024 RCL 098 EXC 000
0030 LBL AB ( CE *1 EE 1
0039 0+ RCL IND 000 -(
0047 CE / RCL 099 ) INT
0054 STO IND 000 *
0059 RCL 099 ) DSZ 000
0066 GTL AB GTL AA
0072 LBL AC ST/ IND 098
0079 RCL 098 STO 000 ADV
0086 LBL AD RCL IND 000
0093 PRT DSZ 000 GTL AD
0100 0 GTO 0000
0104 LBL AE +9=/10= INT
0115 x~t NOP CMS x~t
0119 STO 098 19+5.4*
0129 RCL 098 -.01*
0137 RCL 098 x^2 = INT
0143 STO 099 GTL AA
```

A change in the program sequence at steps 0030-0041 was required due to differences in the way in which the TI-59 and TI-95 handle the EE function. The equivalent sequence at steps 018-024 of the TI-59 program

(INV EE EE 10 + RC* 00 - ...

was explained on V11N1P16. It takes a number less than a thousand which is in EE format and multiplies it by 1e10. The process relies on a characteristic of the EE command that can be most easily explained by an illustration. Press 456 EE = to place the number 456 in EE format in the display register of either calculator.

	TI-59 Display	TI-95 Display
Starting display	4.56 02	4.56 02
Press (INV EE	456	456
Press EE	456 00	4.56 02
Press 1 0	456 10	4.56 10
Press +	4.56 12	4.56 10 +

We see that the sequence on the TI-59 has had the desired result of multiplying the starting value of 456 by 1e10 . On the TI-95 the starting value has only been multiplied by 1e8 because the mantissa was converted to the EE format when EE was pressed, and the new exponent of 10 replaced an existing exponent 02, not an existing exponent 00 as with the TI-59. The solution for the TI-95 conversion is to use a straightforward multiplication by 1e10.

```
0000 BRK GTL AE
0004 LBL AA 1 EE 9 INV
0011 EE ST+ IND 098 INV
0017 DSZ 099 GTL AC
0023 RCL 098 EXC A
0028 LBL AB (*1 EE 10+
0038 RCL IND A -(/
0044 RCL 099 ) INT
0049 STO IND A * RCL 099
0056 ) DSZ A GTL AB
0062 GTL AA
0065 LBL AC ST/ IND 098
0072 RCL 098 STO A ADV
0078 LBL AD RCL IND A
0084 PRT INV TF 74 BRK
0089 DSZ A GTL AD 0
0095 GTO 0000
0098 LBL AE +9=/10= INT
0109 x~t NOP CMS x~t
0113 STO 098 19+5.4*
0123 RCL 098 -.01*
0131 RCL 098 x^2 = INT
0137 STO 099 GTL AA
```

```
0004 LBL AA
0028 LBL AB
0065 LBL AC
0078 LBL AD
0098 LBL AE
```

```
2.718281828
4590452353.
6028747135.
```

The lower listing uses letter addressing for data register 000, deletes the CE immediately after the parenthesis in line 028 to illustrate this capability of the TI-95, and inserts an INV TF 74 BRK sequence after the PRT command at step 0084 to illustrate an more efficient method of printer control than the Y/N technique used in the statistics program on V11N3P23.

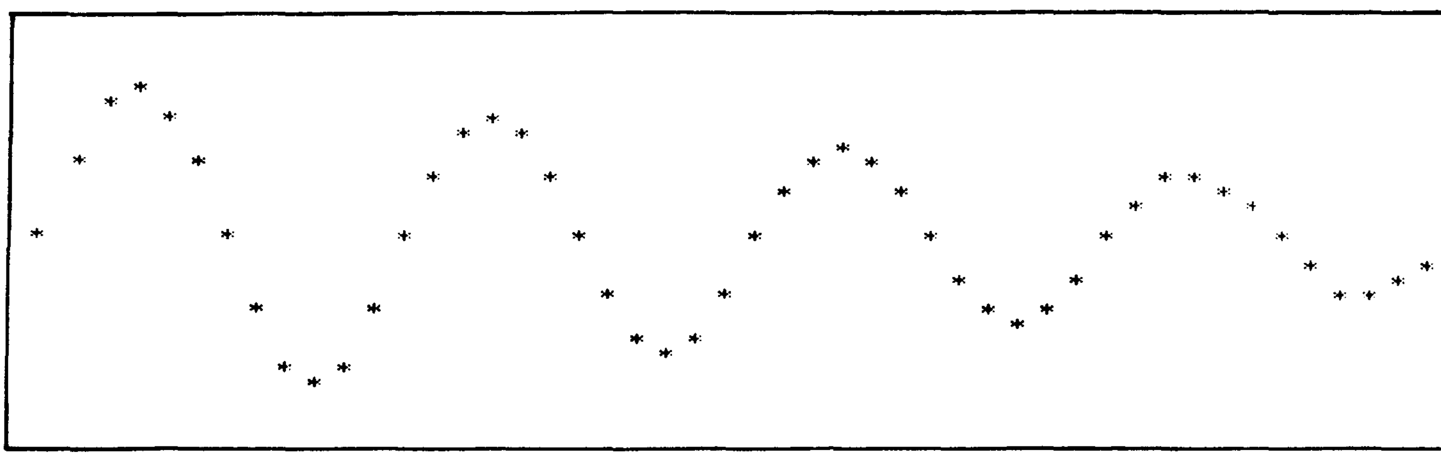
Many Digits of e on the TI-95 - (cont)

To use the program simply enter the number of digits required, press F1 (GO), and wait for the printout if the PC-324 is attached, or see the first block of the solution in the display. Press F1 (GO) to see additional blocks. The program yields 30 digits of e in 16 seconds; the program from V11N1P17 for the TI-59 yielded 30 digits in 80 seconds when operating in fast mode.

OP 07 EQUIVALENT ON THE TI-95 - Robert Stucker.

```
0000 GTL AA
0003 LBL 07 STO A
0008 COL IND A '*' PRT
0013 RTN
0014 LBL AA 0 STO B
0020 LBL AB 11* RCL B
0028 SIN *(2000- RCL B )
0039 /2000+12= SBL 07 30
0053 ST+ B 1440 IF> B
0061 GTL AB HLT
```

One of the "From the Editor" comments on V11N3P24 noted that there didn't seem to be any built-in plot capability for the TI-95 similar to the OP 07 command in the TI-59. An equivalent capability can be easily mechanized as an eleven step subroutine. See the LBL 07 routine in the program at the right. With the integer portion of a point to be plotted in the range from 1 to 24 a SBL 07 call prints an asterisk in the corresponding column. A plot from the demonstration program at the right appears below. This program also illustrates the capability to use labels which are comprised only of numbers.



COPYING THE DISPLAY TO THE T REGISTER OF THE TI-95 - P. Hanson. The TI-95 can raise a negative argument to an integer power. This permits copying the display register to the t register with the sequence y x⇄t 1 = . Negative arguments with non-integer powers produce curious results. See page 18.

MORE TI-59 VS TI-95 - Robert Prins reports that there are more differences between the responses of the TI-59 and TI-95 which will affect program conversions. On the TI-95:

* INV Σ+ doesn't decrement the t register.
* The Signum (SGN) function (OP 10 in the TI-59) returns a 1 for a 0 input.

MORE ON ODDS AGAINST - L. Leeds. V10N4P14-20 discussed the "odds against" problem and challenged members to devise simple and speedy solutions. The following limited range solution mechanizes a version of the basic formula and yields accurate answers.

For n statistically independent trials where each trial has a probability of success p and a probability of failure (1 - p) then the probability of at least x successes in n trials is:

$$P = \sum_{x}^{n} \frac{n!}{x!(n-x)!} p^{x}(1-p)^{n-x}$$

and the odds against success are (1-P)/P.

The first equation can be transformed into

$$P = p^{n}(1 + \frac{n}{1}C(1 + \frac{n-1}{2}C(1 + \frac{n-2}{3}C(1 + \ldots \frac{x+1}{n-x}C(1$$

where C = (1-p)/p.

Program Listing:

000	91	R/S	025	43	RCL	050	03	03	075	95	=	100	75	-			
001	25	CLR	026	02	02	051	95	=	076	42	STO	101	43	RCL			
002	61	GTO	027	85	+	052	42	STO	077	07	07	102	09	09			0
003	00	00	028	01	1	053	06	06	078	69	OP	103	54	)		[illegible]	0
004	25	25	029	95	=	054	01	1	079	24	24	104	55	÷		00	0
005	76	LBL	030	42	STO	055	42	STO	080	69	OP	105	43	RCL	[illegible]	00	0
006	11	A	031	04	04	056	07	07	081	35	35	106	09	09	221	00	0
007	42	STO	032	43	RCL	057	43	RCL	082	97	DSZ	107	95	=	222	00	0
008	01	01	033	01	01	058	06	06	083	00	00	108	42	STO	223	00	0
009	91	R/S	034	75	-	059	65	×	084	00	00	109	10	10	224	00	0
010	76	LBL	035	43	RCL	060	43	RCL	085	57	57	110	66	PAU	225	00	0
011	12	B	036	02	02	061	07	07	086	43	RCL	111	81	RST	226	00	0
012	42	STO	037	95	=	062	65	×	087	03	03	112	76	LBL	227	00	0
013	02	02	038	42	STO	063	43	RCL	088	45	Y^x	113	15	E	228	00	0
014	91	R/S	039	05	05	064	04	04	089	43	RCL	114	43	RCL	229	00	0
015	76	LBL	040	42	STO	065	55	÷	090	01	01	115	09	09	230	09	9
016	13	C	041	00	00	066	43	RCL	091	95	=	116	91	R/S	231	69	OP
017	42	STO	042	53	(	067	05	05	092	65	×	117	00	0	232	17	17
018	03	03	043	01	1	068	95	=	093	43	RCL	118	00	0	233	04	4
019	91	R/S	044	75	-	069	42	STO	094	07	07	119	00	0	234	85	+
020	76	LBL	045	43	RCL	070	08	08	095	95	=	120	00		235	52	EE
021	14	D	046	03	03	071	01	1	096	42	STO	121	[illegible]		236	01	1
022	61	GTO	047	54	)	072	85	+	097	09	09	122			237	02	2
023	02	02	048	55	÷	073	43	RCL	098	53	(	[illegible]			238	95	=
024	30	30	049	43	RCL	074	08	08	099	01	1				239	86	STF

User Instructions:

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

2. Enter the number of successes (x) and press B.

3. Enter the probability of success (p) and press C.

4. To solve press D and see a flashing "4. 12" in the display. Press 7 and then EE. The calculator will stop with the odds against in the display.

5. Press E to see the probability (P).

More on Odds Against - (cont)

The listing at the right is the TI-95 equivalent of the TI-59 program. Prompting for input is provided through the windows and the associated function keys.

```
0000 DFN F1: n @ A
0007 DFN F2: x @ B
0014 DFN F3: p @ C
0021 DFN F4:Ans@ D
0028 DFN F5: P @ E HLT
0036 LBL  A STO 001 HLT
0043 LBL  B STO 002 HLT
0050 LBL  C STO 003 HLT
0057 LBL  D RCL 002 +1=
0066 STO 004 RCL 001 -
0073 RCL 002 = STO 005
0080 STO 000 (1- RCL 003
0089 )/ RCL 003 =
0095 STO 006 1 STO 007
0102 LBL  F RCL 006 *
0109 RCL 007 * RCL 004 /
0117 RCL 005 = STO 008 1
0125 + RCL 008 = STO 007
0133 INC 004 INV INC 005
0140 DSZ 000 GTL  F
0146 RCL 003 y^x RCL 001
0153 =* RCL 007 =
0159 STO 009 (1- RCL 009
0168 )/ RCL 009 =
0174 STO 010 HLT
0178 LBL  E RCL 009 HLT
```

Both programs provide accurate answers at least through the case of 120 trials, 84 successes, and a probability of success of 1/2; i.e., for Z = 4.8 .

Execution speeds are significantly faster than with the ST-20 routine in the Applied Statistics module of the TI-59. For a probability of success for each trial of 1/2 and various cases of trials and successes the execution times in seconds are:

Case	ST-20	TI-59	TI-95
7/10	6	4	1
70/100	63	25	4
140/200	122	45	9

HOW CAN WE DISABLE THE PRINTER WITH THE TI-95? - Peter Messer and Hewlett Ladd have both noted that it would be nice to be able to disable the PC-324 printer while it is connected to the TI-95; that is, the equivalent of what we could do with the TI-59 mounted on the PC-100C by simply turning off the printer switch. This would allow us to use the AC-9201 to save calculator batteries without running through the printer paper. So far we haven't been able to come up with a way to disable the PC-324 from the keyboard or from a program. Any ideas out there?

AVAILABILITY OF THE MATH & STATS CARTRIDGES FOR THE TI-95 - P. Hanson

In early December 1986 I ordered a TI-95, the Cassette Interface, and the Mathematics Cartridge from Educalc. The calculator and interface arrived in a short time. The Mathematics Cartridge had not arrived as of April 15. Educalc says they expect to have the cartridges within the next week. Elek-Tek also indicates that the cartridges are not currently in stock.

PAPER FOR THE PC-324 - A number of members have asked if anyone has found an alternate supply of paper for the PC-324. The suggested retail price for three small rolls is about six dollars. The Elek-Tek price for a package is $4.50. The EduCalc price is $4.95. A roll of PC-324 paper is about one inch in diameter with a 3/16 inch hollow core. For a 2 1/4 inch width that amounts to about 1.7 cubic inches of paper. By comparison, the rolls for the PC-100 were about three inches in diameter (although all the packages I have found say it is 3 1/8 inch) with a one inch core. For the 2 1/2 inch paper width that amounts to about 15.7 cubic inches of paper. The suggested retail price for the PC-100 paper was $10.00 and the Elek-Tek price is $6.50. On a price per volume basis the PC-324 paper is over six times as expensive as the PC-100 paper. Interesting!

SOLUTION OF LINEAR EQUATIONS ON THE TI-95

We expect that the Mathematics cartridge will include a program for solution of linear equations; however, the cartridge is not yet available. The linear equations program at the right is a direct translation of a TI-59 program from V9N1P17. That program was a Robert Prins modification of a program by Henrik Ohlsson which appeared in the Swedish newsletter Programbiten. Pertinent changes introduced as part of the translation are:

* The absolute addressing required by fast mode on the TI-59 was changed to label addressing.
* t register comparisons are not available with the TI-95. The sequence 0 IF EQ 005 GTL AD at program steps 072-078 are the equivalent of an EQ in the TI-59 version.
* Program steps 003-024 combine the value of a matrix element with the associated index for display or printout. The CHR IND command is used to convert the numerical values to alphanumeric format.
* The TF 74 PRT at steps 025-028 skips the print command if the printer is not connected. This substantially speeds up the response when a printer is not connected.

```
0000 GTL BB
0003 LBL PR INC 009
0009 CHR IND 004 `,`
0014 CHR IND 009 BRK
0019 RCA = COL 20 MRG =
0025 TF 74 PRT RTN
0029 LBL A1 INV INC 007
0036 10 STO 003 STO 006
0044 LBL AA RCL 007
0050 STO 002 48 STO 009
0058 RCL 004 -48=
0065 STO 005 INC 004 ADV
0072 0 IF= 005 GTL AD
0079 LBL AB SBL PR +/-
0086 LBL AC * x~t
0091 RCL IND 006 )
0096 ST+ IND 003 INC 003
0103 INC 006 x~t DSZ 002
0110 GTL AC RCL 007
0116 STO 002 ST+ 003
0122 DSZ 005 GTL AB
0128 LBL AD SBL PR
0134 STO 005 RCL IND 003
0141 ST+ 005 INC 002
0147 LBL AE SBL PR
0153 STO IND 003 INC 003
0160 0 EXC IND 003 INV
0166 INC 003 ST+ IND 003
0173 RCL 005 ST/ IND 003
0180 INC 003 DSZ 002
0186 GTL AE RCL 004 -48=
0196 STO 005 INV DSZ 005
0203 GTL AI 9 STO 003
0210 RCL 001 STO 008
0216 LBL AF INC 003 INV
0223 INC 002 0
0227 EXC IND 003 x~t
0232 LBL AG INC 003 0
0239 EXC IND 003 -
0244 RCL 002 ST+ 003 x~t
0251 * x~t RCL IND 006 )
0258 STO IND 003 RCL 002
0265 ST- 003 INC 006
0271 DSZ 008 GTL AG
0277 RCL 001 ST+ 006
0283 STO 008 DSZ 005
0289 GTL AF RCL 002
0295 ST+ 003 INC 003
0301 LBL AH 0
0305 EXC IND 006 INC 006
0312 STO IND 003 INC 003
0319 DSZ 008 GTL AH
0325 LBL AI INC 007 10
0333 STO 006 DSZ 001
0339 GTL AA 48 ST- 004
0347 ADV
0348 LBL AJ RCL IND 006
0355 PRT INV TF 74 BRK
0360 INC 006 DSZ 004
0366 GTL AJ 0 HLT
0371 LBL BB `Order?` BRK
0381 +/- x~t CMS x~t
0385 STO 001 STO 007 48
0393 STO 004 GTL A1
```

User Instructions:

1. Start the program and see "Order?" in the Display and "GO" in the window above F1. Enter the order of the solution (n) and press F1.
2. The prompt "1,1" will appear in the display indicating the program is ready to accept matrix element A and "GO" will appear in the window above F1.
3. Enter the first element of the matrix and press F1. If the PC-324 is connected the element will be printed with annotation. The calculator will run for a short period of time and stop with "1,2" in the display indicating the program is ready to accept matrix element A . Continue entering the matrix elements by row in response to the prompts. When the column portion of the prompt becomes "i,n+1" enter vector element B and press F1. The calculator will run for a short period of time and stop with the prompt for the first matrix element in the next row in the display. Continue entry of the matrix and vector elements in response to the prompts.

Linear Equations on the TI-95 - (cont)

4. When the last vector element has been entered the calculator will run for a short period of time. If the PC-324 is connected the solution will be printed and the calculator will stop with a zero in the display. If the printer is not used the calculator will stop with the first element of the solution in the display. Press F1 to display the remaining elements.

A sample printout for an order 4 problem appears at the right. The problem is the sub-Hilbert matrix with all one's for the vector. In V8N6P18 Geoge Thomson proposed the 7 x 7 sub-Hilbert with all ones on the right hand side as a benchmark test. The following table compares the solution from this program with the solution from the original TI-59 program, and with the solution obtained from the HP-15.

```
1,1            0.5
1,2     .3333333333
1,3           0.25
1,4            0.2
1,5             1.

2,1     .3333333333
2,2           0.25
2,3            0.2
2,4     .1666666667
2,5             1.

3,1           0.25
3,2            0.2
3,3     .1666666667
3,4     .1428571429
3,5             1.

4,1            0.2
4,2     .1666666667
4,3     .1428571429
4,4          0.125
4,5             1.

     -19.99999998
      179.9999998
     -419.9999997
      279.9999998
```

Ohlsson/Prins

Exact	TI-59	TI-95	HP-15	HP-41
56	55.9233	56.0055	55.8025	56.6667
-1512	-1510.2276	-1512.1289	-1507.2559	-1527.3832
12600	12587.0911	12600.9496	12564.7148	12712.2414
-46200	-46157.9673	-46203.1170	-46083.6321	-46566.4960
83160	83091.9632	83165.0774	82970.0537	83755.0102
-72072	-72018.4333	-72076.0179	-71921.5786	-72541.8140
24024	24007.6425	24025.2321	23977.8723	24167.8491
	1.37E-3	9.81E-5	3.53E-3	1.19E-2

where the HP-41 solution was provided by Gene Friel using the Math-Pac Application module. The number at the bottom of each column is the largest relative error, (answer - exact)/exact. By that measure the TI-95 solution is 14 times better than the solution from the equivalent program in the TI-59, and is slightly better than any of the other standard TI-59 solutions reported in V8N6P19; however, it is nearly 7 times worse than the relative error for the TI-59 solutions modified with double divides to compensate for the non-commutative multiply (V9N2P18 & V9N6P24). The HP-15 solution is 3.4 times better than the HP-41 solution.

In V9N2P18 James Walters proposed that another method for evaluating a linear equations solution is to multiply the solution vector by the original matrix. The result should be the original vector, in our test problem, all ones. The results for some solutions are on page 26.

QUIRKS IN THE QUICKDISK FOR THE CC-40 - Louis Krumpelman. Earlier issues have noted the availability of a disk drive for use with the CC-40. There are indications that the Disk drive may also be useable with the TI-74. The following page presents the experience of one user of the device.

Quirks in the QuickDisk for the CC40

It was a great leap forward when the QuickDisk became available for the CC40 because it solves the off-line storage problem. Unfortunately, the QuickDisk has certain quirks which make it difficult to use.

1. Buffer is 16K. Any record which is read from the QuickDisk or written to the QuickDisk is read out of a single buffer in a one-write situation. That buffer has an upper limit of 16.3 Kilobytes. In Data applications, that is small. Whenever you have a Data File that is greater than 16 Kilobytes, you have to break it into several different files and access them individually. This is a problem because of the One-File-Open Rule.

2. One-File-Open Rule. Most disk operating systems will let you open at least 3, up to 8, and some I have seen as many as 32, disk files simultaneously. The QuickDisk lets you open 1 (one) disk file. In order to read another file, you must close the first file, and re-open the second file. This follows because it reads the entire file into the buffer memory, and since the buffer memory is only 16 Kilobytes, it has to clear it to free it for the next record. (problem in Data applications.)

3. Opening Modes. When a file is opened in the Input mode, a copy is not re-written to the diskette when it is closed. This is a very handy option when several files are to be scanned. However, when a file is opened in Output or Append or Update mode, it is essentially written sequentially and it will be rewritten into fresh space. The most practical of course, is the Update, and when you open in Update, the entire file is read from the disk into the buffer, changes are made while it is in the buffer, and the entire file is rewritten to the diskette in fresh space.

4. Pseudo-magnetic ROM. Clearing the disk resets it to all clear. As files are written, they are written sequentially, into unused space. When a file is opened in Update or Append or Output mode, it is read into the buffer. Even if no change is made, closing will make the QuickDisk write the file again in unused space, letting the original one stand also. So we have two copies of the same file on the diskette. The operating software of the QuickDisk will sort thru these and only read in the second one when it is re-opened. And again, if it is opened in Update, Append or Output, it will write a third copy of the file when it is closed.

5. Record Separators. In sequential mode, the QuickDisk seems to insert its own separators, although this is not a constant finding. However, in Random mode, the QuickDisk does not insert its own separators, such as returns or commas, and you have to insert these. If you do not insert these, you will get an error when you try to read the file. (It cannot find the end and continues to read.)

6. Creation of a Record. This QuickDisk basically thinks and acts like a sequential output device, that can provide, in some cases, a limited amount of random access. If a file is opened in the Random mode, and records 1, 2, 3, and 4 are written, and then you choose to write in record 10, you will get an error, because you have to write the records sequentially, even though you can access already written records randomly, you cannot write records in a portions of the file which has not been initialized in a sequential step-wise fashion. Therefore, this is a Pseudorandom Access.

7. Record Zero. When a file is opened, whether it be sequential or random modes, the index pointer in a QuickDisk is at the Zero Record position. You can write into this record, and in sequential files, this is perfectly usable. However, in random files, the record zero cannot be reaccessed. If the file is initially written into Record Space 0, 1, and 2, and then you move the Index Pointer with a Restore Record Command to 0, you would think it would point the index back to record 0, but it doesn't. It increments and points it to 3. It interprets a restore 0 as an increment to the next available record. Therefore the machine thinks like a sequential, although limited random access is available in an already created file.

8. Closing Error. If a CLOSE #N is executed, but there isn't enough room on the disk to save the file, an error message is returned. The QuickDisk does not save the file, and considers the file to stay open. However, the FILE BLOCK in the CC40 is destroyed, and the CC40 considers the file closed. This is unfortunate because future disk requests are blocked because it thinks it has a file open. When this occurs, the solution is to unplug the QuickDisk, wait a few seconds and plug it in, or to send a CALL 10 (8,255).

With these problems in mind, the QuickDisk is a very practical storage device for small files, and these files can be concatinated thru software to use the entire disk.

MATS IS DIFFERENT IN THE MATH MODULES OF THE CC-40 AND THE TI-74

V11N2P19 reported that the program package of the Mathematics module for TI-74 was very similar to that of the Mathematics module for the CC-40. Two discrepancies noted with the CC-40 Mathematics module had been corrected. I had determined that the manuals were very similar, and that the calls to the subprograms were the same.

V8N5P14-17 presented a detailed look at the MATS routines in the CC-40 Mathematics module. A demonstration least squares polynomial curve fit was included. When I tried to convert programs which used the MATS subprogram from the CC-40 to the TI-74 I encountered difficulty. Investigations revealed that although the calls are the same, and the numerical results are the same, there are important differences in the way the subprograms are mechanized. There is even an error in the definition of the test output which tells if the matrix is singular. The differences I have identified so far are:

1. The solution for linear equations appeared in the first column of array1 in the CC-40. It appears in the first row of array1 in the TI-74. Therefore, the conversion of the linear equations program on V8N5P14 for the TI-74 requires that line 420 be changed from

```
PRINT X$;A(I,1):PAUSE    to   PRINT X$;A(1,I):PAUSE
```

Similarly, the subscripts for A must be reversed in lines 320 and 380 of the curve fit program on V8N5P17.

2. The last argument of the MATS call is the "test" value. The definition of "test" in the manuals for the CC-40 and TI-74 is that it passes in a dummy variable, and returns a zero if the matrix is singular. In the CC-40 the MATS program would leave the value of "test" as it was if the matrix was not singular, and would change the value of "test" to zero if the matrix was singular. Thus, to differentiate between the two it was necessary to set the value of test to a non-zero value prior to calling MATS. That is the reason for the R = PI at line 110 of the linear equations program on V8N5P14, and for the R = 1 in line 150 of the polynomial curve fit program on V8N5P17, where R is the "test" variable name used in those programs. Any old non-zero value will do.

In the TI-74 mechanization of MATS the value of "test" is set to -1 if the matrix is singular, and is set to zero if the matrix is not singular. There is no need to pre-load a value for "test" prior to the call of MATS. The definition of "test" on page A-11 of the manual is incorrect.

3. The CC-40 mechanization of the matrix inversion routine (option 4) returned the inverse of an input matrix to array1 but with some columns interchanged, and the inverse of the input matrix with the columns in proper order to array2. In the TI-74 the inverse of the input matrix is returned to array1 with the columns in proper order. The contents of array2 are not disturbed.

4. V8N5P14 noted that for some reason the dimension statement must provide array1 with one additional column if a "Bad Subscript" error is to be avoided when running the linear equations solution on the CC-40. This idiosyncrasy does not appear in the TI-74.

LEAST SQUARES REGRESSION WITH USER DEFINED FUNCTIONS - P. Hanson

Discussions of least squares regression, at least in TI PC Notes, have typically been limited to the use of polynomials. Examples include Thomas Wysmuller's article on polynomial regression in the November/ December issue of PPX Exchange, Gene Friel's polynomial regression with variance (V9N2P20), and the least squares polynomial curve fit for the CC-40 from V8N5P17. There are many applications where the appropriate functions are not polynomials. One example is the solution for sin and (1- cos) functions for inertial navigation error analysis which appeared in V10N4P7. In the past I have also found it necessary to regress using functions such as $\sqrt{x}$, $|x|$, and the like, and have typpically written special programs for each application. To minimize such effort in the future I decided to structure a program such that a subroutine can be changed to define the regression functions, but the remainder of the program is independent of the regression functions. The resulting TI-74 program which follows requires that the Mathematics software module be installed. The program provides the necessary prompts so that separate user instructions are not required.

```
100 DIM A(8,8),B(8),C(8,
8),F(8),X(50),Y(50)
110 CALL UP("Least Squar
es Fit",PN)
120 PRINT "Are the funct
ions correct?":PAUSE 2
130 INPUT "Number of Dat
a Pairs? ";K
140 CALL AU("X","Y",X(),
Y(),1,K,PN)
150 INPUT "Order of the
solution? ";N
160 FOR I=1 TO N:FOR J=1
 TO N
170 A(I,J)=0:NEXT J
180 B(I)=0:F(I)=0:NEXT I
200 FOR L=1 TO K
210 GOSUB 800
300 FOR I=1 TO N:FOR J=1
 TO N
310 A(I,J)=A(I,J)+F(I)*F
(J):NEXT J
320 B(I)=B(I)+F(I)*Y(L):
NEXT I
330 NEXT L
340 CALL MATS(A(,),C(,),
B(),1,1,5,1,N,1,R)
350 IF R=-1 THEN PRINT "
Matrix is Singular":PAUS
E
360 PRINT #PN
400 FOR I=1 TO N
410 X$="A"&STR$(I)&" = "
420 PRINT #PN,X$;A(1,I)
430 IF PN=0 THEN PAUSE
440 NEXT I
450 PRINT #PN
500 INPUT "Display Resid
uals (Y/N)? ";A$
510 S1=0:S2=0
```

Line 110 is the printer useage subprogram. If the user responds "N" to the prompt "Use Printer?" then variable PN is set to zero. If the user responds with "Y" then PN is set to one, and an additional prompt "Enter File Name:" appears. For the TI-74 the response is 12. The text inside the quotation marks in the call is printed and displayed.

Line 120 displays a message for two seconds to remind the user that he must set the regression functions properly.

Line 130 asks for the number of data pairs, a value which is required by the AU subprogram which follows.

Line 140 uses the AU subprogram to enter and edit the two one dimensional arrays for the independent (x) and dependent (y) variables. The printer option set by line 110 provides annotation on one line followed by the input value on the next line and spaces between the pairs of input values (see the sample printout on page 15).

Line 150 asks for the order of the least squares solution.

Lines 160-180 clear the matrix and vector which are used in the least squares solution. These values will have been set to zero at program entry with a RUN command. The values must be reset to zero to permit multiple passes through the program without reentry with a RUN command, say with edited data or a different order (See lines 700-740 below).

Least Squares Regression - (cont)

Lines 200-330 calculate the matrix and vector values needed for the least squares solution. The GOSUB 800 at line 210 calls the subroutine which provides user defined functions.

Line 340 calls the MATS subprogram to solve the linear equations generated in lines 200-330.

Line 350 provides an indication to the operator if the matrix is singular based on the "test" output from the MATS subprogram.

Lines 400-450 provide printout or display of the solution with annotation by recalling values from the first row of the A matrix, array1 in the MATS subprogram call.

Lines 500-600 permit printout or display of the residual errors with annotation. Lines 570-590 provide the annotation and the associated residual on the same line, a technique that we would have preferred with the AU call in line 140.

Line 610 accumulates the sums which are needed for calculation of the mean error and the standard error.

Lines 620-680 calculate the mean error and the standard error, and display or print the results with annotation.

Lines 700-740 provide options for editing of bad input data as determined from the residuals, or for selection of a different order of solution.

Lines 800 through 890 are the subroutine which provides the user defined functions. The function shown is for a polynomial curve fit.

```
520 FOR L=1 TO K
530 GOSUB 800
540 YF=0:FOR J=1 TO N
550 YF=YF+A(1,J)*F(J):NE
XT J
560 D=Y(L)-YF
570 IF A$="N"OR A$="n"TH
EN 610
580 P$="d"&STR$(L)&" = "
590 PRINT #PN,P$;D
600 IF PN=0 THEN PAUSE
610 S1=S1+D:S2=S2+D*D:NE
XT L
620 PRINT #PN
630 PRINT #PN,"Mean Erro
r = ";S1/L
640 IF PN=0 THEN PAUSE
650 PRINT #PN
660 PRINT #PN,"Standard
Error = ";SQR(S2/(K-N))
670 IF PN=0 THEN PAUSE
680 PRINT #PN
700 INPUT "Edit Input Da
ta <Y/N>? ";A$
710 IF A$="N"OR A$="n"TH
EN 730
720 CALL AU("X","Y",X(),
Y(),1,K,PN)
730 INPUT "Different ord
er <Y/N>? ";A$
740 IF A$="Y"OR A$="y"TH
EN 150
799 STOP
800 REM USER DEFINED FUN
CTIONS
810 F(1)=1
820 FOR W=2 TO N
830 F(W)=F(W-1)*X(L)
840 NEXT W
890 RETURN
```

The polynomial generating function in steps 800 through 890 of the baseline program can be replaced by other user-defined functions. The center column on page 14 illustrates the use of sin and (1 - cos) functions to permit solution of a problem such as that presented on V10N4P7. Line 810 scales the equal time increment x values (every six minutes for the problem shown) to angles by dividing by the Schuler frequency (84 minutes) and multiplying by 360 degrees. This problem is not the same as fitting a function A + Bcos(w) +Csin(w); the lack of an independent constant term leads to the non-zero mean for the residuals.

Now consider the problem of finding Fourier series coefficients of a periodic function. An example problem is provided with the MU-19 Discrete Fourier Series program in the manual for the Math/Utilities module for the TI-59. The twelve function values corresponding to twelve angles equally spaced over the range from 30 to 360 degrees are: 2.0, 3.2, 3.7, 2.5, 1.2, 1.5, 2.7, 2.4, 0, -3.3, -3.3, and 0.

Least Squares Regression - (cont)

The regression functions defined in the right hand column below permit the user to use this program to find the coefficients through the third harmonic. The multiplication of the X(L) value by 30 in line 810 allows the user to enter the sequence numbers (1 through 12) associated with each function value instead of entering the angles. Once the data entry is completed the solution appears after about one minute. The answers are in agreement with those on page 73 of the Math/Utilities manual if the user remembers that the A0 coefficient from the MU-19 program is twice the A1 coefficient from this program. Of course, the real power of using a least squares program is that there is no requirement for equally spaced data over the fundamental frequency range as with a discrete Fourier analysis.

```
Least Squares Fit

X(1)=
-7
Y(1)=
-81

X(2)=
-4
Y(2)=
 27

X(3)=
 0
Y(3)=
 3

X(4)=
 2
Y(4)=
 27

X(5)=
 5
Y(5)=
 243


A1 =  18.61841343
A2 =  26.3477128
A3 =  2.460625362

d1 = -35.75506659
d2 =  74.40243196
d3 = -15.61841343
d4 = -54.15634047
d5 =  31.12738854

Mean Error =
 3.166667E-12

Standard Error =
 74.02670139
```

```
800 REM USER DEFINED FUN
CTIONS
810 W=360*X(L)/84
820 F(1)=SIN(W)
830 F(2)=1-COS(W)
890 RETURN
```

```
Least Squares Fit

X(1)=
 6
Y(1)=
 1

X(2)=
 12
Y(2)=
 2

X(3)=
 18
Y(3)=
 3

X(4)=
 24
Y(4)=
 4

X(5)=
 30
Y(5)=
 5

X(6)=
 36
Y(6)=
 6


A1 =  .8934012668
A2 =  2.786304468

Mean Error =
 .0525609272

Standard Error =
 .3178704425
```

```
800 REM USER DEFINED FUN
CTIONS
810 W=30*X(L):F(1)=1
820 F(2)=COS(W)
830 F(3)=SIN(W)
840 F(4)=COS(2*W)
850 F(5)=SIN(2*W)
860 F(6)=COS(3*W)
870 F(7)=SIN(3*W)
890 RETURN
```

```
Least Squares Fit

X(1)=
 1
Y(1)=
 2

X(2)=
 2
Y(2)=
 3.2

X(3)=
 3
Y(3)=
 3.7

X(
  1)=
-3.3

X(12)=
 12
Y(12)=
 0


A1 =  1.05
A2 = -1.417222017
A3 =  1.885961277
A4 = -.55
A5 =  1.905255888
A6 =  .5833333333
A7 =  .0166666667
```

Least Squares Regression - (cont)

The user defined functions must be designated by F(1) up to F(8). Each pair of X,Y values are available in turn at the entry to the subroutine. To obtain a constant in the solution define one of the functions as one, as in line 810 of the program on page 13.

To modify the program on pages 12 and 13 for use with the Mathematics module installed in the CC-40:

* In line 100 change A(8,8) to A(8,9) if you wish to solve eighth order systems.
* Add line 250: 250 R = 1
* In line 350 change IF R=-1 to IF R=0
* In line 420 change A(1,I) to A(I,1)
* In line 550 change A(1,J) to A(J,1)

MORE ON NUMERIC REPRESENTATION - In his column "The Art of Programming" in the March 1987 issue of 80 Micro Bruce Tonkin discusses the effects of digital representation in a computer and proposes a sample problem to illustrate the effects. His sample program is similar to the one at the right. A user who repetitively enters the value 1.11 will find the following display of T for various machines:

```
10 Input X
20 T = T + X
30 PRINT T
40 GOTO 10
```

Commodore 64	TI-74/CC-40
1.11	1.11
2.22	2.22
3.33	3.33
4.44	4.44
5.55	5.55
6.66000001	6.66
7.77000001	7.77
8.88000001	8.88
9.99000001	9.99
11.1	11.1
12.21	12.21

The Radio Shack Color Computer yields the same output as the Commodore 64. The Radio Shack Model 100 yields the same output as the TI-74. Where does the "garbage" come from with the CC-64 and Color Computer? Tonkin explains that the effect is due to the digital representation in those machines, and goes on to explain that the effect would be eliminated if numbers were stored in BCD (base 10). That is exactly what the Model 100 does. The TI-74 and CC-40 actually use base 100. How can a user be sure which base is used another machine? The best way is to use the "Itsy Bit of Paranoia" from the February 1985 issue of BYTE. See V10N2P16 for a description of the test.

SOLUTION OF CUBIC EQUATIONS - Peter Messer writes that his tests show that most cubic equation solutions fail his "benchmark" test except the built-in program in the TI-95. His test equation is

$$x^3 - 2x^2 + (4/3)x - 2/9 = 0$$

which has two complex roots and one real root. The following table compares the results from various cubic solution programs with the exact solution:

Source	Re	Im	R3	Seconds
Exact Expression	$(4 + \sqrt[3]{2})/6$	$(\sqrt[3]{2}\ \sqrt{3})/6$	$(2 - \sqrt[3]{2})/3$	
15 Digits	0.87665 35083 15812	0.36370 78786 57240	0.24669 29833 68375	
Built-in TI-95	0.87665 35083 153	0.36370 78786 563	0.24669 29833 693	1
V7N4/5P30 TI-59	0.87666 42804 886	0.36368 92207 041	0.24667 14390 225	5
V7N4/5P30 TI-95	0.87665 35083 155	0.36370 78786 566	0.24669 29833 692	1
PPX 398001 TI-59	0.87666 42804 890	0.36368 92207 034	0.24667 14390 217	3
PPX 398001 TI-95	0.87665 35083 155	0.36370 78786 565	0.24669 29833 692	1
V10N1P23 TI-57	0.87665 35083 5	0.36370 78786	0.24669 29833 8	45
V10N1P23 TI-59	0.87665 35083 160	0.36370 78786 585	0.24669 29833 681	24
V10N1P23 TI-95	0.87665 35083 160	0.36370 78786 561	0.24669 29833 686	4

where the notation for the roots is from the built-in TI-95 program. Three existing cubic solution were analyzed by converting them to run on both the TI-59 and TI-95 and comparing the results. The TI-59 programs from V7N4/5P30 by Bjorn Gustavsson and from PPX 398001 by E. W. Kamen are based on the Cardan equations. The TI-57 solution from V10N1P23 by Ingvar Magnusson (translated by Robert Prins) uses the method of Newton-Raphson to find the first real root, and then solves the reduced equation with the quadratic formula.

Examination of the results in the table show that the solutions based on the Cardan equations are substantially degraded when run on the TI-59, but are accurate when run on the TI-95. The better solution on the TI-95 has been traced to the improved arithmetic of that device. For the test problem both of the Cardan-based solutions involve taking the cube root of a value which should be zero, and is zero in the TI-95, but which is -1e-14 in the TI-59. The cube root of that value is -0.00002 15443 ... which is the difference between the TI-59 solutions and the exact solution for R3. It seems reasonable to conclude that the better solution from the TI-95 built-in function is the result of improved arithmetic, not of a better algorithm.

For those who wish to pursue the problem further consider the equations on page 2 of the documentation for PPX program 398001. For Messer's test problem the value $(-E/2 + \sqrt{F})$ should be zero, but due to arithmetic errors the TI-59 yields -1e-14. The quantity $(-E/2 + \sqrt{F})$ is zero with the equivalent program on the TI-95. If you would like a copy of the six page documentation for PPX 398001 send one dollar and a SASE.

Solution of Cubic Equations - (cont)

What can a TI-59 user do if he wants more accuracy from cubic equation solutions than that attainable with Cardan-based solutions? Perhaps some member can develop a TI-59 solution based on the Cardan equations which is more tolerant of arithmetic errors. In the mean-time better answers seem to be available from a TI-59 version of Ingvar Magnusson's cubic solution for a TI-57 (V10N1P23). The listing for a program which will run in fast mode is:

000	91	R/S	040	03	03	080	48	EXC	120	43	RCL	160	00	0	200	50	I×I
001	43	RCL	041	22	INV	081	07	07	121	07	07	161	00	0	201	32	X:T
002	04	04	042	44	SUM	082	50	I×I	122	91	R/S	162	00	0	202	43	RCL
003	42	STO	043	04	04	083	34	ΓX	123	43	RCL	163	00	0	203	02	02
004	00	00	044	43	RCL	084	42	STO	124	05	05	164	00	0	204	50	I×I
005	43	RCL	045	05	05	085	07	07	125	91	R/S	165	00	0	205	22	INV
006	01	01	046	22	INV	086	43	RCL	126	43	RCL	166	00	0	206	77	GE
007	44	SUM	047	49	PRD	087	04	04	127	06	06	167	00	0	207	02	02
008	04	04	048	04	04	088	52	EE	128	91	R/S	168	00	0	208	10	10
009	43	RCL	049	02	2	089	22	INV	129	00	0	169	00	0	209	32	X:T
010	04	04	050	94	+/-	090	52	EE	130	00	0	170	00	0	210	43	RCL
011	42	STO	051	22	INV	091	75	-	131	00	0	171	00	0	211	03	03
012	05	05	052	49	PRD	092	48	EXC	132	00	0	172	00	0	212	50	I×I
013	44	SUM	053	07	07	093	08	08	133	00	0	173	00	0	213	77	GE
014	05	05	054	43	RCL	094	95	=	134	00	0	174	00	0	214	02	02
015	42	STO	055	07	07	095	29	CP	135	00	0	175	00	0	215	17	17
016	06	06	056	42	STO	096	67	EQ	136	00	0	176	00	0	216	32	X:T
017	42	STO	057	05	05	097	01	01	137	00	0	177	76	LBL	217	85	+
018	07	07	058	49	PRD	098	02	02	138	00	0	178	11	A	218	01	1
019	43	RCL	059	07	07	099	61	GTO	139	00	0	179	42	STO	219	95	=
020	00	00	060	48	EXC	100	00	00	140	00	0	180	01	01	220	55	÷
021	44	SUM	061	06	06	101	01	01	141	00	0	181	91	R/S	221	02	2
022	04	04	062	22	INV	102	25	CLR	142	00	0	182	76	LBL	222	95	=
023	49	PRD	063	44	SUM	103	43	RCL	143	00	0	183	12	B	223	65	×
024	04	04	064	07	07	104	04	04	144	00	0	184	42	STO	224	43	RCL
025	49	PRD	065	29	CP	105	66	PAU	145	00	0	185	02	02	225	03	03
026	04	04	066	43	RCL	106	81	RST	146	00	0	186	91	R/S	226	69	OP
027	44	SUM	067	07	07	107	76	LBL	147	00	0	187	76	LBL	227	10	10
028	05	05	068	32	X:T	108	15	E	148	00	0	188	13	C	228	95	=
029	49	PRD	069	77	GE	109	43	RCL	149	00	0	189	42	STO	229	94	+/-
030	05	05	070	00	00	110	07	07	150	00	0	190	03	03	230	42	STO
031	49	PRD	071	74	74	111	29	CP	151	00	0	191	91	R/S	231	04	04
032	06	06	072	48	EXC	112	67	EQ	152	00	0	192	76	LBL	232	60	DEG
033	43	RCL	073	07	07	113	01	01	153	00	0	193	14	D	233	22	INV
034	02	02	074	34	ΓX	114	23	23	154	00	0	194	09	9	234	57	ENG
035	44	SUM	075	44	SUM	115	25	CLR	155	00	0	195	69	OP	235	04	4
036	05	05	076	05	05	116	35	1/X	156	00	0	196	17	17	236	05	5
037	44	SUM	077	22	INV	117	43	RCL	157	00	0	197	25	CLR	237	30	TAN
038	06	06	078	44	SUM	118	05	05	158	00	0	198	43	RCL	238	33	X^2
039	43	RCL	079	06	06	119	91	R/S	159	00	0	199	01	01	239	86	STF

User Instructions:

1. Enter the value A and press A.

2. Enter the value B and press B.

3. Enter the value C and press C.

4. Press D to start a solution. The program calculates an estimate for the first root using one-half of Magnusson's rule for a maximum value from V10N1P23 and stops with a flashing "1." in the display.

5. Press 7 and then EE and wait for a real root to appear in the display. To see the other roots press E.

6.a. If the display is flashing the result is the real part of a complex root. Press R/S to see the imaginary part.

6.b. If the display is not flashing the result is the second real root. Press R/S to see the third real root.

Solution of Cubic Equations - (cont)

The solution portion of the program is a brute force translation of the TI-57 program. A built-in comparison of successive iterations of the solution relieves the operator from watching the display to determine when the solution has converged. User defined keys and labels have been added for input and control. The table shows that the program obtains results for the Messer benchmark which are correct to eleven digits, but execution is slow. Even in fast mode the solution requirs 24 seconds.

POWERS AND ROOTS FOR NEGATIVE ARGUMENTS - P. Hanson

A listing of the TI-95 equivalent of the cubic solution from V7N4/5P30 appears at the right. Subroutine B1 (steps 044-058), which corresponds to subroutine A' in the TI-59 program, provides the critical cube root calculations discussed on page 17. A listing of subroutine A' from the TI-59 program also appears at the right.

```
0000 DFN F1: a @A1
0007 DFN F2: b @A2
0014 DFN F3: c @A3
0021 DFN F4: d @A4
0028 DFN F5: e @A5 HLT
0036 LBL B1 - RCL A =
0043 RTN
0044 LBL B2 STO I ABS
0050 INV y^x 3* RCL I
0056 SGN = RTN
0059 LBL A1 STO A HLT
0065 LBL A2 STO B HLT
0071 LBL A3 RF 00 SF 15
0078 NOP x~t RCL B /3
0084 ST/ A - RCL A x^2 =
0091 +/- STO E x~t /2+
0098 RCL A *( x^2 -
0104 RCL B /2= +/- STO C
0112 STO G x^2 - x~t *
0118 x^2 = IF> H GTL B3
0125 IF= H GTL B3 RCL G
0132 / RCL E SQR y^x 3=
0139 INV COS /3= STO F
0146 SF 00
0148 LBL B4 120 ST+ F 2*
0158 RCL E SQR * RCL F
0164 COS GTL B1
0168 LBL B3 SQR ST+ C
0174 +/- + RCL G =
0179 SBL B2 EXC C SBL B2
0187 CE STO B + RCL C
0193 GTL B1
0196 LBL A4 TF 00 GTL B4
0204 RCL B + RCL C =/2
0212 +/- GTL B1
0216 LBL A5 CLR TF 00
0222 GTL B4 3 SQR /2*(
0231 RCL B - RCL C = HLT
```

```
007  76 LBL
008  16 A'
009  22 INV
010  45 Y^X
011  32 X:T
012  03   3
013  65   x
014  32 X:T
015  69 OP
016  10  10
017  95   =
018  92 RTN
```

The TI-59 solution depends on a characteristic of the y^x and INV y^x functions where a negative argument is changed to a positive value, and the function is calculated as though the argument were positive, but an error is indicated with a flashing display. The calculator will not stop unless Flag 8 is set, which is not the case in this program.

The TI-95 reacts in an entirely different way to a negative a argument for INV y^x. From the keyboard the sequence 64 +/- INV y^x 3 + will yield the error message "INVALID ARGUMENT" and the user can go no further without pressing clear. If that same sequence is part of a progam, for example

64 +/- INV y^x 3 + 6 = HLT

then the program will stop with 6.333333333 in the display, and the error annunciator will be set. For a different root, say 5, the display will be 6.2. In general, for the program sequence

A +/- INV y^x n + K =

the result will be K + 1/n and the argument A has been discarded. Clearly, the sequence from the TI-59 would not work in a TI95 program.

A similar effect occurs with the y^x function in the TI-95, but only if n is not an integer. Then for

A +/- y^x n + K =

the result will be K + n , and the argument A is discarded as with the INV y^x function.

<u>Exact Factorials on the TI-95</u> - V8N4P5 presented a TI-59 program for calculating exact factorials in fast mode. The TI-59 program was by Peter Messer. The TI-95 program at the right is a direct translation of Lem Matteson's modification of Peter's program which appeared on V8N5P3. Pertinent changes introduced as part of the translation include:

* The absolute addressing required by the TI-59 fast mode was converted to label addressing for ease in translation.

* t register comparisons are not available on the TI-95. The IF= 100 GTL B3 sequence at steps 045-050 provides the equivalent of an EQ on the TI-59. The sequence assumes that register 100 contains zero. That is assured by the CMS at step 078. A similar equivalent EQ sequence is the IF= 100 GTL B5 at steps 151-156.

* The INV TF 74 BRK sequence at steps 181-184 stops the calculator with each block of the solution in the display if the PC-324 is not connected.

* The labelling includes examples of the use of single letters with a space either before the letter as with labels □A and □D or after the letter as with labels B□ and C□.

<u>User Instructions</u>:

1. See (n!) and (n/m) in the windows over F1 and F2 at the start of the program.

2. To find n! enter n in the display and press F1. The complete solution will printed if a PC-324 is connected. The number of trailing zeroes is printed last with a minus sign. If a printer is not connected the calculator will stop with the first block of the result in the display. A (GO) in the window above F1 indicates that additional blocks are available. Press F1 to see the next block. When the calculator stops with (n!) and (n/m) in the windows above F1 and F2 the last block is in the display. The program is at step 1 ready for a new problem.

3. At step 1, to find n!/m! press F2. See (m) and (n) in the windows over F1 and F2. Enter m, the denominator, first and press F1. Then enter n and press F2. The solution is made available in the same manner as that for n!.

The program finds 100! in 1 minute 50 seconds. The TI-59 program running in fast mode required 10 minutes 30 seconds. Sample printouts appear at the right for n! for n = 34 and 100, and for n!/m! for n = 100 and m = 50.

```
0000 DFN F1: n!@ A
0007 DFN F2:n/m@ D HLT
0015 LBL B0 1 ST+ 098
0022 LBL B1 STO 000
0028 LBL B2 RCL IND 000
0035 * RCL 098 + RCL 094
0043 INT = IF= 100
0048 GTL B3 / RCL 099 =
0056 STO 094 FRC *
0061 RCL 099 =
0065 EXC IND 000 INC 000
0072 GTL B2
0075 LBL AA CMS 1 EE 10
0083 STO 099 CLR 1
0088 STO 001 2 STO 098
0095 RTN
0096 LBL B  PRT x~t
0101 SBL AA x~t ST+ 001
0108 ST+ 098 HLT
0112 LBL  A x~t SBL AA
0119 x~t
0120 LBL C  PRT -
0125 RCL 098 +1 ST- 098
0133 = ADV STO 095
0138 GTL B0
0141 LBL B3 INV INC 000
0148 RCL 000 IF= 100
0154 GTL B5 STO 097
0160 DSZ 095 GTL B0
0166 RCL 097 STO 000
0172 LBL B4 CLR
0176 RCL IND 000 PRT INV
0182 TF 74 BRK DSZ 000
0188 GTL B4 RCL 096 *10=
0198 +/- PRT GTO 0000
0203 LBL B5 RCL 097
0209 STO 000 0
0213 LBL B6 EXC IND 000
0220 DSZ 000 GTL B6 1
0227 ST+ 096 GTL B1
0233 LBL  D
0236 DFN F1: m @B
0243 DFN F2: n @C  HLT
```

```
        34.

 295232799.
 396041408.
4761860964.
3520000000.
         0.
```

```
        50.
       100.

      3068.
5187562549.
6603720273.
 459529469.
7392284597.
2168468895.
9447786986.
9821589587.
7235507200.
       -10.
```

```
       100.

  93326215.
4439441526.
8169923885.
6266700490.
7159682643.
8162146859.
2963895217.
5999932299.
1560894146.
3976156518.
2862536979.
2082722375.
8251185210.
9168640000.
       -20.
```

<u>EXTENDED PRECISION MULTIPLICATION</u> - Robert Prins. This program is based on an earlier program "Extended Multiplication with the TI-58" which appeared in the November 1979 issue of BYTE. The author was Michael E. Manwaring. In this modification:

* The multiplication loop is optimized.

* The read-out routine only displays the answer, and not parts of the multiplicands.

* The program was converted to fast mode for increased speed.

<u>User Instructions:</u>

1. Initialize by pressing E.

2. Initialize for entry of the first multiplicand by pressing A.

3. Divide the multiplicand into groups of 6 digits. If the number of digits is not evenly divisible by 6, the first group may contain less than 6 digits. Enter each group in sequence into the display and press R/S. The group containing the MSD is entered first.

4. Initialize for entry of the second multiplicand by pressing B.

5. Enter the second multiplicand as in step 3.

6. Press C to enter fast mode. After a few seconds the calculator will stop with a flashing "1." in the display. Press 7 and then EE. When the solution is complete the calculator will stop with a zero in the display.

7. Recall the product by pressing D, followed by successive R/S commands to read out additional groups. The read-out is complete when a flashing zero appears in the display.

<u>Listing:</u>

000	92	RTN	027	73	RC*	054	42	STO	081	25	CLR	108	12	B	135	01	01
001	43	RCL	028	00	00	055	01	01	082	81	RST	109	43	RCL	136	27	27
002	00	00	029	54	)	056	69	OP	083	76	LBL	110	00	00	137	69	OP
003	42	STO	030	55	÷	057	25	25	084	15	E	111	48	EXC	138	66	66
004	02	02	031	01	1	058	43	RCL	085	29	CP	112	01	01	139	08	8
005	42	STO	032	52	EE	059	05	05	086	01	1	113	69	OP	140	00	0
006	05	05	033	06	6	060	48	EXC	087	00	0	114	31	31	141	92	RTN
007	42	STO	034	72	ST*	061	00	00	088	69	OP	115	92	RTN	142	00	0
008	06	06	035	00	00	062	97	DSZ	089	17	17	116	72	ST*	143	00	0
009	53	(	036	75	-	063	04	04	090	47	CMS	117	00	00	144	00	0
010	43	RCL	037	22	INV	064	00	00	091	08	8	118	69	OP	145	00	0
011	01	01	038	59	INT	065	09	09	092	48	EXC	119	20	20	146	00	0
012	85	+	039	64	PD*	066	42	STO	093	00	00	120	69	OP	147	00	0
013	69	OP	040	00	00	067	00	00	094	92	RTN	121	24	24	148	00	0
014	32	32	041	54	)	068	22	INV	095	76	LBL	122	61	GTO	149	00	0
015	43	RCL	042	69	OP	069	44	SUM	096	11	A	123	01	01	150	00	0
016	03	03	043	20	20	070	06	06	097	92	RTN	124	15	15	151	76	LBL
017	42	STO	044	74	SM*	071	73	RC*	098	72	ST*	125	76	LBL	152	13	C
018	07	07	045	00	00	072	00	00	099	00	00	126	14	D	153	50	I×I
019	53	(	046	69	OP	073	67	EQ	100	69	OP	127	73	RC*	154	60	DEG
020	53	(	047	31	31	074	00	00	101	20	20	128	00	00	155	04	4
021	73	RC*	048	97	DSZ	075	79	79	102	69	OP	129	69	OP	156	05	5
022	02	02	049	07	07	076	69	OP	103	23	23	130	30	30	157	30	TAN
023	65	×	050	00	00	077	36	36	104	61	GTO	131	99	PRT	158	33	X^2
024	73	RC*	051	19	19	078	87	IFF	105	00	00	132	92	RTN	159	86	STF
025	01	01	052	00	0	079	69	69	106	97	97	133	97	DSZ			
026	85	+	053	54	)	080	30	TAN	107	76	LBL	134	06	06			

Extended Precision Multiplication - (cont)

Editor's Note: Steps 078 through 080 look strange, but Robert's transmittal letter stated that they were OK. Closer examination shows that if x = t at the EQ test at steps 073-075 then the program jumps into the middle of the IFF 69 sequence at steps 078=079 and performs an OP 30 command. If x = t then an OP 36 is performed, followed by two commands which have no effect on the result, IFF 69 and TAN.

As I used the program I found that I wanted a record of the input multiplicands. I modified the program to provide that capability, and in the process replaced the unusual command sequence at steps 078-080 of Robert's program with a more conventional sequence at a cost of one addditional program step. The user instructions are the same, but if a PC-100 is connected then each block of input data will be printed as it is entered. The revised listing is:

000	92	RTN	040	00	00	080	69	DP	120	99	PRT
001	43	RCL	041	54	)	081	30	30	121	72	ST*
002	00	00	042	69	DP	082	25	CLR	122	00	00
003	42	STD	043	20	20	083	81	RST	123	69	DP
004	02	02	044	74	SM*	084	76	LBL	124	20	20
005	42	STD	045	00	00	085	15	E	125	69	DP
006	05	05	046	69	DP	086	29	CP	126	24	24
007	42	STD	047	31	31	087	01	1	127	61	GTD
008	06	06	048	97	DSZ	088	00	0	128	01	01
009	53	(	049	07	07	089	69	DP	129	19	19
010	43	RCL	050	00	00	090	17	17	130	76	LBL
011	01	01	051	19	19	091	47	CMS	131	14	D
012	85	+	052	00	0	092	08	8	132	98	ADV
013	69	DP	053	54	)	093	48	EXC	133	73	RC*
014	32	32	054	42	STD	094	00	00	134	00	00
015	43	RCL	055	01	01	095	92	RTN	135	69	DP
016	03	03	056	69	DP	096	76	LBL	136	30	30
017	42	STD	057	25	25	097	11	A	137	99	PRT
018	07	07	058	43	RCL	098	98	ADV	138	92	RTN
019	53	(	059	05	05	099	92	RTN	139	97	DSZ
020	53	(	060	48	EXC	100	99	PRT	140	06	06
021	73	RC*	061	00	00	101	72	ST*	141	01	01
022	02	02	062	97	DSZ	102	00	00	142	33	33
023	65	×	063	04	04	103	69	DP	143	69	DP
024	73	RC*	064	00	00	104	20	20	144	68	68
025	01	01	065	09	09	105	69	DP	145	00	0
026	85	+	066	42	STD	106	23	23	146	92	RTN
027	73	RC*	067	00	00	107	61	GTD	147	00	0
028	00	00	068	22	INV	108	00	00	148	00	0
029	54	)	069	44	SUM	109	99	99	149	00	0
030	55	÷	070	06	06	110	76	LBL	150	00	0
031	01	1	071	73	RC*	111	12	B	151	76	LBL
032	52	EE	072	00	00	112	98	ADV	152	13	C
033	06	6	073	67	EQ	113	43	RCL	153	50	I×I
034	72	ST*	074	00	00	114	00	00	154	60	DEG
035	00	00	075	80	80	115	48	EXC	155	04	4
036	75	-	076	69	DP	116	01	01	156	05	5
037	22	INV	077	36	36	117	69	DP	157	30	TAN
038	59	INT	078	25	CLR	118	31	31	158	33	X^2
039	64	PD*	079	81	RST	119	92	RTN	159	86	STF

```
    12.
599210.
498948.
731648.

    17.
320508.
 75688.
772935.

   218.
224727.
194344.
280714.
235789.
911720.
346880.
```

The sample printout at the right above is for the problem $\sqrt[3]{2} * \sqrt{3}$.

TI-66 CODE TABLE - Earlier issues have discussed some unpublished capabilities of the TI-66; e.g., see V10N4P13. V11N1P24 and V11N2P23 presented examples of the use of synthesized hexadecimal code in programs. Dave Leising and Robert Prins have examined the various possibilities for TI-66 code. The table on the next two pages summarizes the results of their investigations. The four entries in each block are defined as:

CODE = the actual internal code in hexadecimal.

DISP = how the code shows in a data display.

MNEM = how the code shows in LRN mode.

LBL = can the code be used as a label, Y or N?

CODE	DISP
MNEM	
LBL	

TI-66 Code Table - (cont)

00 00 0 No	01 01 1 No	02 02 2 No	03 03 3 No	04 04 4 No	05 05 5 No	06 06 6 No	07 07 7 No
10 10 A Yes	11 11 ** Yes	12 12 B Yes	13 13 ** Yes	14 14 C Yes	15 15 ** Yes	16 16 D Yes	17 17 ** Yes
20 20 ** No	21 21 ** No	22 22 ** No	23 23 OP Yes	24 24 X⇌T Yes	25 25 R/S Yes	26 26 RST Yes	27 27 GTO Yes
30 30 ** Yes	31 31 INV Yes	32 32 STO Yes	33 33 RCL Yes	34 34 SUM Yes	35 35 LNX Yes	36 36 1/X Yes	37 37 X↑2 Yes
40 40 ADV Yes	41 41 CP Yes	42 42 FIX Yes	43 43 ENG Yes	44 44 P-R Yes	45 45 DMS Yes	46 46 PI Yes	47 47 SIN Yes
50 50 A' Yes	51 51 ** Yes	52 52 B' Yes	53 53 ** Yes	54 54 C' Yes	55 55 ** Yes	56 56 D' Yes	57 57 ** Yes
60 60 PAR Yes	61 61 ** No	62 62 ** No	63 63 NOP Yes	64 64 X=T Yes	65 65 DSZ Yes	66 66 PAU Yes	67 67 IFF Yes
70 70 ** No	71 71 *IN No	72 72 IND No	73 73 EXC Yes	74 74 PRD Yes	75 75 LOG Yes	76 76 INT Yes	77 77 X Yes
80 80 10 No	81 81 11 No	82 82 12 No	83 83 13 No	84 84 14 No	85 85 15 No	86 86 16 No	87 87 17 No
90 90 26 No	91 91 27 No	92 92 28 No	93 93 29 No	94 94 30 No	95 95 31 No	96 96 32 No	97 97 33 No
A0 00	A1 01 ST* Yes	A2 02	A3 03 OP* Yes	A4 04	A5 05 RTN Yes	A6 06	A7 07 GO* Yes
B0 10	B1 11 ST* Yes	B2 12 ST* Yes	B3 13 RC* Yes	B4 14 SM* Yes	B5 15 90 No	B6 16 91 No	B7 17 92 No
C0 20 42 No	C1 21 43 No	C2 22 44 No	C3 23 45 No	C4 24 46 No	C5 25 47 No	C6 26 48 No	C7 27 49 No
D0 30 58 No	D1 31 59 No	D2 32 60 No	D3 33 61 No	D4 34 62 No	D5 35 63 No	D6 36 64 No	D7 37 65 No
E0 40 74 No	E1 41 75 No	E2 42 76 No	E3 43 77 No	E4 44 78 No	E5 45 79 No	E6 46 80 No	E7 47 81 No
F0 50	F1 51 ST* Yes	F2 52	F3 53 EX* Yes	F4 54 PD* Yes	F5 55 00 No	F6 56 01 No	F7 57 02 No

TI-66 Code Table - (cont)

08 08 8 No	09 09 9 No	0A 00	0B 01	0C 02	0D 03	0E 04	0F 05
18 18 E Yes	19 19 . Yes	1A 10 +/- Yes	1B 11 EE Yes	1C 12 =	1D 13	1E 14	1F 15
28 28 SBR Yes	29 29 LBL No	2A 20 (Yes	2B 21) Yes	2C 22 CE Yes	2D 23 CLR Yes	2E 24	2F 25
38 38 √x Yes	39 39 Y↕X Yes	3A 30 + Yes	3B 31 - Yes	3C 32 x Yes	3D 33 / Yes	3E 34	3F 35
48 48 COS Yes	49 49 TAN Yes	4A 40	4B 41	4C 42	4D 43	4E 44	4F 45
58 58 E' Yes	59 59 PRT Yes	5A 50 LST Yes	5B 51	5C 52	5D 53	5E 54	5F 55
68 68 STF Yes	69 69 X≥T Yes	6A 60 CSR Yes	6B 61 CMS Yes	6C 62	6D 63	6E 64	6F 65
78 78 $\bar{x}$ Yes	79 79 Σ+ Yes	7A 70 TRC Yes	7B 71 GRD Yes	7C 72 RAD Yes	7D 73 DEG Yes	7E 74	7F 75
88 88 18 No	89 89 19 No	8A 80 20 No	8B 81 21 No	8C 82 22 No	8D 83 23 No	8E 84 24 No	8F 85 25 No
98 98 34 No	99 99 35 No	9A 90 36 No	9B 91 37 No	9C 92 38 No	9D 93 39 No	9E 94 40 No	9F 95 41 No
A8 08	A9 09 RTN Yes	AA 00	AB 01	AC 02	AD 03	AE 04	AF 05 GO* Yes
B8 18 93 No	B9 19 94 No	BA 10 95 No	BB 11 96 No	BC 12 97 No	BD 13 98 No	BE 14 99 No	BF 15
C8 28 50 No	C9 29 51 No	CA 20 52 No	CB 21 53 No	CC 22 54 No	CD 23 55 No	CE 24 56 No	CF 25 57 No
D8 38 66 No	D9 39 67 No	DA 30 68 No	DB 31 69 No	DC 32 70 No	DD 33 71 No	DE 34 72 No	DF 35 73 No
E8 48 82 No	E9 49 83 No	EA 40 84 No	EB 41 85 No	EC 42 86 No	ED 43 87 No	EE 44 88 No	EF 45 89 No
F8 58 03 No	F9 59 04 No	FA 50 05 No	FB 51 06 No	FC 52 07 No	FD 53 08 No	FE 54 09 No	FF 55

<u>EXTENDED PRECISION DIVISION</u> - Peter Messer. This fast mode program provides the capability to divide two numbers of up to 210 digits. In addition, the editor modified Peter's program to print the denominator and the numerator as they are entered, and to insert NOP's after the PRT commands to permit use of the program without a printer.

<u>User Instructions</u>:

1. Initialize for entry of the denominator by pressing A.

2. Divide the denominator into groups of 6 digits. If the number of digits is not evenly divisible by 6, add trailing zeroes. Enter each group of six digits into the display and press R/S. If the PC-100 is connected, each block of 6 digits will be printed as it is entered. The group containing the MSD is entered first. Up to 35 blocks may be entered.

> NOTE: These rules for grouping the digits are different from the rules for the extended precision multiply on page 20. If you forget and let the first group contain less than 6 digits, you may obtain a correct solution, or you may not. So, to be sure of getting correct answers, be sure that the first group contains 6 digits, and add trailing zeroes to the last group as needed.

3. Initialize for entry of the numerator by pressing B.

4. Enter the numerator as in step 2.

5. Enter the number of desired quotient blocks (no limit) and press C.

6. After a few seconds the calculator will stop with a flashing "1." in the display. Press 7 and then EE. Quotient blocks are printed. A sample printout which is the inverse of the multiplication problem on page 21 appears at the right. In some cases the first block of the quotient will contain 7 digits. The program will run until:

   a. The desired blocks have been printed. The display will be a flashing "0.".

   b. The quotient terminates. A "-1." will be printed. The display will be a flashing "0.".

7. In case a. you can obtain additional digits of the quotient by entering the number of additional blocks, pressing D, and going to step 6.

8. If you are not using a printer change the NOP's at program steps 133, 300, and 304 to R/S's. Then, the calculator will stop as each block of the quotient is completed. A "-1." will be displayed if the quotient terminates as in step 6.b. above.

```
 125992.
 104989.
 487316.
 480000.

 218224.
 727194.
 344280.
 714235.
 789911.
 720346.
 880000.

1732050.
 807568.
 877293.
 500000.
     -1.
```

Extended Precision Division - (cont)

Program Listing:

000	91	R/S	054	42	STO	108	65	×	162	78	78	216	01	1	270	02	02
001	25	CLR	055	00	00	109	43	RCL	163	65	×	217	61	GTO	271	33	33
002	05	5	056	25	CLR	110	76	76	164	73	RC*	218	01	01	272	75	-
003	32	X:T	057	98	ADV	111	95	=	165	00	00	219	91	91	273	43	RCL
004	43	RCL	058	91	R/S	112	58	FIX	166	95	=	220	43	RCL	274	36	36
005	73	73	059	99	PRT	113	05	05	167	55	÷	221	74	74	275	95	=
006	67	EQ	060	72	ST*	114	52	EE	168	43	RCL	222	42	STO	276	94	+/-
007	01	01	061	00	00	115	22	INV	169	76	76	223	00	00	277	67	EQ
008	46	46	062	69	OP	116	52	EE	170	95	=	224	85	+	278	02	02
009	61	GTO	063	20	20	117	22	INV	171	42	STO	225	03	3	279	82	82
010	00	00	064	61	GTO	118	58	FIX	172	75	75	226	06	6	280	42	STO
011	96	96	065	00	00	119	42	STO	173	22	INV	227	95	=	281	71	71
012	76	LBL	066	58	58	120	78	78	174	59	INT	228	42	STO	282	43	RCL
013	13	C	067	76	LBL	121	22	INV	175	65	×	229	77	77	283	71	71
014	98	ADV	068	11	A	122	59	INT	176	43	RCL	230	00	0	284	67	EQ
015	42	STO	069	47	CMS	123	29	CP	177	76	76	231	42	STO	285	02	02
016	72	72	070	08	8	124	67	EQ	178	85	+	232	71	71	286	97	97
017	43	RCL	071	69	OP	125	02	02	179	43	RCL	233	85	+	287	67	EQ
018	74	74	072	17	17	126	20	20	180	71	71	234	43	RCL	288	01	01
019	75	-	073	01	1	127	43	RCL	181	75	-	235	78	78	289	27	27
020	43	RCL	074	52	EE	128	78	78	182	32	X:T	236	65	×	290	01	1
021	00	00	075	06	6	129	59	INT	183	95	=	237	73	RC*	291	22	INV
022	85	+	076	42	STO	130	42	STO	184	94	+/-	238	00	00	292	44	SUM
023	03	3	077	76	76	131	78	78	185	29	CP	239	95	=	293	78	78
024	06	6	078	25	CLR	132	99	PRT	186	22	INV	240	55	÷	294	61	GTO
025	95	=	079	98	ADV	133	68	NOP	187	77	GE	241	43	RCL	295	01	01
026	77	GE	080	72	ST*	134	97	DSZ	188	02	02	242	76	76	296	27	27
027	03	03	081	00	00	135	72	72	189	11	11	243	95	=	297	43	RCL
028	14	14	082	91	R/S	136	01	01	190	32	X:T	244	42	STO	298	78	78
029	94	+/-	083	99	PRT	137	46	46	191	42	STO	245	75	75	299	99	PRT
030	44	SUM	084	69	OP	138	05	5	192	71	71	246	22	INV	300	68	NOP
031	74	74	085	20	20	139	42	STO	193	01	1	247	59	INT	301	01	1
032	61	GTO	086	61	GTO	140	73	73	194	22	INV	248	65	×	302	94	+/-
033	03	03	087	00	00	141	25	CLR	195	44	SUM	249	43	RCL	303	99	PRT
034	14	14	088	80	80	142	69	OP	196	77	77	250	76	76	304	68	NOP
035	76	LBL	089	76	LBL	143	99	99	197	32	X:T	251	75	-	305	61	GTO
036	12	B	090	14	D	144	66	PAU	198	63	EX*	252	73	RC*	306	01	01
037	43	RCL	091	42	STO	145	81	RST	199	77	77	253	77	77	307	41	41
038	01	01	092	72	72	146	29	CP	200	32	X:T	254	95	=	308	00	0
039	65	×	093	61	GTO	147	43	RCL	201	43	RCL	255	94	+/-	309	00	0
040	43	RCL	094	03	03	148	74	74	202	75	75	256	67	EQ	310	00	0
041	76	76	095	14	14	149	42	STO	203	59	INT	257	02	02	311	00	0
042	85	+	096	43	RCL	150	00	00	204	97	DSZ	258	61	61	312	00	0
043	43	RCL	097	36	36	151	85	+	205	00	00	259	42	STO	313	00	0
044	02	02	098	65	×	152	03	3	206	01	01	260	71	71	314	60	DEG
045	95	=	099	43	RCL	153	06	6	207	60	60	261	01	1	315	04	4
046	42	STO	100	76	76	154	95	=	208	61	GTO	262	22	INV	316	05	5
047	79	79	101	85	+	155	42	STO	209	00	00	263	44	SUM	317	30	TAN
048	43	RCL	102	43	RCL	156	77	77	210	96	96	264	77	77	318	33	x^2
049	00	00	103	37	37	157	00	0	211	85	+	265	43	RCL	319	86	STF
050	42	STO	104	95	=	158	42	STO	212	43	RCL	266	75	75			
051	74	74	105	55	÷	159	71	71	213	76	76	267	59	INT			
052	03	3	106	43	RCL	160	85	+	214	95	=	268	97	DSZ			
053	06	6	107	79	79	161	43	RCL	215	32	X:T	269	00	00			

THE TI-59 BECOMES AN ORPHAN - P. Hanson. A non-member called for help with repair or replacement of his TI-59. He reported that TI is no longer repairing the TI-59. A call to 1-800-TI-CARES verified that repair service has been discontinued for the both the TI-59 and the PC-100. Repair service continues to be available for the TI-58C and the TI-66. TI is offering replacement of a defective TI-59 with a TI-74 or TI-95, and replacement of a defective PC-100 with a PC-324. The replacement fees are:

TI-74 for a TI-59 + $99.00 + $6.00 shipping + local sales tax.

TI-95 for a TI-59 + $144.00 + $6.00 shipping + local sales tax.

PC-324 for a PC-100 + $64.00 + $6.00 shipping + local sales tax.

In the case of the TI-59 replacements a user can get a better price from Elek-Tek.

MORE ON ERRORS WITH LINEAR EQUATION SOLUTIONS - P. Hanson.

Some results for the Walters test of linear equation solutions include:

		Ohlsson/Prins		
Exact	TI-59	TI-95	TI-59 Plus	HP-15
1	0.999999996	0.999999954	1.000000028	1.000000002
1	0.999999981	0.999999969	1.000000021	0.999999990
1	0.999999993	0.999999969	1.000000026	0.999999988
1	0.999999997	0.999999976	1.000000012	0.999999987
1	0.999999992	0.999999992	1.000000007	0.999999981
1	0.999999993	0.999999990	1.000000002	0.999999982
1	1.000000013	1.000000001	0.999999997	1.000000001
Max Error	1.9E-8	4.6E-8	2.8E-8	1.9E-8
RMS Error	1.0E-8	2.6E-8	1.7E-8	1.3E-8

where TI-59 Plus means a program which uses the double-divide technique to avoid the non-commutative multiply problem. As in V9N2P18 we see that the better solution vector does not always yield the better result on the Walters test.

PC-100 PRINTER PAPER - Somehow I obtained a roll of LABELON Thermal Calculator Paper Type CR-025. It delivers a well-defined black-on-white printout. Some of the best definition I have seen so far. I will report on storage tests next month. Unfortunately, the local office supply stores say that that item is not listed in their catalogs. Does anyone know of source? Do not confuse this with Type CR-022 which prints light-blue-on-white, barely readable.

AREA FINDING - An article "Calculating the Area of an Irregular Shape" by Rene Stolk and George Ettershank appears in the February 1987 issue of BYTE. This is one more discussion of the idea of finding irregular areas by decomposing the area into triangles which can then be solved with the formulas found in most analytic geometry texts. V8N2P6/7 presented a TI-59 program which would perform the same function. V11N2P22 described an article on the same subject which appeared in the September 1986 issue of Computer Shopper. A BASIC language program was included.

SOLUTION OF LINEAR EQUATIONS - Recent issues of Computer Shopper have presented a series of "Mathpak" articles written by Wynn Rostek. Among other items the series is working its way through the development of linear equations solutions and Gaussian elimination with emphasis on pitfalls, pivoting, and the like. The articles contain lots of tutorial material with examples.
