TI PPC NOTES

NEWSLETTER OF THE TI PROGRAMMABLE CALCULATOR CLUB

P.O. Box 1421, Largo, FL 34294

Volume 8, Number 5 September/October 1983

The dominant theme of this issue is optimization. Examples include the extension of exact factorial capability of the TI-59 to 610!, a better algorithm for the thirteen digit speedy factor finder, and an extension of the Grosh of Finn solution up to exponent 9.

There is also substantial coverage of the new calculators ranging from the TI-57LCD through the BA-55 to The TI-66. As of this writing the TI-66 continues to be unavailable. Perhaps by Christmas.

Some additional programs are included for the CC-40, with the emphasis on use of the Mathematics module. The important discovery is a way to use the matrix routines in a program including identification of the locations of the solution to a set of linear equations.

There is also an extensive list of errata for old issues as compiled by Robert Prins. There are even more waiting to be published in the next issue once I have verified the errors. The surprising aspect to this is that the errors have gone unreported for as long as three years. I would encourage all members to report errors promptly so that they may be corrected. Report even "obvious" errors. What is obvious to you may not be so obvious to other members.

Table of Contents

Magnetic Card Service

Magnetic cards will be provided for programs in this issue for a price of one dollar per card plus a stamped and self-addressed envelope. Details of the service appear in V8N1P32. Each program in this issue will fit on one card. In addition I will provide both the Matteson and Fast Mode Fujimoto factorial programs on one magnetic card.

ERRATA - Robert Prins of Holland has been reviewing old issues of TI PPC Notes and has found several errors not previously published:

SR-56 Speedy Factor Finder - The SR-56 SFF by Bill Skillman on V5N9/10P9 skips factors 11, 31, etc., the same error as in his TI-59 program on V5N6P7. His SR-56 SFF on V5N7P6 is correct.

Inverse List Print All - Fast Mode - V5N9/10P15. Due to the scaling routine used (steps 166 to 182) this program by John Worthington and Emil Regelman sometimes prints leading zeroes, especially for numbers consisting of a lot of leading nines. For examples try entries of 999999999.9999 and 9999999999999 .

Codebreaker SR-56 - This version of Mastermind for the SR-56 by Don O'Grady must have step 59 changed from a 2 (code 02) to x⇌t (code 32).

A-Maze-ing - Maurice Swinnen reported that his program in V6N4-5P23 would sometimes produce "duds". Changing the address at steps 426/427 from 429 to 434 (as suggested by Andreas Biek in V6N9-10P17) will eliminate the "dud" mode.

Roots of $ax^4 + bx^3 + cx^2 + dx + e = y$ - The listing for this TI-57 program by Isaac Sanchez is incorrect. A RCL 3 must be inserted between steps 38 and 39. (V7N1/2P14)

New CROM Discoveries - In this discussion by Robert Prins in V7N1/2P28 the SBR IND 00 at lines 32 and 43 of the text (don't count the blank lines) should be SBR IND 0 .

Lots of Pie - Robert asks: "Are we ever going to see the first 160 steps of program I and the first 40 steps of program IIB. I'd love to see them, even though we now have the Science et Vie program. Editor's Note - I have written to Bob Fruit for clarification on his "Lots of Pi" program in V7N4/5P27.

Min-Max Sorter - This problem was originally incorrectly stated in V7N1/2P9. V8N1P5 included a solution to the correctly stated problem which was submitted by Henrik Klein. In V8N2P4 I noted that Charlie Williamson's solution in V7N3P12 could not sort data pairs which were separated by a factor of 10^{13} or more, and used the example of starting with 100000000 in the t register and 0.00000001 in the display register. Henrik Klein's routine solves that problem satisfactorily, but will not work with other widely separated pairs. For example, try it with 1 in the display and 10^{-13} in the t register. The 1 will be returned properly to the display, but X⇌T will return a zero, not 10^{-13}, to the display. The routine works satisfactorily if the positions of the two numbers are reversed at the start.

Editor's Note: Robert submitted several more corrections and clarifications which I will present in subsequent issues when I have had time to verify them. Many of the corrections listed above are the result of incorrect listings, probably due to incorrect transcription. Until we get a printer for the CC-40 the potential for similar problems with the BASIC listings will exist.

MORE EXACT FACTORIALS IN FAST MODE - Don Graham and Lem Matteson were intrigued by Peter Messer's exact factorial algorithm (V8N3P10 and V8N4P5). Don modified the normal mode program (V8N3P10) to be more user friendly. Lem did the same for the fast mode version (V8N4P5). His program appears below. The instructions are:

(1) Enter the program.

(2) To find n! :

(a) Enter n and press A. The entered value is printed and the calculator stops with a flashing "1" in the display. Ignore the flashing and press 7 and then press EE. When the calculations are complete the factorial is printed. The number of trailing zeroes is printed last with a minus sign. The calculator stops with the number of trailing zeroes in the display.

(3) To find n!/m! :

(a) Enter m and press B. m is printed and the calculator stops with m in the display.

(b) Enter n and press C. The value of n is printed and the calculator stops with a flashing "1" in the display. Press 7 and then EE to enter fast mode. The printout format is the same as described in 2.a above.

Lem also provided a version for use without a printer. It is necessary to change steps 113 through 125 as indicated in the right hand column in the program listing below. The instructions are the same as listed above, but the calculator stops with the first part of the factorial in the display. The user presses R/S to see additional portions, followed by the number of trailing zeroes with a minus sign.

Program Listing:

000	91	R/S	032	00	00	064	32	X:T	096	00	00	128	97	97
001	01	1	033	69	OP	065	44	SUM	097	67	EQ	129	42	STO
002	44	SUM	034	20	20	066	01	01	098	01	01	130	00	00
003	98	98	035	61	GTO	067	44	SUM	099	27	27	131	00	0
004	42	STO	036	00	00	068	98	98	100	42	STO	132	63	EX*
005	00	00	037	06	06	069	91	R/S	101	97	97	133	00	00
006	73	RC*	038	76	LBL	070	76	LBL	102	97	DSZ	134	97	DSZ
007	00	00	039	16	A'	071	11	A	103	95	95	135	00	00
008	65	×	040	47	CMS	072	32	X:T	104	00	00	136	01	01
009	43	RCL	041	01	1	073	16	A'	105	01	01	137	32	32
010	98	98	042	00	0	074	32	X:T	106	43	RCL	138	01	1
011	85	+	043	69	OP	075	76	LBL	107	97	97	139	44	SUM
012	43	RCL	044	17	17	076	13	C	108	42	STO	140	96	96
013	94	94	045	01	1	077	99	PRT	109	00	00	141	61	GTO
014	59	INT	046	52	EE	078	75	-	110	25	CLR	142	00	00
015	95	=	047	01	1	079	43	RCL	111	73	RC*	143	04	04
016	67	EQ	048	00	0	080	98	98	112	00	00	144	29	CP
017	00	00	049	42	STO	081	85	+	113	99	PRT	145	02	2
018	93	93	050	99	99	082	01	1	114	97	DSZ	146	52	EE
019	55	÷	051	25	CLR	083	22	INV	115	00	00	147	01	1
020	43	RCL	052	01	1	084	44	SUM	116	01	01	148	02	2
021	99	99	053	42	STO	085	98	98	117	10	10	149	94	+/-
022	95	=	054	01	01	086	95	=	118	43	RCL	150	85	+
023	42	STO	055	02	2	087	98	ADV	119	96	96	151	01	1
024	94	94	056	42	STO	088	42	STO	120	65	×	152	95	=
025	22	INV	057	98	98	089	95	95	121	01	1	153	22	INV
026	59	INT	058	92	RTN	090	61	GTO	122	00	0	154	52	EE
027	65	×	059	76	LBL	091	01	01	123	95	=	155	58	FIX
028	43	RCL	060	12	B	092	44	44	124	94	+/-	156	00	00
029	99	99	061	99	PRT	093	69	OP	125	99	PRT	157	60	DEG
030	95	=	062	32	X:T	094	30	30	126	81	RST	158	86	STF
031	63	EX*	063	16	A'	095	43	RCL	127	43	RCL	159	40	IND

w/o printer

113	66	PAU
114	91	R/S
115	97	DSZ
116	00	00
117	01	01
118	10	10
119	43	RCL
120	96	96
121	65	×
122	01	1
123	00	0
124	95	=
125	94	+/-

MORE EXACT FACTORIALS IN FAST MODE (cont)

PPX program 398171 by Frank M. Fujimoto, dated 9/4/79, will find exact factorials to 610 factorial. The program uses data register R00 as a counter and the hierarchy registers for data manipulation. Data registers R01 through R99 are divided into two pseudo-registers each, seven digits after the decimal point and six before. (You will recognize that a similar scheme was used in the "1287 Digits of Pi" program, V8N3P8/9). Each trailing zero is truncated as it is found and the number of trailing zeroes is accumulated in a hierarchy register. If this were not done, only 556! could be calculated. The program multiplies by two factors at a time to increase speed.

Frank's program ran in normal mode--fast mode was unknown at the time. It would find 34! in 2 minutes 54 seconds, slightly faster than Peter Messer's normal mode program in V8N3P11. When converted to fast mode Frank's program will find 34! in 1 minute 16 seconds. As with the conversion of the "1287 Digits of Pi" program to fast mode (V8N4P25) the implementation of fast mode was considerably simplified by using Patrick Acosta's method for jumping to a location other than 001 at fast mode entry. The other major conversion involved replacement of subroutines with in-line code. The elimination of the subroutines accounts for the increase in operating speed by more than a factor of two. The penalty for using fast mode is a requirement to re-enter the program for each new factorial calculation. The primary deficiency relative to Peter Messer's program (see previous page of this issue) is the data packing method which permits extension to the 610!. A more complex procedure is required for retrieval of the solution; for example one such as that used with the "1287 Digits of Pi" program (V8N3P9). Thus, for factorials of 461! or less the use of Lem Matteson's modification of Peter Messer's program on page 3 of this issue is recommended.

Operating Instructions:

(1) Enter the magnetic card. You may use the startup partitioning of 6 Op 17 for recording the card. The program automatically changes the partitioning to 9 Op 17 for fast mode entry, and to 10 Op 17 for calculations once fast mode entry is complete.

(2) Enter the integer for which the factorial is desired into the display and press A. The calculator will run for about two seconds and stop with a flashing "1" in the display.

(3) Do not clear the flashing display, but press 7 and then EE in sequence. The calculator will run again, but will be in fast mode. You will not be able to interrupt the calculations with either R/S or RST.

(4) The calculator will stop with the number of trailing zeroes in the display. Pressing X ⇌ T will indicate the data register at which the factorial begins. You may then read out the exact factorial less the trailing zeroes by recalling the data registers in ascending sequence up through register 99, recording the six digits to the left of the decimal point for each data register, pressing INV Int and recording the seven digits to the right of the decimal point for that data register, and so on for succeeding registers.

(5) To find another factorial, re-enter the card and return to step 2 above. Remember to return to turn-on partitioning.

MORE EXACT FACTORIALS IN FAST MODE (cont)

Program Listing for 610!

000	91	R/S	040	95	=	080	00	00	120	03	03	160	00	0	200	11	A
001	09	9	041	74	SM*	081	73	RC*	121	22	INV	161	00	0	201	75	-
002	09	9	042	00	00	082	00	00	122	86	STF	162	00	0	202	02	2
003	42	STO	043	25	CLR	083	67	EQ	123	07	07	163	00	0	203	95	=
004	00	00	044	82	HIR	084	00	00	124	61	GTO	164	00	0	204	82	HIR
005	73	RC*	045	12	12	085	90	90	125	00	00	165	00	0	205	05	05
006	00	00	046	52	EE	086	97	DSZ	126	01	01	166	00	0	206	09	9
007	67	EQ	047	07	7	087	00	00	127	01	1	167	00	0	207	69	OP
008	00	00	048	94	+/-	088	00	00	128	82	HIR	168	00	0	208	17	17
009	54	54	049	85	+	089	81	81	129	34	34	169	00	0	209	00	0
010	22	INV	050	97	DSZ	090	69	OP	130	00	0	170	00	0	210	82	HIR
011	59	INT	051	00	00	091	20	20	131	65	×	171	00	0	211	04	04
012	65	×	052	00	00	092	43	RCL	132	82	HIR	172	00	0	212	01	1
013	82	HIR	053	05	05	093	00	00	133	18	18	173	00	0	213	52	EE
014	13	13	054	95	=	094	32	X:T	134	64	PD*	174	00	0	214	06	6
015	75	-	055	72	ST*	095	82	HIR	135	00	00	175	00	0	215	82	HIR
016	59	INT	056	00	00	096	14	14	136	69	OP	176	00	0	216	08	08
017	65	×	057	43	RCL	097	66	PAU	137	19	19	177	00	0	217	01	1
018	95	=	058	99	99	098	81	RST	138	87	IFF	178	00	0	218	52	EE
019	63	EX*	059	65	×	099	22	INV	139	07	07	179	00	0	219	07	7
020	00	00	060	82	HIR	100	58	FIX	140	00	00	180	00	0	220	94	+/-
021	59	INT	061	18	18	101	01	1	141	68	68	181	00	0	221	32	X:T
022	65	×	062	95	=	102	00	0	142	85	+	182	00	0	222	01	1
023	82	HIR	063	22	INV	103	69	OP	143	73	RC*	183	00	0	223	02	2
024	13	13	064	59	INT	104	17	17	144	00	00	184	00	0	224	02	2
025	85	+	065	67	EQ	105	47	CMS	145	59	INT	185	00	0	225	02	2
026	82	HIR	066	01	01	106	25	CLR	146	55	÷	186	00	0	226	52	EE
027	12	12	067	27	27	107	32	X:T	147	01	1	187	00	0	227	01	1
028	95	=	068	25	CLR	108	42	STO	148	52	EE	188	00	0	228	02	2
029	65	×	069	82	HIR	109	99	99	149	07	7	189	00	0	229	94	+/-
030	82	HIR	070	15	15	110	82	HIR	150	95	=	190	00	0	230	85	+
031	18	18	071	75	-	111	15	15	151	63	EX*	191	00	0	231	01	1
032	35	1/X	072	03	3	112	85	+	152	00	00	192	00	0	232	95	=
033	75	-	073	95	=	113	01	1	153	22	INV	193	00	0	233	22	INV
034	59	INT	074	77	GE	114	85	+	154	59	INT	194	00	0	234	52	EE
035	65	×	075	01	01	115	82	HIR	155	69	OP	195	00	0	235	58	FIX
036	95	=	076	12	12	116	05	05	156	20	20	196	00	0	236	00	00
037	65	×	077	09	9	117	33	X²	157	61	GTO	197	00	0	237	60	DEG
038	82	HIR	078	09	9	118	95	=	158	01	01	198	00	0	238	86	STF
039	18	18	079	42	STO	119	82	HIR	159	31	31	199	76	LBL	239	40	IND

Execution Times: The time to complete 461! is about 4 hours 12 minutes; for 610! the execution time is about 7 hours 29 minutes.

MODULE SELECTOR IDIOSYNCRASIES - V8N4P8 reported that Joseph Thomas had experienced some unexpected results when using the module selector (V5N8P3 and V7N1/2P25). Joseph reports the results of additional experiments: "I began to wonder what would happen to the module selector, that is, what module port it would access, when the TI-58 or TI-59 are run in the TRACE mode znd one of the program steps is Σ+ (key code 78, alphanumeric print character code 77). It seems that, at this point no longer surprising, that the module selection process is initiated and port 0 is accessed. Port 0 is a reasonable guess since a blank space in front of the module select code accesses port 0. This is another example where the module selection process can be inadvertently initiated. So far, it seems that whether the symbol Σ is sent to the (printer) via a print code or as a key code in the TRACE mode, then the module selection process will be initiated."

Don Graham reports similar effects. "... if a Σ+ is keyed by hand, or encountered in a program, while the printer is in TRACE mode, the AMS switches to Port 0, regardless of where it started. Also, despite what the instruction manual says, the printer does not have to be turned on for the AMS to work, although the calculator has to be mounted on the printer."

PROGRAMMING TRICKS - Don Graham writes: "Recently, while writing a program for a special application, I was faced with a situation in which a register had to be incremented by a different amount in the first pass through a loop as compared with subsequent passes. While this is not a particularly difficult problem to solve, I had a crowded program that left me very little elbow room, which meant minimizing the number of program steps to an extent that at first I did not believe possible. ..." An example of Don's method appears in the left hand listings below. The program stores a zero in data register R42, increments R42 by 40 on the first pass, and increments R42 by 5 on subsequent passes. The listing from 000 through 022 is the program as keyed in. The listing from 010 through 022 can be obtained with the original program in place, but with the listing starting at location 010. This is the code which will be executed by the calculator after the GTO 010 at steps 020 through 022. Old-timers will recognize the technique as jumping into the middle of a command to obtain a different result (V5N6P10).

The right hand listing below provides an even shorter routine which will accomplish the same result. The technique used was discussed under the heading "Neutralization" in V7N4/5P17 and attributed to Markus S. Markusson. The LBL 5 sequence at locations 007/008 causes the calculator to skip over the 5 on the first pass and sum 40 into R42. The GTO 008 at program locations 015 through 017 directs the calculator to jump into the middle of the LBL 5 sequence and read the 5 for subsequent passes. A variation of this technique is used in Dejan Ristanovic's "Minefield 2" program in this issue. See program locations 009/010 on page 8.

```
000  76 LBL
001  15  E
002  00  0
003  42 STO
004  42  42
005  04  4
006  00  0
007  44 SUM
008  42  42
009  43 RCL
010  43  43
011  42 STO
012  41  41
013  43 RCL
014  42  42
015  99 PRT
016  91 R/S
017  05  5
018  44 SUM
019  42  42
020  61 GTO
021  00  00
022  10  10
```

```
010  43 RCL
011  42  42
012  41 SST
013  43 RCL
014  42  42
015  99 PRT
016  91 R/S
017  05  5
018  44 SUM
019  42  42
020  61 GTO
021  00  00
022  10  10
```

```
000  76 LBL
001  15  E
002  00  0
003  42 STO
004  42  42
005  04  4
006  00  0
007  76 LBL
008  05  5
009  44 SUM
010  42  42
011  43 RCL
012  42  42
013  99 PRT
014  91 R/S
015  61 GTO
016  00  00
017  08  08
```

PC-100C AVAILABILITY - Several members have inquired as to sources for new or used equipment now that TI is no longer manufacturing the TI-59. In response to my question ELEK-TEK, Inc., 6557 North Lincoln Avenue, Chicago Illinois 60645 stated "The TI-PC-100C printers are still available at $85.00 each. We are sorry that we have no more TI-58C's or TI-59's." I suggest that you call (800)-621-1269 for confirmation of availability before ordering.

<u>MINEFIELD 2</u> - Dejan Ristanovic. This TI-59 game was inspired by John Ionidis' HP41C program of the same title which is available from the HP Users Library in Geneva. This program allows more options and is easier to use.

You are on square (0,0) of a 10 square by 10 square minefield. Your task is to reach the (9,9) square, in one piece. There are a few mines and rocks on the field. If you touch a mine you travel heavenward (hellward?). If you run into a rock, nothing happens and you are returned to the square you were on before your last move.

```
POS.  00  0 MINES
POS.  01  0 MINES
POS.  02  0 MINES
POS.  03  1 MINES
POS.  13  2 MINES
POS.  12  0 MINES
POS.  22  1 MINES
POS.  21  0 MINES
POS.  31  0 MINES
POS.  41  0 MINES
POS.  42  1 MINES
POS.  52       ROCK
POS.  43  1 MINES
POS.  53  1 MINES
POS.  54  1 MINES
POS.  64  1 MINES
POS.  65  0 MINES
POS.  75  0 MINES
POS.  85  1 MINES
POS.  86       ROCK
POS.  95  1 MINES
POS.  96      SHLD.
POS.  97       ROCK
POS.  85  1 MINES
POS.  86       ROCK
POS.  75  0 MINES
POS.  76  0 MINES
POS.  77  0 MINES
POS.  78       ROCK
POS.  87
POS.  88       ROCK
POS.  77
POS.  67
POS.  68
POS.  69
POS.  79
POS.  89
      38.      SCR.
```

In each move you control the direction with the user defined keys. You can go north (C), south (D), east (B), or west (A). You cannot move diagonally. You have a detector that will tell you how many mines there are on the squares surrounding your present position, a total of eight including those in a diagonal direction. The energy supply of your detector is limited and after a certain number of moves it won't work any more. You also have special shields that can protect you. You may use the shields only once during a game. If you use the shields you can cross a square with a mine or a rock without any consequence. There is only one problem with shields. Your detector will not give any information for the move with the shields in place.

To start the game enter any seed and press E'. The sample printout at the right used a starting seed of pi. After about ten minutes (that's correct, it takes that long to set up a game) everything will be ready and the printout will be POS. YX M MINES. That means that you are on the YX square and there are M mines around you. As you move across the mine field the printout can change to: ROCK (there is a rock on the square you tried to go to; SHLD. (you are using shields for this move); BOOM (you hit a mine and lose the game); or nothing (your detector is disabled for lack of energy). Before pressing one of the user defined keys that determines the direction of your move you select your options by pressing E if you want to disable the detector (if you have determined that you don't need information on how many mines there are on the surrounding squares and want to conserve energy), or by pressing SBR SBR to use the shields. When, and if, you reach square (9,9) you win and your score in terms of the number of moves requires is printed.

The Nop at step 459 of the program is left there as an aid when you are learning the game. You can continue the game, even though you have hit a mine. To remove that feature simply change step 459 to Cms. The program can run without a printer. After each move the number of mines surrounding you is displayed. If you see a 9 there is a rock on that square. If you see a flashing zero you hit a mine and lose the game. If you see some large number, you have won. Record your score and try another game.

MINEFIELD 2 (cont)

Program Listing:

```
000 76 LBL
001 17 B'
002 85 +
003 32 X:T
004 08 8
005 77 GE
006 00 00
007 10 10
008 03 3
009 76 LBL
010 01 1
011 95 =
012 92 RTN
013 76 LBL
014 19 D'
015 82 HIR
016 08 08
017 71 SBR
018 00 00
019 48 48
020 42 STO
021 03 03
022 71 SBR
023 00 00
024 48 48
025 42 STO
026 04 04
027 85 +
028 43 RCL
029 03 03
030 95 =
031 29 CP
032 67 EQ
033 00 00
034 17 17
035 08 8
036 32 X:T
037 71 SBR
038 00 00
039 58 58
040 77 GE
041 00 00
042 17 17
043 82 HIR
044 18 18
045 61 GTO
046 00 00
047 97 97
048 36 PGM
049 15 15
050 71 SBR
051 88 DMS
052 65 ×
053 01 1
054 00 0
055 95 =
056 59 INT
057 92 RTN
058 01 1
059 00 0
060 44 SUM
061 03 03
062 69 OP
063 24 24
064 43 RCL
065 04 04
066 94 +/-
067 22 INV
068 28 LOG
069 52 EE
070 65 ×
071 73 RC*
072 03 03
073 95 =
074 82 HIR
075 03 03
076 22 INV
077 59 INT
078 82 HIR
079 53 53
080 65 ×
081 01 1
082 00 0
083 22 INV
084 44 SUM
085 03 03
086 95 =
087 82 HIR
088 02 02
089 59 INT
090 82 HIR
091 52 52
092 22 INV
093 52 EE
094 69 OP
095 34 34
096 92 RTN
097 69 OP
098 24 24
099 85 +
100 82 HIR
101 12 12
102 95 =
103 55 ÷
104 01 1
105 00 0
106 44 SUM
107 03 03
108 85 +
109 82 HIR
110 13 13
111 95 =
112 65 ×
113 43 RCL
114 04 04
115 22 INV
116 28 LOG
117 52 EE
118 95 =
119 72 ST*
120 03 03
121 01 1
122 00 0
123 22 INV
124 44 SUM
125 03 03
126 61 GTO
127 00 00
128 92 92
129 76 LBL
130 11 A
131 69 OP
132 34 34
133 61 GTO
134 95 =
135 76 LBL
136 12 B
137 69 OP
138 24 24
139 61 GTO
140 95 =
141 76 LBL
142 13 C
143 69 OP
144 33 33
145 61 GTO
146 95 =
147 76 LBL
148 14 D
149 69 OP
150 23 23
151 61 GTO
152 95 =
153 76 LBL
154 15 E
155 86 STF
156 01 01
157 92 RTN
158 76 LBL
159 10 E'
160 47 CMS
161 42 STO
162 09 09
163 01 1
164 00 0
165 42 STO
166 06 06
167 02 2
168 04 4
169 42 STO
170 08 08
171 09 9
172 42 STO
173 00 00
174 08 8
175 19 D'
176 97 DSZ
177 00 00
178 01 01
179 74 74
180 09 9
181 19 D'
182 69 OP
183 33 33
184 02 2
185 44 SUM
186 04 04
187 03 3
188 42 STO
189 05 05
190 03 3
191 94 +/-
192 44 SUM
193 04 04
194 03 3
195 42 STO
196 00 00
197 22 INV
198 86 STF
199 00 00
200 43 RCL
201 03 03
202 71 SBR
203 03 03
204 88 88
205 43 RCL
206 04 04
207 71 SBR
208 03 03
209 88 88
210 87 IFF
211 00 00
212 02 02
213 28 28
214 08 8
215 32 X:T
216 71 SBR
217 00 00
218 58 58
219 77 GE
220 02 02
221 28 28
222 85 +
223 01 1
224 95 =
225 71 SBR
226 00 00
227 97 97
228 69 OP
229 24 24
230 97 DSZ
231 00 00
232 01 01
233 97 97
234 69 OP
235 23 23
236 97 DSZ
237 05 05
238 01 01
239 90 90
240 97 DSZ
241 06 06
242 01 01
243 80 80
244 25 CLR
245 42 STO
246 03 03
247 42 STO
248 04 04
249 42 STO
250 08 08
251 82 HIR
252 04 04
253 02 2
254 04 4
255 42 STO
256 07 07
257 76 LBL
258 95 =
259 69 OP
260 28 28
261 69 OP
262 00 00
263 01 1
264 08 8
265 32 X:T
266 43 RCL
267 03 03
268 85 +
269 43 RCL
270 04 04
271 95 =
272 67 EQ
273 04 04
274 42 42
275 22 INV
276 86 STF
277 00 00
278 43 RCL
279 03 03
280 71 SBR
281 03 03
282 88 88
283 43 RCL
284 04 04
285 71 SBR
286 03 03
287 88 88
288 87 IFF
289 00 00
290 04 04
291 24 24
292 03 3
293 03 3
294 03 3
295 02 2
296 03 3
297 06 6
298 04 4
299 00 0
300 69 OP
301 01 01
302 43 RCL
303 03 03
304 17 B'
305 82 HIR
306 06 06
307 43 RCL
308 04 04
309 17 B'
310 85 +
311 82 HIR
312 16 16
313 65 ×
314 01 1
315 00 0
316 00 0
317 95 =
318 69 OP
319 02 02
320 01 1
321 32 X:T
322 82 HIR
323 14 14
324 67 EQ
325 04 04
326 57 57
327 71 SBR
328 00 00
329 58 58
330 82 HIR
331 02 02
332 32 X:T
333 09 9
334 67 EQ
335 04 04
336 24 24
337 08 8
338 67 EQ
339 04 04
340 02 02
341 43 RCL
342 03 03
343 42 STO
344 01 01
345 43 RCL
346 04 04
347 42 STO
348 02 02
349 25 CLR
350 87 IFF
351 01 01
352 03 03
353 82 82
354 22 INV
355 97 DSZ
356 07 07
357 03 03
358 82 82
359 32 X:T
360 17 B'
361 65 ×
362 01 1
363 00 0
364 00 0
365 95 =
366 69 OP
367 03 03
368 03 3
369 00 0
370 02 2
371 04 4
372 03 3
373 01 1
374 01 1
375 07 7
376 03 3
377 06 6
378 69 OP
379 04 04
380 82 HIR
381 12 12
382 29 CP
383 69 OP
384 05 05
385 61 GTO
386 04 04
387 76 76
388 65 ×
389 53 (
390 40 IND
391 75 -
392 09 9
393 95 =
394 32 X:T
395 00 0
396 77 GE
397 04 04
398 01 01
399 86 STF
400 00 00
401 92 RTN
402 43 RCL
403 01 01
404 42 STO
405 03 03
406 43 RCL
407 02 02
408 42 STO
409 04 04
410 03 3
411 05 5
412 03 3
413 02 2
414 01 1
415 05 5
416 02 2
417 06 6
418 69 OP
419 04 04
420 69 OP
421 05 05
422 09 9
423 92 RTN
424 01 1
425 04 4
426 03 3
427 02 2
428 03 3
429 02 2
430 03 3
431 00 0
432 07 7
433 03 3
434 69 OP
435 04 04
436 69 OP
437 05 05
438 25 CLR
439 68 NOP
440 61 GTO
441 61 GTO
442 03 3
443 06 6
444 01 1
445 05 5
446 03 3
447 05 5
448 04 4
449 00 0
450 69 OP
451 04 04
452 43 RCL
453 08 08
454 69 OP
455 06 06
456 91 R/S
457 03 3
458 06 6
459 02 2
460 03 3
461 02 2
462 07 7
463 01 1
464 06 6
465 04 4
466 00 0
467 69 OP
468 04 04
469 69 OP
470 05 05
471 76 LBL
472 71 SBR
473 01 1
474 82 HIR
475 34 34
476 22 INV
477 86 STF
478 01 01
479 92 RTN
```

BA-55 PROFESSIONAL BUSINESS ANALYST - The BA-55 is one of the new calculators which will work with the PC-200 printer. The TI-66 will also be capable of operation with the PC-200 (V8N3P12). The BA-55 is available now. I purchased mine from Elek-Tek.

The BA-55 has the same type of chassis as the TI-55-II and the TI-57-LCD, including the tilted LCD display. The calculator comes with many pre-programmed business functions such as interest, statistics, linear regression, internal rate of return, annuities, etc. The statistics routine has the same quirks as the TI-55-II (V8N1P26). A constant memory feature retains stored data when the calculator is turned off.

Up to forty programming steps are available depending on the partitioning. Program codes are not merged resulting in inefficient use of program steps.

No decision commands are available, but one can be devised by performing a 1/X command on the contents recalled from a data register. If the data register contained a zero then an error condition occurs and the calculator stops. If the data register contained a number other than zero the calculations continue (This technique was described in the review of the Snover and Spikell book on Programming the TI-55; see V7N6P12). When used in a program the RST command resets the program counter and does not halt program execution. This permits programming of a loop with exit from the loop provided by the decision test described earlier. (In the TI-55-II a RST command in a program causes a return to program location 00, but program execution is halted as well.) The printer commands are Advance, List, Print and Trace -- reminiscent of the SR-52.

The sample program listed at the right illustrates the use of a loop using the RST function combined with the decision function using the 1/X command. The program is used to investigate the consistency of the square root - square sequence operating on input integers (see page 2 of this issue for a discussion of that problem with other calculators). To run the program store zero in data register 1 and press RST R/S. The program will run for a short period and stop with "Error" in the display, indicating that a division by zero has occurred. Press On/C and then RCL 1 to see a 1, the first integer that was tested. Press R/S to continue. The calculator will run for a few seconds and again stop with "Error" in the display. Again press On/C and RCL 1 and see a 4 in the display. If you continue testing in this way you will find that the input integers for which the x x sequence returns the exact input are 1, 4, 11, 30, 100, 121, 400, 484,etc. This is even more erratic than the HP-11 or HP-41.

Step	Code	Key
00	01	1
01	81	SUM
02	01	01
03	71	RCL
04	01	01
05	53	x
06	58	x
07	75	-
08	71	RCL
09	01	01
10	95	=
11	54	1/X
12	37	RST

The same test run with the older MBA programmable calculator also yields erratic results. For that device the integers for which the x x sequence returns the exact integer are 1, 4, 5, 9 through 25, 27 through 29, 31, 33, 40 through 44, 46, 47, 49, etc., where the frequency of successful tests more closely approaches that of the HP calculators. It seems that, for some unknown reason, the TI business calculators are less consistent than the TI scientific calculators, at least for this particular test!

EVEN MORE ON FINDING PI IN BASIC - V8N3P25/26 and V8N4P14 reported various results in response to the problem of finding pi in BASIC using the formula 4*ATN(1). George Thomson has recognized the editor's fascination with that subject by addressing recent letters to "Palmer Pi Hanson". But George also sent a mnemonic for the first thirty places of pi from an old notebook. Deena Koniver gave it in a letter to the editor on page 309 of the April 1969 issue of Datamation, saying it came from a friend who "learned it from one of his college professors." Count the letters in each word to obtain each successive digit:

Now I, even I, would celebrate in rhymes inept,
The great immortal Syracusan, rivaled nevermore,
Who in his wondrous lore passed on before,
Left men his guidance how to circles mensurate.

If a TI-58/58C/59 user presses 2nd-pi the calculator responds with 3.141592654 in the display. Experienced users know that the guard digits contain additional information. For the value of pi the guard digits can be viewed by the sequence

pi - 3.14 = x 100 =

which will yield a display of 0.159265359. This indicates that the calculator evaluates pi as 3.14159265359. Only twelve digits are returned since the thirteenth digit is a zero. Different methods of stripping off the leading digits will not change the result. For example, the sequence

pi - 3.14159 = x 100000 =

will yield a display of 0.265359 , again indicating that the calculator evaluates pi as 3.14159265359 . Users of the TI product line have come to expect this sort of predictable behavior from their calculators. Among personal computers the TI CC-40, the TI 99/4A and the Radio Shack Model 100 all exhibit this same sort of bahavior; but some of the personal computers exhibit rather unexpected behavior. Consider the responses of the BASIC provided with the Color Computer and with the Apple II Plus to various sequences:

Sequence	Color Computer	Apple II Plus
4*ATN(1)	3.1415 9266	3.1415 9266
4*ATN(1)-3.	.1415 92653	.1415 92653
4*ATN(1)-3.1	.0415 92653 9	.0415 92653 6
4*ATN(1)-3.14	.0015 92653 8	.0015 92653 8
4*ATN(1)-3.141	.0005 92652 708	.0005 92653 174
4*ATN(1)-3.1415	.0000 92653 5577	.0000 92653 5577
4*ATN(1)-3.14159	.0000 02652 40669	.0000 02652 87235
4*ATN(1)-3.141592	.0000 00653 78844 8	.0000 00653 78844 8
4*ATN(1)-3.1415926	.0000 00053 08538 67	.0000 00053 31821 74
4*ATN(1)-3.14159265	.0000 00003 72529 03	.0000 00004 19095 159
4*ATN(1)-3.141592653	0	0
AMS-55 Reference	3.1415 92653 58979	3.1415 92653 58979

Both computers generate unpredictable "garbage" in response to the stripping procedures. Readers are invited to report the responses of other personal computers.

<u>NOTES ON CARD READING</u> - I had always been puzzled by reports of problems with the magnetic card mechanism of the TI-59. For example, V6N3P10 warned that the use of graphite (lead) pencils to mark the cards would ruin the mechanism in less than a few months. For years I had used a fine lead pencil to mark my cards in accordance with page VII-8 of Personal Programming. My experience had been so good that I simply assumed that cards could be written and read without problems. That kind of reliable performance was a key factor in the development of the "load-and-go" technique which made the Pgm-02-SBR-239-9 method of fast mode entry a reasonable procedure. In V7N6P10 I even speculated that the source of the "hundreds upon hundreds of misreads of my cards" (see "Mailbag", V6N6/7P2) might be the operation of the card reader with a defective battery, rather than a defective card mechanism. But, last winter I began to encounter reader problems with my TI-59. In April of this year George Thomson also encountered magnetic card problems. Excerpts from our correspondence about our mutual problems follow:

GWT to POH , 11 April 1983 - "My TI-59 is sick. Everything OK but card writing. Seems to write OK but when I try to load another day, I usually get a flashing zero. Despite this some of the commands may be in correctly..."

POH to GWT, 17 April 1983 - " ... My TI-59 also developed card reader problems this winter. I found some interesting things while trying to get it to read. First, it was decidedly sensitive to temperature. Second, there was some sort of magic which can be done with the cards. I have reached the conclusion that the cards, tracks on the cards, or something else might be dirty. You have to stay with me in the following account--it offends reason, but I swear it is true. Anyway, I used various solvents to wipe the cards clean, ranging from water, through soapy water, to alcohol. Eventually, I found that the best cleaner was good old fashioned saliva! I don't understand it, but it worked. I could take a card that wouldn't read, wet my handkerchief with a little saliva, and wipe the card, preferably on the back side. Eureka! The card would read. Another card which had not been able to be read before, and which had not been wiped, would continue not to read. ... But the cleaning would not yield a permanent reading capability. If I came back a few hours later, I would have to redo the wiping procedure, and I needed to wipe each card I wanted to use. It all seems like a little witchcraft, and I don't pretend to understand it..."

GWT to POH, 27 April 1983 - "... So help me, I tried Hanson's Universal Solvent ... alias, spit ... and at least one card got written and read. Hope springs eternal..."

GWT to POH, 24 June 1983 - "Now that I apply Hanson's Universal Solvent before loading data on a magnetic card I have had no more troubles."

Recently, while going through some of the residual material I received from Maurice Swinnen I came on a description of the CCL144 Cleaning Strip, including a sample. I tried it in my TI-59 and was immediately rewarded with trouble-free magnetic card operation--no more need for spit! The strips are available in boxes of ten strips for twelve dollars from CMPI, Inc., 7200 Jersey Avenue North, Minneapolis, Minnesota 55428 (telephone 612-566-1848). The box of strips is also available under TI part number 1105782-1. For the benefit of US club members only I will provide sample single strips for $2.00 each including packing and postage. No checks, please.

AMICABLE NUMBERS - Bob Fruit. Don't believe what you hear other people say about me, I AM NOT CRAZY! It is a sign of an intelligent, inquisitive mind to set a high goal and pursue it, and pursue it, ...

Let me tell you my story. Dejan Ristanovic's article on Perfect Numbers (V6N2P2) finally got me going on a project in Sociable Numbers. That is something that I have been meaning to do for a long time. Perfect and Amicable numbers are the first two members of the class of Sociable Numbers.

Perfect Numbers, the first member of the Sociable Number Class, was defined in Dejan's article. Amicable Numbers, the second member of the Sociable Number class, are pairs of numbers with the following property: the sum of the proper divisors of the first number equals the second number; and, the sum of the proper divisors of the second number equals the first number. 220 and 284 are the lowest pair of Amicable Numbers. The sum of the divisors of 220 is 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284. The sum of the proper divisors of 284 is 1 + 2 + 4 + 71 + 142 = 220. There are algorithms for finding amicable pairs; however, they only find a subset of all amicable numbers. That is, there are amicable pairs that are not found by using a known algorithm. Not only that, not all of the same amicable pairs are found by the different algorithms. I wanted to find all amicable pairs for the range searched. That meant that every number would need to be tested. That is a much longer way to search for amicable numbers, but it is the only way to be certain that every amicable pair has been found.

I reviewed the speedy factor finders published in TI PPC Notes to come up with the fastest way to find all the prime divisors of a number. There is a formula for a number N where

$$S = [(P_1^{n_1+1} - 1)/(P_1 - 1)*(P_2^{n_2+1} - 1)/(P_2 - 1)* \ldots\ldots]$$

is equal to the sum of all divisors of N, including N itself, where $P_1^{n_1}$, $P_2^{n_2}$ ... are the prime factors of N. An elementary book on number theory is a good place to review this formula. Using this formula the sum of the proper divisors of N is (S - N). I also took advantage of the fast mode feature of the TI-59 to speed up the whole process.

This is an easy program to run. Save the program on two sides pf a card. Load side 1 and press A to enter fast mode. Reload card side 1 and see the dim [at the left side of the display. Load card side 2. The program now takes over control of the calculator and the printer. There is nothing else for you to do, ever again, with your calculator. The program is in an infinite loop and can only be halted by turning the calculator off. Remember that in fast mode the R/S and RST keys are disabled. It is cheating to pay a neighbor to come over and accidentally (on purpose) pull out the plug.

Any Amicable pairs found are printed. The display flashes the next number to be tested. You can visit your calculator every day (or less often if you want, the calculator won't mind), watch the display for a couple of minutes (or hours, if you are so inclined) and see the numbers that are being tested. If you live right it may even print an amicable pair while you are visiting it. Isn't this an exciting relationship to have with your calculator. Just think what you can now say to all those people who used to ask what use you were getting from that expensive calculator. Instead of a sheepish nothing, you can proudly tell them about this program.

AMICABLE NUMBERS (cont)

Editor's Note: Bob let his calculator run for over 2662 hours (nearly 111 days). Is that the longest continuous run of a TI-59? A list of the amicable pairs he found appears at the right where the highest pair is 308620 and 389924. Note that Bob's program does not print the leading zeroes in each quarter of the line.

220	284
1184	1210
2620	2924
5020	5564
6232	6368
10744	10856
12285	14595
17296	18416
63020	76084
66928	66992
67095	71145
69615	87633
79750	88730
1 485	124155
122265	139815
122368	123152
141664	153176
142310	168730
171856	176336
176272	180848
185368	2 3432
196724	2 2444
280540	365084
3 8620	389924

Fast mode entry is obtained with Martin Neef's Pgm-02-SBR-239-9 sequence which has generally been superseded by the Stflg Ind technique.

Program Listing:

000	76	LBL	064	43	RCL	128	55	÷	192	06	06	256	61	GTO	320	62	62
001	11	A	065	08	08	129	05	5	193	61	GTO	257	01	01	321	32	X:T
002	00	0	066	22	INV	130	54	)	194	02	02	258	96	96	322	42	STO
003	00	0	067	86	STF	131	65	×	195	01	01	259	02	2	323	01	01
004	00	0	068	01	01	132	01	1	196	82	HIR	260	61	GTO	324	01	1
005	36	PGM	069	61	GTO	133	00	0	197	36	36	261	01	01	325	67	EQ
006	02	02	070	00	00	134	00	0	198	04	4	262	96	96	326	03	03
007	71	SBR	071	85	85	135	49	PRD	199	44	SUM	263	06	6	327	47	47
008	02	02	072	69	OP	136	01	01	200	03	03	264	61	GTO	328	32	X:T
009	39	39	073	03	03	137	34	√X	201	82	HIR	265	01	01	329	65	×
010	09	9	074	43	RCL	138	75	-	202	15	15	266	96	96	330	43	RCL
011	00	0	075	03	03	139	59	INT	203	55	÷	267	82	HIR	331	06	06
012	22	INV	076	69	OP	140	44	SUM	204	82	HIR	268	15	15	332	97	DSZ
013	58	FIX	077	04	04	141	01	01	205	16	16	269	34	√X	333	05	05
014	02	2	078	69	OP	142	97	DSZ	206	95	=	270	75	-	334	03	03
015	22	INV	079	05	05	143	00	00	207	22	INV	271	02	2	335	29	29
016	96	WRT	080	61	GTO	144	01	01	208	59	INT	272	03	3	336	75	-
017	66	PAU	081	00	00	145	23	23	209	22	INV	273	05	5	337	01	1
018	91	R/S	082	21	21	146	25	CLR	210	67	EQ	274	42	STO	338	95	=
019	42	STO	083	68	NOP	147	48	EXC	211	40	IND	275	03	03	339	69	OP
020	07	07	084	68	NOP	148	01	01	212	03	03	276	82	HIR	340	36	36
021	07	7	085	55	÷	149	48	EXC	213	82	HIR	277	16	16	341	55	÷
022	42	STO	086	01	1	150	03	03	214	15	15	278	95	=	342	43	RCL
023	02	02	087	52	EE	151	22	INV	215	32	X:T	279	77	GE	343	06	06
024	22	INV	088	05	5	152	87	IFF	216	82	HIR	280	02	02	344	95	=
025	86	STF	089	75	-	153	02	02	217	16	16	281	35	35	345	49	PRD
026	00	00	090	22	INV	154	01	01	218	82	HIR	282	22	INV	346	04	04
027	61	GTO	091	59	INT	155	09	09	219	65	65	283	86	STF	347	01	1
028	01	01	092	42	STO	156	22	INV	220	61	GTO	284	03	03	348	42	STO
029	70	70	093	03	03	157	86	STF	221	03	03	285	82	HIR	349	05	05
030	32	X:T	094	95	=	158	02	02	222	15	15	286	15	15	350	43	RCL
031	01	1	095	32	X:T	159	87	IFF	223	01	1	287	61	GTO	351	01	01
032	67	EQ	096	01	1	160	01	01	224	61	GTO	288	03	03	352	42	STO
033	00	00	097	52	EE	161	00	00	225	01	01	289	15	15	353	06	06
034	40	40	098	05	5	162	56	56	226	96	96	290	82	HIR	354	29	CP
035	22	INV	099	49	PRD	163	22	INV	227	02	2	291	15	15	355	87	IFF
036	87	IFF	100	03	03	164	52	EE	228	61	GTO	292	82	HIR	356	03	03
037	00	00	101	25	CLR	165	61	GTO	229	01	01	293	65	65	357	02	02
038	03	03	102	42	STO	166	00	00	230	96	96	294	43	RCL	358	01	01
039	70	70	103	01	01	167	72	72	231	02	2	295	06	06	359	61	GTO
040	43	RCL	104	67	EQ	168	68	NOP	232	61	GTO	296	32	X:T	360	02	02
041	07	07	105	01	01	169	68	NOP	233	01	01	297	01	1	361	90	90
042	69	OP	106	49	49	170	73	RC*	234	96	96	298	22	INV	362	69	OP
043	27	27	107	32	X:T	171	02	02	235	04	4	299	67	EQ	363	25	25
044	32	X:T	108	22	INV	172	66	PAU	236	61	GTO	300	02	02	364	61	GTO
045	43	RCL	109	86	STF	173	29	CP	237	01	01	301	85	85	365	03	03
046	09	09	110	02	02	174	82	HIR	238	96	96	302	43	RCL	366	54	54
047	22	INV	111	52	EE	175	05	05	239	02	2	303	04	04	367	68	NOP
048	67	EQ	112	55	÷	176	01	1	240	61	GTO	304	75	-	368	68	NOP
049	00	00	113	52	EE	177	42	STO	241	01	01	305	73	RC*	369	68	NOP
050	21	21	114	00	0	178	04	04	242	96	96	306	02	02	370	43	RCL
051	86	STF	115	01	1	179	42	STO	243	04	4	307	69	OP	371	08	08
052	01	01	116	94	+/-	180	05	05	244	61	GTO	308	22	22	372	32	X:T
053	61	GTO	117	42	STO	181	42	STO	245	01	01	309	95	=	373	43	RCL
054	00	00	118	00	00	182	06	06	246	96	96	310	72	ST*	374	07	07
055	85	85	119	95	=	183	02	2	247	02	2	311	02	02	375	77	GE
056	69	OP	120	28	LOG	184	02	2	248	61	GTO	312	61	GTO	376	00	00
057	00	00	121	48	EXC	185	03	3	249	01	01	313	00	00	377	40	40
058	69	OP	122	00	00	186	42	STO	250	96	96	314	30	30	378	61	GTO
059	01	01	123	85	+	187	03	03	251	04	4	315	32	X:T	379	00	00
060	43	RCL	124	93	.	188	86	STF	252	61	GTO	316	43	RCL	380	25	25
061	03	03	125	01	1	189	03	03	253	01	01	317	06	06	381	00	0
062	69	OP	126	85	+	190	02	2	254	96	96	318	67	EQ	382	00	0
063	02	02	127	59	INT	191	82	HIR	255	06	6	319	03	03	383	00	0

<u>MATRIX OPERATIONS WITH THE CC-40 MATHEMATICS MODULE</u> - In V8N4P12/13 I reported that the execution speed of the prime factors program in the CC-40 Mathematics module was disappointing, and that the module had other deficiencies as well. I am happy to report that the matrix manipulation programs seem to be more carefully constructed. The capabilities are similar to those of the ML-02 and ML-03 programs in the Master Library module of the TI-59. In fact, the discussion of the use of the lower upper (LU) decomposition method is identical for the CC-40 Mathematics module and the TI-59 ML-02 programs.

Execution speed is substantially improved. The CC-40 finds the determinant for the third order matrix problem on page 12 of the manual for the TI-59 Master Library module in about two seconds, while the TI-59 requires sixteen seconds to complete the same problem. The CC-40 finds the determinant of a fifth order matrix in about six seconds, while the TI-59 requires about fifty-three seconds for the same problem using ML-02.

A deficiency of the CC-40 program is that the result is brought to the display with a BASIC Print command and the user cannot perform any chain calculations on the result without reentering the value. The reentering process necessarily drops any digits which were not displayed. The TI-59 solution is displayed in a manner such that chain calculations on the displayed result is possible. The loss in accuracy for the chain calculations with the CC-40 caused by the reentry can be duplicated with the TI-59 by performing EE-INV-EE to truncate to the displayed value before proceeding with user entered chain calculations. If the variable names of the solution were available the user could recall the solutions as a part of his keyboard BASIC chain calculations and retain the full accuracy; but the documentation with the CC-40 provides no information as to the variable names. To remedy this situation I have written a short demonstration program for solution of a system of linear equations (AX = B) which provides identification of variable names for at least some elements of the solution:

```
100 DIM A(8,9),C(8,8),B(8)
110 R = PI
120 INPUT "Enter Order of Matrix - ";N
150 CALL MI("A",A(,),1,N,N,0)
200 CALL AK("B",B(),1,N,0)
250 PRINT "Solving"
300 CALL MATS(A(,),C(,),B(),1,1,5,1,N,1,R)
350 IF R<>0 THEN 400
360 PRINT "MATRIX IS SINGULAR":PAUSE
400 FOR I = 1 TO N
410 X$ = "X" & STR$(I) & " = "
420 PRINT X$;A(I,1):PAUSE
430 NEXT I
999 STOP
```

Line 100 - The dimension statement sets up the array names to be used in the various subroutine calls. For reasons that are not very clear to me the array for the entry matrix A(m,n) must have one more column than the order of the problem if the MATS subroutine call at line 300 is to operate properly.

Line 110 - The dummy variable R will be used to indicate whether or not the input matrix is singular. See the discussion of the TEST variable on page 94 of the Mathematics Module manual.

Matrix Operations with the CC-40 Mathematics Module - (cont)

Line 120 - Provides operator control of the order of the problem to be solved.

Line 150 - This subroutine call provides for input and edit of the elements of the matrix A into a two dimensional array. See page 95 of the manual. The single line subroutine call provides a thorough set of prompts for entry and editing, including indication of the row and column on each element to be entered.

Line 200 - This subroutine call provides for input and edit of the elements of the vector B into a one dimensional array. See pages 85-86 of the manual. Again, the subroutine call also provides a thorough set of prompts..

Line 250 - This only provides a clear indication that the computer has changed from the edit mode to the solve mode.

Line 300 - This subroutine call provides the solution for the set of linear equations. See page 94 of the manual. The subroutine ends with the elements of the solution in the subscript 1 column of the A array, and with the inverse of the A matrix in array C. If the input A matrix was singular then R is changed to zero.

Line 350 - Tests the value of R to determine if the input matrix A was singular.

Line 360 - Displays an appropriate message if the input matrix was singular.

Lines 400 to 430 - Display the elements of the solution with appropriate annotation.

To illustrate use of the program use the problem on page 12 of the manual for the TI-59 Master library module:

1. Press RUN and ENTER. See the prompt "Enter Order of Matrix - ".

2. Press 3 and press ENTER. See the prompt "Enter A(1,1):"

3. Press 4 and press ENTER to insert the A(1,1) element. The computer accepts the input and returns with the prompt "Enter A(1,2):". Continue to enter the remaining elements of the matrix. Note that the CC-40 accepts the matrix elements by row in contrast with the TI-59 which accepted the elements by column. But note that there is nothing to remember since the MI subroutine call supplies the necessary prompts. When the last element A (3,3) has been entered the computer responds with the prompt "Edit?". If you choose to edit by responding with a Y the computer response is the prompt "Edit All Input?". If you respond with a N the computer response is "Enter Row To Be Edited:". You enter the row number and the computer response is another prompt "Enter Column to Be Edited:". You enter the column number and the computer response is "Enter A(i,j): Aij" where i and j are the row and column you selected, and Aij is the value which was entered for that element earlier. If you decide to edit that element you replace the displayed value with the desired one and press ENTER. If you decide not to change the element you simply press ENTER. In either case the computer responds with the prompt "Edit Other Elements?".

Matrix Operations with the CC-40 Mathematics Module - (cont)

4. When you have completed any editing of the A matrix the final N response to the edit prompts will cause the computer to move forward to the entry of the vector elements. The prompt message will be "Enter B(1)". You proceed to enter the elements of the vector in a manner similar to that used for the matrix. Again, you will be given an opportunity to edit. The important point is that all the prompts for the entry of both matrix elements and vector elements are provided by the module in response to the subroutine calls MI and AK.

5. When you have completed the editing process by responding with an N at the appropriate point the program immediately proceeds to solution of the problem, with the indication "Solving" in the display. When the solution is complete the computer response is the display "X1 = 4" if you entered the problem from page 12 of the Master Library correctly. Press ENTER as many times as needed to see the remainder of the solution.

6. After the display of the solution has been completed you may use keyboard BASIC (or you may add commands to the program) to read out other parameters, or the same parameters in other formats. The elements of the input matrix have been destroyed. The elements of the inverse of the input matrix appear in array C properly located; that is, the i,j element of the inverse can be recalled with the command PRINT C(i,j). For our example, the sequence PRINT C(2,2) will yield a ten digit display of .0416666667. The user can view additional digits with the command

```
PRINT USING ".##############";C(2,2)
```

to yield a fourteen digit display display of .04166666666667 ; or, in a technique similar to that used to observe the guard digits of the TI-59, the user can use the command

```
PRINT (C(2,2)-.04166)*100000
```

to yield a display of .666666667 .

7. If the user changes the sixth element in the argument for the MATS subroutine call from a 5 to a 4, then the program will only proceed through the calculation of the inverse of the input matrix. The elements of the inverse will appear in array C, again with the appropriate subscripts. The elements of the inverse will also appear in array A, but with the first and second columns interchanged. This is exactly the same orientation in which the inverse appears in a TI-59, where there is also an indication of the interchanged columns through observation of the pivoting index; that is, for the particular third order example used here TI-59 memory registers R17, R18 and R19 will contain the numbers 2, 1, and 3 respectively. I have been unable to find a way to recall the pivoting index from the CC-40 solution. Hopefully, this helps to explain the note in the discussion of "Inversion" on page 52 of the manual for the Mathematics module which states "..The inverse of A may be stored with its columns permuted and must be reentered for subsequent calculations." That statement is true if one uses the CALL "MAT" method to obtain the inversion. If one uses the CALL MATS method illustrated here then the columns in array A may (or may not) be permuted depending on the particular input matrix, but the inverse which appears in array C will not have permuted columns and can be used directly for further calculations.

A least squares polynomial curve fitting program using the techniques described here appears on the following page.

LEAST SQUARES POLYNOMIAL CURVE FIT WITH THE CC-40 MATHEMATICS MODULE

This program uses the same techniques described on the previous pages with the addition of a call of subroutine AU (see pages 87-88 of the manual) to provide entry of the data pairs into two one-dimensional arrays. Again, the subroutine call provides valuable prompts. I believe that the prompts with this program are sufficient such that no detailed program description is required. There is one idiosyncrasy of the prompts for editing the entry of the data pairs which is described on page 18 of this issue.

```
100 DIM A(8,9),B(8),C(8,8),H(8),X(50),Y(50)
110 INPUT "Number of Data Pairs? ";K
120 CALL AU("X","Y",X(),Y(),1,K,0)
130 INPUT "Degree of Polynomial? ";N
140 PRINT "Solving"
150 N=N+1:R=1:P$="":Q$=""
160 FOR I=1 TO N:FOR J=1 TO N
170 A(I,J)=0:NEXT J
180 B(I)=0:NEXT I
190 FOR L=1 TO K
200 H(1)=1
210 FOR I=2 TO N
220 H(I)=H(I-1)*X(L):NEXT I
230 FOR I=1 TO N:FOR J=1 TO N
240 A(I,J)=A(I,J)+H(I)*H(J):NEXT J
250 B(I)=B(I)+H(I)*Y(L):NEXT I
260 NEXT L
270 CALL MATS(A(,),C(,),B(),1,1,5,1,N,1,R)
280 IF R<>0 THEN 300
290 PRINT "Matrix is singular":PAUSE:GOTO 470
300 FOR I=1 TO N
310 X$="A"&STR$(I-1)&" = "
320 PRINT X$;A(I,1):PAUSE:NEXT I
330 INPUT "Display Residuals (Y/N)? ";P$
340 S1=0
350 FOR I =1 TO K
360 Y1=A(N,1)
370 FOR J=(N-1) TO 1 STEP -1
380 Y1=A(J,1)+X(I)*Y1:NEXT J
390 D1=Y(I)-Y1
400 IF P$="y" OR P$="Y" THEN 410 ELSE 430
410 A$="d"&STR$(I)&" = "
420 PRINT A$;D1:PAUSE
430 S1=S1+D1*D1:NEXT I
440 PRINT "Standard Error = ";SQR(S1/(K-N)):PAUSE
450 INPUT "Try a Different Degree (Y/N)? ";Q$
460 IF Q$="y" OR Q$="Y" THEN 130
470 STOP
```

LANGUAGES ON THE CC-40 - V8N4P12 discussed the various languages which are available with the CC-40 by using the CALL SETLANG command. The Mathematics and Statistics modules support English, German, and French. Tne Finance module supports only English and German.

A PROMPTING ANOMALY IN THE MATHEMATICS MODULE FOR THE CC-40

There is an apparent error in that portion of the Mathematics module for the CC-40 which provides for editing of the entry of two one-dimensional arrays. An example occurs when running the Cubic Splines program. Go to page 31 of the manual and follow the example through step 13. At that point the display will read "Edit?". Do not proceed to step 14. Rather respond with a Y for yes and press ENTER. The display will prompt with the message "Edit All Input?". This time respond with an N for no and press ENTER. The display will prompt with the message "Enter Element to Be Edited:". Press 3 and ENTER and see "Enter X(3): 1" in the display. The 1 was loaded into that location by step 10. Press ENTER again assuming that you did not want to edit the value in X(3). The display changes to "Enter X(3): .8413". You would have expected the display to read "Enter Y(3): .8413". Although the indication of which element is available to be edited is incorrect, the value displayed is that which was stored in Y(3) at step 11. There is no harm done by the improper indication, but it will surprise an unwary operator. The same effect can be seen when using the AU routine on page 87 in the manual. Users of the Least Squares Polynomial Curve Fit on page 17 of this issue can expect to encounter this anomaly.

FACTORIALS WITH THE CC-40 MATHEMATICS MODULE

Factorials can be calculated with the Mathematics module of the CC-40 by recognizing that N! = Gamma(N+1). With this technique the CC-40 with the Mathematics module installed will return Ln(Gamma(70)) = 226.1905483 in about one second and pressing ENTER will immediately yield Gamma(70) = 1.711225E+98 which is equal to 69!. By comparison the ML-16 program on the TI-59 takes about 16 seconds to obtain the equivalent answer; but, the MU-11 program in the Math/Utilities module for the TI-59 will find 69! in about four seconds with the Gamma function method. The CC-40 can obtain factorials up to 85! = 3.31424E+126 with this method.

MORE ON THE TI-66 - V8N3P12 provided an introduction to the TI-66. Dave Leising has received an advance copy of the user's manual. Dave has started a review of the functions. There is no magnetic card reader capability and no port for a Soid State module. New commands relative to the TI/58/59 include:

TRC - Trace. Control has been moved from printer to keyboard.

CSR - Clear statistics registers. Replaces TI-58/59's Pgm-01-SBR-CLR

PAR - Set Partition. Replaces TI-58/59's Op-17 function.

An appendix lists the items in TI-58/58C-59 programs which must be changed to run on the TI-66. An interesting note states:

> "There are no HIR commands or other hidden features on the TI-66 that you may have accessed on the TI-58/58C/59 through illegal key sequences."

That will be a disappointment. We trust that TI will not be offended if we disregard that notice and look for additonal capability. Dave has already started by generating a table of instructions versus key codes.

---

13 DIGIT MODULO 30 SFF REVISITED - G. L. Wilson of England. V8N4P15 presented a modification of a 13 digit modulo 30 Speedy Factor Finder to incorporate the Stflg Ind method of fast mode entry. Both the original program (PPX 398278) and the revised program used the test algorithm

RCL 01 - (CE DIV RCL 02)INT x RCL 02 =

In late August Mr. Wilson submitted a modification to the PPX 398278 program which used a much improved test algorithm. Each iteration of the test begins with the integer to be factored in the t register, and the preceding factor which was tested in data register R02. Then, the test sequence is

x⇌t DIV x⇌t n SUM 02 RCL 02 = INT x RCL 02 =

where n is the increment to be added to obtain the new test integer. A faster method of generating return addresses was also included. The result in an improvement in execution speed by about 10 per cent relative to the PPX 398278 program. A listing of the program appears on page 20. The listing shows program steps 223 through 239 as they appear before fast mode initialization.

The program submitted by Mr. Wilson used the same h12 method for fast mode entry which was used in PPX 398278. The initialization process is:

Starting from the turn-on condition load card sides 1 and 2. If starting from some other condition where there may be data in the data registers, then press Cms after loading the cards. Then proceed with the following sequence. Ignore the flashing displays--do not clear the flashing display at any of the initialization steps.

9-Op-17	239.89 in the display. This sets the proper partitioning for the fast mode initialization.
GTO-224-CLR	0 in the display. Sets the program counter to the location which is to be converted to h12
Pgm-12-SBR-999	Flashing 0. in the display
R/S	Flashing 0.00 in the display. If you see anything else in the display it means that you failed to clear the data registers. If so, press RST-CLR-Cms and start over.
DMS	Flashing 0 in the display.
LRN	224 44 in the display.
Ins	224 44 in the display
LRN	0 in the display
RST-CLR	0 in the display. These two key strokes return the calculator from the initialization mode to the run mode.

You may now proceed to find prime factors by placing the integer to be factored in the display register and pressing A. If you are using a printer the input integer and the factors will be printed. If you are not using a printer you can press B to call back the input integer and the factors with their multiplicity.

If you get tired of all this fast mode operation you can run in normal mode by simply entering card sides 1 and 2 and skipping the fast mode initialization process.

13 Digit Modulo 30 SFF Revisited - (cont)

Program Listing for the h12 Version of G. L. Wilson's SFF:

```
000 92 RTN
001 32 X:T
002 55 ÷
003 32 X:T
004 06 6
005 44 SUM
006 02 02
007 43 RCL
008 02 02
009 95 =
010 59 INT
011 65 ×
012 43 RCL
013 02 02
014 95 =
015 67 EQ
016 03 03
017 80 80
018 32 X:T
019 55 ÷
020 32 X:T
021 04 4
022 44 SUM
023 02 02
024 43 RCL
025 02 02
026 95 =
027 59 INT
028 65 ×
029 43 RCL
030 02 02
031 95 =
032 67 EQ
033 01 01
034 60 60
035 32 X:T
036 55 ÷
037 32 X:T
038 02 2
039 44 SUM
040 02 02
041 43 RCL
042 02 02
043 95 =
044 59 INT
045 65 ×
046 43 RCL
047 02 02
048 95 =
049 67 EQ
050 01 01
051 65 65
052 32 X:T
053 55 ÷
054 32 X:T
055 04 4
056 44 SUM
057 02 02
058 43 RCL
059 02 02
060 95 =
061 59 INT
062 65 ×
063 43 RCL
064 02 02
065 95 =
066 67 EQ
067 01 01
068 70 70
069 32 X:T
070 55 ÷
071 32 X:T
072 02 2
073 44 SUM
074 02 02
075 43 RCL
076 02 02
077 95 =
078 59 INT
079 65 ×
080 43 RCL
081 02 02
082 95 =
083 67 EQ
084 01 01
085 75 75
086 32 X:T
087 55 ÷
088 32 X:T
089 04 4
090 44 SUM
091 02 02
092 43 RCL
093 02 02
094 95 =
095 59 INT
096 65 ×
097 43 RCL
098 02 02
099 95 =
100 67 EQ
101 04 04
102 30 30
103 32 X:T
104 55 ÷
105 32 X:T
106 06 6
107 44 SUM
108 02 02
109 43 RCL
110 02 02
111 95 =
112 59 INT
113 65 ×
114 43 RCL
115 02 02
116 95 =
117 67 EQ
118 04 04
119 36 36
120 02 2
121 44 SUM
122 02 02
123 43 RCL
124 02 02
125 32 X:T
126 55 ÷
127 42 STO
128 01 01
129 43 RCL
130 02 02
131 95 =
132 22 INV
133 77 GE
134 01 01
135 50 50
136 59 INT
137 65 ×
138 43 RCL
139 01 01
140 32 X:T
141 95 =
142 22 INV
143 67 EQ
144 00 00
145 01 01
146 01 1
147 61 GTO
148 03 03
149 82 82
150 43 RCL
151 01 01
152 32 X:T
153 86 STF
154 01 01
155 69 OP
156 28 28
157 61 GTO
158 02 02
159 32 32
160 03 3
161 05 5
162 61 GTO
163 03 03
164 82 82
165 05 5
166 02 2
167 61 GTO
168 03 03
169 82 82
170 06 6
171 09 9
172 61 GTO
173 03 03
174 82 82
175 08 8
176 06 6
177 61 GTO
178 03 03
179 82 82
180 76 LBL
181 12 B
182 25 CLR
183 29 CP
184 09 9
185 42 STO
186 00 00
187 43 RCL
188 09 09
189 91 R/S
190 69 OP
191 20 20
192 73 RC*
193 00 00
194 22 INV
195 67 EQ
196 01 01
197 89 89
198 76 LBL
199 98 ADV
200 85 +
201 95 =
202 01 1
203 81 RST
204 00 0
205 76 LBL
206 11 A
207 98 ADV
208 32 X:T
209 06 6
210 69 OP
211 17 17
212 47 CMS
213 09 9
214 42 STO
215 08 08
216 60 DEG
217 02 2
218 85 +
219 52 EE
220 01 1
221 02 2
222 95 =
223 86 STF
224 50 50
225 28 LOG
226 39 COS
227 69 OP
228 66 66
229 25 CLR
230 22 INV
231 58 FIX
232 01 1
233 42 STO
234 06 06
235 52 EE
236 01 1
237 00 0
238 32 X:T
239 68 NOP
240 72 ST*
241 08 08
242 22 INV
243 52 EE
244 22 INV
245 77 GE
246 03 03
247 16 16
248 55 ÷
249 03 3
250 42 STO
251 05 05
252 04 4
253 52 EE
254 09 9
255 22 INV
256 52 EE
257 69 OP
258 04 04
259 32 X:T
260 55 ÷
261 28 LOG
262 59 INT
263 42 STO
264 07 07
265 69 OP
266 27 27
267 22 INV
268 28 LOG
269 85 +
270 22 INV
271 59 INT
272 32 X:T
273 01 1
274 85 +
275 28 LOG
276 59 INT
277 65 ×
278 01 1
279 00 0
280 00 0
281 49 PRD
282 03 03
283 02 2
284 95 =
285 59 INT
286 44 SUM
287 03 03
288 32 X:T
289 65 ×
290 01 1
291 00 0
292 97 DSZ
293 07 07
294 02 02
295 69 69
296 95 =
297 48 EXC
298 03 03
299 84 OP*
300 06 06
301 69 OP
302 26 26
303 05 5
304 42 STO
305 07 07
306 25 CLR
307 48 EXC
308 03 03
309 97 DSZ
310 05 05
311 02 02
312 69 69
313 69 OP
314 05 05
315 76 LBL
316 99 PRT
317 87 IFF
318 01 01
319 01 01
320 99 99
321 52 EE
322 43 RCL
323 09 09
324 55 ÷
325 32 X:T
326 02 2
327 42 STO
328 02 02
329 95 =
330 59 INT
331 65 ×
332 02 2
333 95 =
334 67 EQ
335 04 04
336 42 42
337 32 X:T
338 55 ÷
339 32 X:T
340 03 3
341 42 STO
342 02 02
343 95 =
344 59 INT
345 65 ×
346 03 3
347 95 =
348 67 EQ
349 04 04
350 48 48
351 32 X:T
352 55 ÷
353 32 X:T
354 05 5
355 42 STO
356 02 02
357 95 =
358 59 INT
359 65 ×
360 05 5
361 95 =
362 67 EQ
363 04 04
364 54 54
365 32 X:T
366 55 ÷
367 32 X:T
368 07 7
369 42 STO
370 02 02
371 95 =
372 59 INT
373 65 ×
374 07 7
375 95 =
376 22 INV
377 67 EQ
378 00 00
379 18 18
380 01 1
381 08 8
382 42 STO
383 04 04
384 00 0
385 42 STO
386 07 07
387 32 X:T
388 55 ÷
389 43 RCL
390 02 02
391 95 =
392 69 OP
393 27 27
394 55 ÷
395 32 X:T
396 43 RCL
397 02 02
398 95 =
399 59 INT
400 65 ×
401 43 RCL
402 02 02
403 95 =
404 67 EQ
405 03 03
406 88 88
407 25 CLR
408 43 RCL
409 02 02
410 85 +
411 93 .
412 00 0
413 01 1
414 65 ×
415 43 RCL
416 07 07
417 95 =
418 69 OP
419 28 28
420 72 ST*
421 08 08
422 99 PRT
423 01 1
424 67 EQ
425 01 01
426 99 99
427 52 EE
428 83 GO*
429 04 04
430 01 1
431 00 0
432 03 3
433 61 GTO
434 03 03
435 82 82
436 01 1
437 02 2
438 00 0
439 61 GTO
440 03 03
441 82 82
442 03 3
443 03 3
444 07 7
445 61 GTO
446 03 03
447 82 82
448 03 3
449 05 5
450 01 1
451 61 GTO
452 03 03
453 82 82
454 03 3
455 06 6
456 05 5
457 61 GTO
458 03 03
459 82 82
```

13 Digit Modulo 30 SFF Revisited - (cont)

Providing a program sequence such that both normal mode and fast mode are available requires a good understanding of how the fast mode initialization sequence alters program code. Hexadecimal codes can only be formed at program locations which are evenly divisible by eight. The process involves entering the ROM which contains the statistics and conversions functions. See the article "Hard-wired Functions" in the March/April issue of PPX Exchange for a listing of the ROM. For this program it was convenient to place the h12 command for fast mode entry at program location 224. The ROM code at locations 224 ff is shown at the right.

224	44	SUM
225	04	04
226	82	HIR
227	48	48
228	82	HIR
229	07	07
230	33	X^2
231	94	+/-
232	44	SUM
233	05	05
234	01	1
235	94	+/-
236	44	SUM
237	03	03
238	82	HIR
239	37	37

In addition to generating the hexadecimal code the synthesizing process also alters the subsequent seven program locations and inserts a new code at the eighth subsequent location. Some "cookbook" rules by which the code is altered have been defined by Patrick Acosta. For the case which pertains to this program we first write the code from program locations 224 through 231 of the ROM starting from the right. Above that code we write 00 at location 224 and the code from ROM for locations 224 through 230 at locations 225 through 231 respectively. We then perform a hexadecimal subtraction at the ones digit in the location 224 column and perform ordinary decimal subtraction for the remaining digits, borrowing from the next column to the left as required. A borrow at the left hand column, or most significant digit, is lost. The resulting 5C code at location 224 would display and print as code 62, but would not act like a code 62 if encountered in a program. The process completed so far would look like:

Location	231	230	229	228	227	226	225	224
	33	07	82	48	82	04	44	00
	94	33	07	82	48	82	04	44
	--	--	--	--	--	--	--	--
	38	74	74	66	33	22	39	5C

The values generated for each location by the above process are then added to the value of the code at locations 224 through 231 in user memory. For this program the values selected for user memory in locations 224 through 231 are from the listing on page 20. We use the same organizing format as above. We add the ones digits at location 224 in a hexadecimal sense, and add the remaining columns in a decimal sense, with carries as required. Any carry out to the left is lost. The process would look like:

Location	231	230	229	228	227	226	225	224
From RAM	58	22	25	66	69	39	28	50
From Above	38	74	74	66	33	22	39	5C
	--	--	--	--	--	--	--	--
	96	97	00	33	02	61	68	0C

The last row defines the code which will be seen in locations 224 through 231 after the hexadecimal initialization process. The synthesizing process also inserts a new code at location 232 which is comprised of the tens digit from the RAM code at location 225 and the ones digit from the ROM code at location 225; that is, a code 24 is inserted at location 232 in our

13 Digit Modulo 30 SFF Revisited - (cont)

program. The insertion process pushes down the remaining code, and pushes out of memory the code in the last program location for the partitioning in use; in our program the Nop which was at program location 239 before the initialization process.

So much for science. Programming art enters into the process when one tries to provide code which will permit the calculator to run in normal mode if the initialization is not done. or in fast mode if the initialization process is done. In this program, the location for the h12 command and the code for normal mode were selected such that the program could run through the "garbage" in locations 223 through 228 without altering the results of previous calculations. The Stflg-50 sequence will set flag 0. The Op-66 sequence will cause a flashing display, which will be cleared by the CLR at location 229. After the initialization process is complete, the Stflg-h12 sequence provides fast mode entry and the GTO-233 sequence skips past the "garbage" at locations 229 through 232.

As good as it is, Mr. Wilson's first program does contain some minor bugs:

1. The program provided a good vehicle to discuss the mechanics of the h12 method of fast mode entry. But unless there is a requirement to transition in and out of fast mode many times during execution of a program (see David Lobbestael's Profile Plot program in V8N1P24/25 for an example) the Stflg Ind method for fast mode entry will minimize user keystrokes.

2. The program, as was PPX 398278, is erratic in the indication of a multiplicity of one for the last factor. The version in V8N4P15 remedied the deficiency for last factors which are less than 1EE+10 but did not remedy the problem if the last factor was 1EE+10 or greater. Clearly, this is a minor discrepancy. George Vogel's program on page 4 of the January/February 1981 issue of PPX Exchange did no better. Nor do the various HP-41 programs which have been published in the HP PPC Calculator Journal---see V8N5P19 for example. But we are building a Rolls-Royce here, and Toyota parts just won't do.

3. It is possible to cause the program on page 20 to misbehave by a properly timed interruption in normal mode. The culprit is the Stflg 01 at locations 153/154 which controls the printer output for the last factor. The program relies on the RST at location 203 to clear the flag register. If the user should stop the program with a R/S during the time between the setting of Flag 1 at 153/154 and the RST at 202, then an incorrect condition of Flag 1 will persist for the next entry and the program will malfunction. An unlikely occurrence which will not happen in fast mode.

Mr. Wilson was told of these discrepancies. In only two weeks he returned a modified program which remedies all three deficiencies---truly a Rolls-Royce program.

The program, which is listed on page 23 is easy to run. You simply enter the integer to be tested and press A. After a short period 1. 12 flashes in the display. You press 7 EE and the program runs. If you are using a printer the input integer and all factors are printed with correct multiplicity. If you are not using a printer you wait until the solution is complete as indicated by a flashing 1 in the display, and then press B to recall the input integer and R/S as many times as required to view the factors. For large integers the solution time is $0.117\sqrt{N}$ seconds.

13 Digit Modulo 30 SFF Revisited - (cont)

Program Listing for the Stflg Ind Version of G. L. Wilson's SFF

000	91	R/S
001	76	LBL
002	12	B
003	25	CLR
004	29	CP
005	09	9
006	42	STO
007	00	00
008	43	RCL
009	09	09
010	91	R/S
011	69	OP
012	20	20
013	73	RC*
014	00	00
015	22	INV
016	67	EQ
017	00	00
018	10	10
019	85	+
020	95	=
021	01	1
022	81	RST
023	76	LBL
024	11	A
025	32	X:T
026	98	ADV
027	61	GTO
028	04	04
029	68	68
030	25	CLR
031	47	CMS
032	09	9
033	42	STO
034	08	08
035	22	INV
036	86	STF
037	01	01
038	03	3
039	02	2
040	00	0
041	33	X²
042	82	HIR
043	08	08
044	01	1
045	42	STO
046	06	06
047	52	EE
048	01	1
049	00	0
050	32	X:T
051	72	ST*
052	08	08
053	22	INV
054	52	EE
055	77	GE
056	03	03
057	06	06
058	99	PRT
059	87	IFF
060	01	01
061	03	03
062	71	71
063	52	EE
064	43	RCL
065	09	09
066	55	÷
067	32	X:T
068	02	2
069	42	STO
070	02	02
071	95	=
072	59	INT
073	65	×
074	02	2
075	95	=
076	67	EQ
077	03	03
078	97	97
079	32	X:T
080	55	÷
081	32	X:T
082	03	3
083	42	STO
084	02	02
085	95	=
086	59	INT
087	65	×
088	03	3
089	95	=
090	67	EQ
091	04	04
092	02	02
093	32	X:T
094	55	÷
095	32	X:T
096	05	5
097	42	STO
098	02	02
099	95	=
100	59	INT
101	65	×
102	05	5
103	95	=
104	67	EQ
105	04	04
106	07	07
107	01	1
108	42	STO
109	02	02
110	32	X:T
111	55	÷
112	32	X:T
113	06	6
114	44	SUM
115	02	02
116	43	RCL
117	02	02
118	95	=
119	59	INT
120	65	×
121	43	RCL
122	02	02
123	95	=
124	67	EQ
125	04	04
126	13	13
127	32	X:T
128	55	÷
129	32	X:T
130	04	4
131	44	SUM
132	02	02
133	43	RCL
134	02	02
135	95	=
136	59	INT
137	65	×
138	43	RCL
139	02	02
140	95	=
141	67	EQ
142	04	04
143	19	19
144	32	X:T
145	55	÷
146	32	X:T
147	02	2
148	44	SUM
149	02	02
150	43	RCL
151	02	02
152	95	=
153	59	INT
154	65	×
155	43	RCL
156	02	02
157	95	=
158	67	EQ
159	04	04
160	25	25
161	32	X:T
162	55	÷
163	32	X:T
164	04	4
165	44	SUM
166	02	02
167	43	RCL
168	02	02
169	95	=
170	59	INT
171	65	×
172	43	RCL
173	02	02
174	95	=
175	67	EQ
176	04	04
177	31	31
178	32	X:T
179	55	÷
180	32	X:T
181	02	2
182	44	SUM
183	02	02
184	43	RCL
185	02	02
186	95	=
187	59	INT
188	65	×
189	43	RCL
190	02	02
191	95	=
192	67	EQ
193	04	04
194	37	37
195	32	X:T
196	55	÷
197	32	X:T
198	04	4
199	44	SUM
200	02	02
201	43	RCL
202	02	02
203	95	=
204	59	INT
205	65	×
206	43	RCL
207	02	02
208	95	=
209	67	EQ
210	04	04
211	43	43
212	32	X:T
213	55	÷
214	32	X:T
215	06	6
216	44	SUM
217	02	02
218	43	RCL
219	02	02
220	95	=
221	59	INT
222	65	×
223	43	RCL
224	02	02
225	95	=
226	67	EQ
227	04	04
228	49	49
229	02	2
230	44	SUM
231	02	02
232	43	RCL
233	02	02
234	32	X:T
235	55	÷
236	42	STO
237	01	01
238	43	RCL
239	02	02
240	95	=
241	22	INV
242	77	GE
243	03	03
244	76	76
245	59	INT
246	65	×
247	43	RCL
248	01	01
249	32	X:T
250	95	=
251	22	INV
252	67	EQ
253	01	01
254	10	10
255	01	1
256	01	1
257	00	0
258	42	STO
259	04	04
260	00	0
261	42	STO
262	07	07
263	32	X:T
264	55	÷
265	43	RCL
266	02	02
267	95	=
268	69	OP
269	27	27
270	55	÷
271	32	X:T
272	43	RCL
273	02	02
274	95	=
275	59	INT
276	65	×
277	43	RCL
278	02	02
279	95	=
280	67	EQ
281	02	02
282	64	64
283	69	OP
284	28	28
285	25	CLR
286	43	RCL
287	02	02
288	85	+
289	93	.
290	00	0
291	01	1
292	65	×
293	43	RCL
294	07	07
295	95	=
296	72	ST*
297	08	08
298	99	PRT
299	01	1
300	67	EQ
301	03	03
302	71	71
303	52	EE
304	83	GO*
305	04	04
306	55	÷
307	03	3
308	42	STO
309	05	05
310	32	X:T
311	55	÷
312	28	LOG
313	59	INT
314	42	STO
315	07	07
316	69	OP
317	27	27
318	22	INV
319	28	LOG
320	85	+
321	22	INV
322	59	INT
323	32	X:T
324	01	1
325	85	+
326	28	LOG
327	59	INT
328	65	×
329	01	1
330	00	0
331	00	0
332	49	PRD
333	03	03
334	02	2
335	95	=
336	59	INT
337	44	SUM
338	03	03
339	32	X:T
340	65	×
341	01	1
342	00	0
343	97	DSZ
344	07	07
345	03	03
346	20	20
347	95	=
348	48	EXC
349	03	03
350	84	OP*
351	06	06
352	69	OP
353	26	26
354	05	5
355	42	STO
356	07	07
357	00	0
358	48	EXC
359	03	03
360	97	DSZ
361	05	05
362	03	03
363	20	20
364	69	OP
365	05	05
366	22	INV
367	87	IFF
368	01	01
369	00	00
370	63	63
371	98	ADV
372	85	+
373	95	=
374	01	1
375	81	RST
376	43	RCL
377	01	01
378	85	+
379	93	.
380	00	0
381	01	1
382	95	=
383	32	X:T
384	86	STF
385	01	01
386	93	.
387	01	1
388	00	0
389	02	2
390	82	HIR
391	38	38
392	69	OP
393	28	28
394	61	GTO
395	00	00
396	44	44
397	07	7
398	09	9
399	61	GTO
400	02	02
401	58	58
402	09	9
403	03	3
404	61	GTO
405	02	02
406	58	58
407	01	1
408	00	0
409	07	7
410	61	GTO
411	02	02
412	58	58
413	01	1
414	02	2
415	07	7
416	61	GTO
417	02	02
418	58	58
419	01	1
420	04	4
421	04	4
422	61	GTO
423	02	02
424	58	58
425	01	1
426	06	6
427	01	1
428	61	GTO
429	02	02
430	58	58
431	01	1
432	07	7
433	08	8
434	61	GTO
435	02	02
436	58	58
437	01	1
438	09	9
439	05	5
440	61	GTO
441	02	02
442	58	58
443	02	2
444	01	1
445	02	2
446	61	GTO
447	02	02
448	58	58
449	02	2
450	02	2
451	09	9
452	61	GTO
453	02	02
454	58	58
455	00	0
456	00	0
457	00	0
458	00	0
459	00	0
460	00	0
461	00	0
462	00	0
463	00	0
464	00	0
465	00	0
466	00	0
467	00	0
468	03	3
469	05	5
470	02	2
471	85	+
472	01	1
473	52	EE
474	01	1
475	02	2
476	95	=
477	60	DEG
478	86	STF
479	40	IND

PPX PROGRAM LISTING - In January of this year Texas Instruments announced plans to discontinue the PPX-59 Club by the end of the first quarter of 1983. The same TI letter announced that all programs in stock would be discounted and made available for purchase for $2.50 each. Recent correspondence from our newer club members reported that programs were no longer available from PPX Exchange. I called the TI Customer Relations Department at 800-858-1802. I was told that the group which supported PPX for the TI-58/59 had been disbanded and no support was available. I guess that confirms that we are orphans of sorts.

Immediately after the announcement that the PPX-59 Club would be discontinued both Maurice Swinnen and I entered into extended discussions with TI in an attempt to preserve access to the program library. We were not successful. In order to preserve access to at least some of the programs in PPX Exchange the editor suggests that club members should set up an informal exchange. To that end I have listed the programs which I own. The six programs listed on this page are ones which I submitted. The fifteen programs on page 25 are ones which I either purchased or which were received in return for an accepted program. I have provided the names and addresses of the authors in case you would wish to contact them. I am willing to loan any of these programs to club members with the understanding that they will be promptly returned. I ask that you send one dollar (no checks please, two dollars overseas) for each program requested to cover postage and handling. Other club members who are willing to provide a similar service are asked to send a list of the programs that they can make available and the terms under which they can be made available. I will publish what I can about availability, consistent with other constraints of the news-letter. I have no intent or inclination to provide a copying service. Is some member is so inclined?

208059 - Polynomial Curve Fit with Errors - 26 pages

208081 - Linearity Analysis from Multipoint Data - 24 pages

398225 - Prime Factors of an Integer - 8 pages

398278 - 13 Digit Modulo 30 Speedy Factor Finder - 17 pages

908175 - Memory Malfunction Diagnostic - 13 pages

908192 - High Speed Calendar Printer - 19 pages

DOUBLE PRECISION LIMITATIONS - In the discussions of the ability of the various computers to recover pi in BASIC we have neglected to mention that some versions of BASIC provide double precision options. Examples include the IBM PC and the Radio Shack Model 3; but while the delivered BASIC for both of those computers provides designation of double precision variables, various users report that the calculation of the transcendental functions still are performed in single precision. This helps to understand why the double precision pi value recovered from the IBM PC as reported in V8N4P18 was correct only to the single precision accuracy.

PPX Program Listing (cont)

188052 - Portfolio Monitor - 13 pages
John E. Binns
2600 S. Kenner Highway, M-10, Stuart FL 33494

208009 - Two Variable Polynomial Curve Fit - 13 pages
Randall Mundt
8111 Morley, Houston TX 77061

208077 - Nonlinear Least Squares - 15 pages
John R. Long
Dept. of Chemistry, Drew University, Madison NJ 07940

248007 - Random Date Generator - 8 pages
Barbara C. Hevener
2111 Robin Road, Columbia SC 29204

398010 - Least Mean Square Fit of a Polynomial - 7 pages
Joseph A. Walston
9745 Wisterwood, Dallas TX 75238

398131 - High Speed Prime Tester - 6 pages
Alan L. Zeichick
28 Kennebec Place, Bangor ME 04401

398171 - Exact Factorials to 610 Factorial - 6 pages
Frank M. Fujimoto
7430 Hondo Street, Downey CA 90242

398201 - Prime Number Data Base Generator - 16 pages
John H. Brundage
18 Farrington Street, W. Caldwell NJ 07006

638034 - Four Function 16 Bit Binary Arithmetic - 18 pages
Charles S. Gaylord
12004 W. 82 Terrace, Lenexa, KS 66215

698004 - Perspective 3d Option, Illustrator's Aid - 8 pages
Texas Instruments

908063 - Hierarchy Register Functions - 9 pages
John C. Sellers
Box 151C, R.D. #1, Lee Center NY 13363

908104 - Stop Watch Timer - 11 pages
Rob Wegink
Woltersweg 50, Hengelo OV Netherlands

908119 - Magnetic Card Comparator - 6 pages
William Skillman
605 Forest View Road, Linthicum MD 21090

918217 - Backgammon - 21 pages
J. Brian Sladen
3872 North Lakewood Drive, Memphis TN 38128

918182 - Dungeon Master's Aid I - 9 pages
Michael A. Henry
316 Corduba, St. Peters MO 63376

<u>TI-57 VS TI-57LCD</u> - Bjorn Gustavsson of Sweden submitted a comparison of the capability of the TI-57 and the newer TI-57LCD in July of 1982. I obtained a TI-57LCD from Elek-Tek late last year. I delayed publishing much material about the device until it became available at local distributors.

Improvements in the TI-57LCD relative to the TI-57 include:

1. Continuous memory.
2. The liquid crystal display with the attendant reduction in power requirements. The user is free of the battery charging concerns which are present with all of the LED calculators.
3. There is a factorial function and it is fast. 69! is obtained in only four seconds.
4. Register arithmetic is obtained with STO followed by the operation to be performed and the register involved. Thus, what would have been a INV SUM 1 on the TI-57 becomes STO - 1. Similarly, the Prd 1 of the TI-57 becomes STO x 1 . This is the same methodology as with the TI-55II.
5. CM (Clear Memories) will just clear the registers, not the display and the t register. INV C.t. on the TI-57 would clear everything except for the program.
6. Special conditions are indicated in the display; examples include 2nd, INV, and the RAD and GRAD angular modes.

Deteriorations in the TI-57LCD relative to the TI-57 include:

1. The TI-57LCD has only 48 program steps, one memory register, and a t register. Eight program steps can be traded for each additional memory register up to a total of seven memory registers (no program steps then) plus a t register. Contrast that with the TI-57 which has eight memory registers including register 7 which doubles as a t register <u>and</u> 50 program steps. Although program steps cannont be traded for additional memory registers with the TI-57, the TI-57 is clearly a more powerful device. As a result many of the good old TI-57 programs will not be able to be run on the TI-57LCD. An example is Peter Van Roy's Mastermind program from V5N6P12, since it required 50 program steps <u>and</u> 8 working memory registers. Similarly, R. van Genechten's Exact Factorial program from V8N3P12 cannot be converted to the TI-57LCD.
2. Statistics functions are not available on the TI-57LCD.
3. There is only one level of subroutine on the TI-57LCD. The TI-57 had two.
4. As indicated in V8N2P19 the TI-57LCD can be over a factor of two slower for some calculations as compared with the TI-57.

The conclusion: the TI-57LCD provides improved portability at the expense of reduced capability.

<u>STILL MORE ON NUMERICAL PRECISION</u> - Bob Fruit's proposed test using a compound interest problem was interesting (no pun intended) to several readers. An important issue was the value of the correct answer. Laurance Leeds calculated the value as

2260.48792 47960 86067 64793 +

George Thomson calculated 35 digits to be

2260.48792 47960 86067 64793 83933 44540 3

And, Carl Rabe found 49 digits to be

2260.48792 47960 86067 64793 83933 44540 34437 92694 53760

OMISSION - Stereo Graphics with Ball-Stick Option - The discussion of this program system in V8N4P27/28 failed to give the address for obtaining the material. Send orders to:

Dr. D. M. Graham
2149 Scarboro Avenue
Vancouver, B.C.
Canada V5P 2L2

A Note on the HP-41 Accuracy (see PPC Notes V8N4P2 & 4)--W.J. Widmer

(Page 2): for 2.543210631**2.5 the HP-41 outputs 10.31468159 with rounded 9, not 8 as given by HP-67.
(Page 4): on HP-41 the output for 10% monthly for 30 years = 2260.487641

Also, I did a series of [$\sqrt{x}$][x^2] tests on the HP-41 (& on HP-67 which gave the same results)--see Editor's Note on page 2--with the following results (incidentally, [y^x] done n times followed by [$y^{1/x}$] done n times gives same results as shown--this in partial answer to Laurance's query):

Sequence for $\sqrt{x}$ n times followed by x^2 n times for indicated values of x

x	n = 1	n = 2	n = 3	n = 4
2	1.999999999	1.999999999	2.000000007	2.000000007
10	9.999999999	9.999999999	9.999999999	10.00000004
20	20.00000000	20.00000001	19.99999996	19.99999996

x	n = 5	n = 10	n = 15	n = 20
2	2.000000022	2.000001028	1.999963249	1.999897829
10	9.999999929	9.999997842	10.00010228	9.989948090
20	20.00000027	19.99999784	20.00003791	20.00068575

For x = 20, I note that

n = 7	gives 20.00000027	n = 12	gives 19.99995703
n = 8	" 19.99999784	n = 13	" 20.00003791
n = 9	" 19.99999784	n = 14	" 20.00003791
n = 11	" 19.99999784	n = 25	" 19.81288610

The errors seem unevenly periodic and probably involve unevenly cumulative effects of round-up & round-down in the round-off process

Now all of the above might be of superfluous interest (especially to TI/AOS users) except for one important point. Note the result for x = 20 with n = 1. The common "trick" for generating a small non-zero value by pressing (or programming) [$\sqrt{x}$] [x^2] for testing against y (or against t on TI's) for =, ≠, or GE may not always work (as in this case for 20 in the display initially). Thus, 20[$\sqrt{x}$][x^2] [20][-][1/x] gives "data error" on HP-41 and "error" on HP-67. This does not occur on the TI-59 (with x = 20); the key-in sequence on this begets -0.00000000001, so that 1/x then gives non-zero. Certainly, RPN users should be aware of this in translating TI-59 AOS programs into RPN on the HP-41/67. I, myself, had not appreciated this before doing the testing above-noted.

Editors Note: Limited tests with an HP-11 indicate entirely unpredictable results for the sequence (N)($\sqrt{x}$)(x^2)(N)(-)(1/x). The test yields an "Error 0" for input integers of 1, 4, 7, 9, 11 through 31, 34 through 40, 43, 44, 46, 48 through 50, etc. In contrast the sequence (N)($\sqrt{x}$)(x^2)(-)(N)(=)(1/x) will yield error indications on a TI-58 or TI-59 only for the input integers which are perfect squares. The same consistency holds with the TI-57 and TI-57LCD, where I have tested up to input integers of 6400.

<u>THE GROSH OF FINN REVISITED</u> - C. Williamson and L. Leeds. In V8N2P5 W. J. Widmer stated the problem as: "Consider any positive integer N(0). Raise each digit in this integer to the same power p, and sum these to form a new integer N(1); do the same with N(1) to form N(2), and then with N(2) to form N(3), etc. If this process is continued, the sequence of successive N's thus formed eventually becomes cyclic in a finite number of steps, i, which is unique for the given N(0); and, further, only a finite number of cyclic patterns are possible for a given power, p." He provided a program for p = 2. A program was requested which would solve for at least p = 6, and which would not require entry of both the input integer and the number of digits in the input integer.

Charlie Williamson responded with the following program which does not even require calculation of the number of digits in the input integer. The program makes effective use of the t register--that is a characteristic of many of his programs. Charlie also uses BST (code 51) as the dummy operator to bring a value inside parenthesis. Others have proposed the use of IND (code 40) and INS (code 46) -- see V5N6P3 and V5N8P2. Code 51 can be generated by the sequence RCL 51 BST BST Del SST. CE (code 24) works just as well if the user doesn't worry about the clearing of error indications.

```
000  76 LBL    011  59 INT    022  01  1     033  85  +     044  11  A
001  12  B     012  53  (     023  00  0     034  00  0     045  42 STO
002  32 X:T    013  51 BST    024  54  )     035  22 INV    046  00  00
003  00  0     014  85  +     025  53  (     036  67  EQ    047  25 CLR
004  85  +     015  32 X:T    026  51 BST    037  00  00    048  43 RCL
005  53  (     016  00  0     027  45 YX     038  04  04    049  00  00
006  32 X:T    017  54  )     028  43 RCL    039  95  =     050  92 RTN
007  55  ÷     018  54  )     029  00  00    040  22 INV
008  01  1     019  53  (     030  54  )     041  52 EE
009  00  0     020  51 BST    031  52 EE     042  92 RTN
010  75  -     021  65  ×     032  95  =     043  76 LBL
```

To run the program, enter p and press A. Enter N(0) and press B. See N(1) returned to the display. Record that value, Press B again to generate N(2) from N(1), etc.

Laurance Leeds provided a similar program which will handle N(0) of ten digits and p of 2 through 9, and which prints each new N as it is found and proceeds to calculate and print the succeeding n.

```
000  47 CMS    011  00  00    022  01  1     033  02  02    044  99 PRT
001  42 STO    012  55  ÷     023  00  0     034  29 CP     045  42 STO
002  00  00    013  01  1     024  95  =     035  43 RCL    046  00  00
003  98 ADV    014  00  0     025  45 YX     036  00  00    047  25 CLR
004  99 PRT    015  95  =     026  43 RCL    037  22 INV    048  42 STO
005  91 R/S    016  75  -     027  01  01    038  67  EQ    049  02  02
006  42 STO    017  59 INT    028  95  =     039  00  00    050  61 GTO
007  01  01    018  42 STO    029  52 EE     040  12  12    051  00  00
008  98 ADV    019  00  00    030  22 INV    041  43 RCL    052  10  10
009  99 PRT    020  95  =     031  52 EE     042  02  02
010  43 RCL    021  65  ×     032  44 SUM    043  98 ADV
```

To use the program, enter N(0) and press RST R/S. Enter the exponent p and press R/S. The program will run and print out succeeding N's until interrupted with R/S. Laurance observes that if both the leader and the cyclic period are long, then the user would have difficulty in recognizing when the cyclic period begins. He then provides the following general solution, which requires the determination of

(1) The number of terms in the cyclic period (CP) and their values.

(2) The number of terms in the leader and their values.

(3) The first term of the CP which follows the leader.

The Grosh of Finn Revisited (cont)

The algorithm used is as follows:

Two separate generators are operated sequentially.
Let X(0) generate X(1) in path 1.
Let X(0) generate X(1), and X(1) generate X(2) in path 2.
When X(n) = X(2n), X(n) is a term somewhere in the CP.
Seed a single step path with X(n).
When X(n) generates X(n) the number of terms in the CP and thier values have been determined.
Seed two single step paths, one with X(0), the other with X(n).
Record the number of terms and the term values in the X(0) path. When both paths generate X(k), X(k) is the first term of the CP and (k - 1) is the count of the leader.

The program listing is on page 30.

User Instructions:

Limitations: N = ten digits maximum. Exponent = 2 through 9.

Enter N(0) and press RST R/S. Enter the exponent and press R/S. There are three possible returns:

(1) Return a flashing 11. (A series where the CP is 1).
Press R/S. Return = 1 , the number of terms of the CP.
Press R/S. Return the leader count.
Press R/S again and again to obtain the terms of the leader.
A return of 1/3 = .3333333333 indicates the end. If you press R/S again you can repeat the terms or the leader.

(2) Return a flashing 22. (A series where the seed is a member of the CP).
Press R/S. Return = the number of terms of the CP.
Press R/S again and again to obtain the terms of the CP. A return of 1/3 = .3333333333 indicates the end.

(3) Return a flashing 33. (A general series of leader plus CP).
Press R/S. Return = the number of terms of the CP.
Press R/S again and again to obtain the terms of the CP. A return of 1/3 = .3333333333 indicates the end.

Press E to obtain the leader count and leader values. The program will run for some time and return a flashing 44.
Press R/S. Return = the leader count.
Press R/S again and again to return the terms of the leader and the first term of the CP. A return of 1/3 = .3333333333 indicates the end.

LBL E is to be used only after a flashing 33 return. Only 50 registers are available for recording term values. If either the CP or the leader exceed 50 terms the storage will cease, but the number of terms will accumulate correctly.

Some Results:

N(0) = 13139, p = 4. A seven term cycle with no leader.

N(0) = 33, p = 3. A five term leader and a one term CP of 153.

N(0) = 9,999,999,999 , p = 9. A 101 term leader and an 80 term CP. The first term of the CP is 433,916,322 .

N(0) = 9,999,999,997 , p = 9. A 24 term leader and a 24 term CP. The first term of the CP is 54,639,064.

The Grosh of Finn Revisited - (cont)

Program Listing:

```
000  47 CMS
001  42 STO
002  00  00
003  42 STO
004  04  04
005  42 STO
006  05  05
007  42 STO
008  08  08
009  01  1
010  00  0
011  42 STO
012  09  09
013  91 R/S
014  42 STO
015  01  01
016  43 RCL
017  00  00
018  55  ÷
019  01  1
020  00  0
021  95  =
022  75  -
023  59 INT
024  42 STO
025  00  00
026  95  =
027  65  ×
028  01  1
029  00  0
030  95  =
031  45 YX
032  43 RCL
033  01  01
034  95  =
035  52 EE
036  22 INV
037  52 EE
038  44 SUM
039  02  02
040  29 CP
041  43 RCL
042  00  00
043  22 INV
044  67  EQ
045  00  00
046  18  18
047  69 OP
048  23  23
049  43 RCL
050  02  02
051  72 ST*
052  09  09
053  32 X:T
054  69 OP
055  29  29
056  43 RCL
057  04  04
058  67  EQ
059  01  01
060  58  58
061  01  1
062  67  EQ
063  01  01
064  96  96
065  87 IFF
066  00  00
067  02  02
068  55  55
069  43 RCL
070  05  05
071  55  ÷
072  01  1
073  00  0
074  95  =
075  75  -
076  59 INT
077  42 STO
078  05  05
079  95  =
080  65  ×
081  01  1
082  00  0
083  95  =
084  45 YX
085  43 RCL
086  01  01
087  95  =
088  52 EE
089  22 INV
090  52 EE
091  44 SUM
092  06  06
093  29 CP
094  43 RCL
095  05  05
096  22 INV
097  67  EQ
098  00  00
099  71  71
100  43 RCL
101  06  06
102  42 STO
103  05  05
104  25 CLR
105  42 STO
106  06  06
107  43 RCL
108  05  05
109  55  ÷
110  01  1
111  00  0
112  95  =
113  75  -
114  59 INT
115  42 STO
116  05  05
117  95  =
118  65  ×
119  01  1
120  00  0
121  95  =
122  45 YX
123  43 RCL
124  01  01
125  95  =
126  52 EE
127  22 INV
128  52 EE
129  44 SUM
130  06  06
131  20 CLR
132  43 RCL
133  05  05
134  22 INV
135  67  EQ
136  01  01
137  09  09
138  43 RCL
139  06  06
140  42 STO
141  05  05
142  32 X:T
143  43 RCL
144  02  02
145  42 STO
146  00  00
147  67  EQ
148  02  02
149  15  15
150  25 CLR
151  42 STO
152  02  02
153  42 STO
154  06  06
155  61 GTO
156  00  00
157  16  16
158  87 IFF
159  00  00
160  02  02
161  65  65
162  00  0
163  35 1/X
164  02  2
165  02  2
166  91 R/S
167  25 CLR
168  43 RCL
169  03  03
170  42 STO
171  07  07
172  91 R/S
173  01  1
174  00  0
175  42 STO
176  09  09
177  73 RC*
178  09  09
179  91 R/S
180  69 OP
181  29  29
182  97 DSZ
183  07  07
184  01  01
185  77  77
186  03  3
187  35 1/X
188  91 R/S
189  43 RCL
190  03  03
191  42 STO
192  07  07
193  61 GTO
194  01  01
195  73  73
196  00  0
197  35 1/X
198  01  1
199  01  1
200  91 R/S
201  25 CLR
202  43 RCL
203  03  03
204  75  -
205  01  1
206  95  =
207  91 R/S
208  43 RCL
209  03  03
210  42 STO
211  07  07
212  61 GTO
213  01  01
214  73  73
215  43 RCL
216  04  04
217  42 STO
218  06  06
219  43 RCL
220  02  02
221  42 STO
222  00  00
223  42 STO
224  04  04
225  43 RCL
226  03  03
227  42 STO
228  07  07
229  42 STO
230  08  08
231  01  1
232  00  0
233  42 STO
234  09  09
235  86 STF
236  00  00
237  25 CLR
238  42 STO
239  02  02
240  72 ST*
241  09  09
242  69 OP
243  29  29
244  97 DSZ
245  07  07
246  02  02
247  40  40
248  01  1
249  00  0
250  42 STO
251  09  09
252  61 GTO
253  00  00
254  16  16
255  43 RCL
256  02  02
257  42 STO
258  00  00
259  25 CLR
260  42 STO
261  02  02
262  61 GTO
263  00  00
264  16  16
265  00  0
266  35 1/X
267  03  3
268  03  3
269  91 R/S
270  25 CLR
271  43 RCL
272  03  03
273  75  -
274  43 RCL
275  08  08
276  95  =
277  42 STO
278  03  03
279  42 STO
280  07  07
281  91 R/S
282  61 GTO
283  01  01
284  73  73
285  00  0
286  00  0
287  00  0
288  00  0
289  00  0
290  00  0
291  00  0
292  00  0
293  00  0
294  00  0
295  00  0
296  00  0
297  00  0
298  00  0
299  00  0
300  76 LBL
301  15  E
302  01  1
303  00  0
304  42 STO
305  09  09
306  43 RCL
307  02  02
308  42 STO
309  05  05
310  43 RCL
311  06  06
312  42 STO
313  00  00
314  43 RCL
315  03  03
316  42 STO
317  07  07
318  25 CLR
319  42 STO
320  03  03
321  42 STO
322  02  02
323  42 STO
324  06  06
325  72 ST*
326  09  09
327  69 OP
328  29  29
329  97 DSZ
330  07  07
331  03  03
332  25  25
333  01  1
334  00  0
335  42 STO
336  09  09
337  43 RCL
338  00  00
339  55  ÷
340  01  1
341  00  0
342  95  =
343  75  -
344  59 INT
345  42 STO
346  00  00
347  95  =
348  65  ×
349  01  1
350  00  0
351  95  =
352  45 YX
353  43 RCL
354  01  01
355  95  =
356  52 EE
357  22 INV
358  52 EE
359  44 SUM
360  02  02
361  29 CP
362  43 RCL
363  00  00
364  22 INV
365  67  EQ
366  03  03
367  39  39
368  69 OP
369  23  23
370  43 RCL
371  02  02
372  72 ST*
373  09  09
374  69 OP
375  29  29
376  43 RCL
377  05  05
378  55  ÷
379  01  1
380  00  0
381  95  =
382  75  -
383  59 INT
384  42 STO
385  05  05
386  95  =
387  65  ×
388  01  1
389  00  0
390  95  =
391  45 YX
392  43 RCL
393  01  01
394  95  =
395  52 EE
396  22 INV
397  52 EE
398  44 SUM
399  06  06
400  29 CP
401  43 RCL
402  05  05
403  22 INV
404  67  EQ
405  03  03
406  78  78
407  43 RCL
408  02  02
409  32 X:T
410  43 RCL
411  06  06
412  67  EQ
413  04  04
414  31  31
415  43 RCL
416  06  06
417  42 STO
418  05  05
419  43 RCL
420  02  02
421  42 STO
422  00  00
423  25 CLR
424  42 STO
425  02  02
426  42 STO
427  06  06
428  61 GTO
429  03  03
430  37  37
431  00  0
432  35 1/X
433  04  4
434  04  4
435  91 R/S
436  25 CLR
437  43 RCL
438  03  03
439  42 STO
440  07  07
441  42 STO
442  08  08
443  75  -
444  01  1
445  95  =
446  91 R/S
447  01  1
448  00  0
449  42 STO
450  09  09
451  73 RC*
452  09  09
453  91 R/S
454  69 OP
455  29  29
456  97 DSZ
457  07  07
458  04  04
459  51  51
460  03  3
461  35 1/X
462  91 R/S
463  43 RCL
464  08  08
465  42 STO
466  07  07
467  61 GTO
468  04  04
469  47  47
470  00  0
471  00  0
472  00  0
473  00  0
474  00  0
475  00  0
476  00  0
477  00  0
478  00  0
479  00  0
```