TI PPC NOTES

NEWSLETTER OF THE TI PROGRAMMABLE CALCULATOR CLUB

P.O. Box 1421, Largo, FL 34294

Volume 9, Number 2 March/April 1984

Once again I must apologize for a late issue. My recovery from surgery in January has been unusually slow, and complicated by back problems. As a result I expect the next issue will be late as well. Thank you for your patience.

This issue extends the coverage of the TI-66. More users are coming on-line and are discovering differences between the TI-66 and the TI-58C/59 which are not described in the TI-66 manual.

A "quirk" which seems most unusual is the non-commutative multiply on the TI-58C/59. I have yet to describe it to anyone who doesn't express complete surprise. Perhaps some users were aware of the effect--if so, they weren't sharing their knowledge. I have searched old issues of 52 Notes, TI PPC Notes, PPX Exchange and TISOFT withour finding any mention of the problem.

Equally surprising is the availability of a relatively straightforward workaround, the use of a double divide. Users should apply that idea with caution until it has been more thoroughly tested. My tests showed that double divides yielded two orders of magnitude improvement in accuracy for the 7 x 7 sub-Hilbert test; but, for the 4 x 4 sub-Hilbert problem the use of multiplies yields somewhat better results--perhaps due to proportionately more terminating fractions in the matrix elements. Even with the double divides the 9 x 9 sub-Hilbert problem yields unsatisfactory results. Also, a last minute input from George Thomson indicates that he has found double divide errors as high as seven in the thirteenth digit--still much better than most results from the use of multiplies.

ERRATA - Prins,Hanson,Leeds 2
USE OF RTN TO LEAVE FAST MODE - G. Thomson 3
AN UNUSUAL TI-66 ERROR STATE - D. Leising 4
TI-66 ERROR WITH FULL SBR RTN REGISTER - D. Wisander .. 4
INSIDE THE TI-66 - D. Leising & K. Ward 5
THE GREAT CALENDAR RACE - P. Hanson 6
A 83 SECOND CALENDAR - P. Acosta 7
A CALENDAR FOR THE TI-66/PC-200 - D. Leising 9
ANOTHER TEST OF PRECISION - from Scientific American .. 11
POWERS AND ROOTS FOR TI-66 AND TI-59 - M. Boland 12
TRIGONOMETRY ON THE TI-66 - P. Hanson - 12
ROBERT PRINS BRAINTEASER 13
PRINTER PAPER - L. Matteson 13
CARD READER CLEANING - P. Sperry 13
PRINT HEAD CLEANING - P. Hanson 13
A TAPE DRIVE FOR THE CC-40 ? 14
BACK ISSUES OF 52-NOTES - R. Vanderburgh 14
HARDWARE AVAILABILITY - J. Gallego 14
MULTIPLY NOT COMMUTATIVE ON THE TI-59 - G. Thomson 15
WORK-AROUND FOR NOM-COMMUTATIVE MULTIPLY - P. Hanson .. 17
POLYNOMIAL REGRESSION WITH VARIANCE - G. Friel 20

Magnetic Card Service

Magnetic cards are available for programs in this issue on the same basis as in other issues--one dollar per card plus a stamped and self-addressed envelope. Thus the price is two dollars for Acosta's calendar program, and three dollars for Friel's Polynomial Curve Fit.

ERRATA:

Hidden Digits Viewer - V6N9/10P29 - Robert Prins finds that the divide at location 056 should be deleted. To demonstrate this he suggests creating a difficult number by pressing 2 $\sqrt{x}$ x^2 INV INT which yields .9999999999950 in the display register. Running the routine with the divide in place at location 056 yields an incorrect result of 1. -87 . With the divide removed the revised routine will yield the correct 950 in the display. This routine does not display leading zeroes. In response to the number 1,111,111,111 x 1000 + 13 = in the display register the routine returns 13 to the display, and the user must assume the leading zero to obtain the correct answer of 013.

In the EE mode there will be five hidden digits. Thus, other readers have noted that this routine is a guard digit viewer not a hidden digit viewer. Some earlier routines such as that in V6N3P7 return the eight or nine least significant digits, and as such are hidden digit viewers, but fail to return trailing zeroes. Most of these earlier routines will return leading zeroes, and you have to count the displayed digits in order to determine whether there are trailing zeroes. But with the V6N3P7 routine counting the digits may not suffice. In response to an input of pi the routine returns 0.59265359, but in response to 2 $\sqrt{x}$ x^2 INV INT the routine returns 0.999999995 . There is no obvious way to determine the existence of the trailing zero. The V6N3P7 routine works best when there are clearly discernible digit sequences which assist the user in determining the existence of trailing zeroes.

(Editor's Note: Charlie Williamson's routine did not contain the extra divide. I inadvertently added it during a review of the routine for Maurice Swinnen, and no one noticed until Robert's careful testing.)

TI-58C Memory Protection - Palmer Hanson. V9N1P19 discussed memory protection for the TI-66, the CC-40 and the TI-58C. Testing at that time showed that a TI-58C would hold memory for two days without the battery installed. Subsequent testing showed that the TI-58C memory was held for seven days without the battery, but was lost after 14 days without the battery. I have not tried to interpolate.

Enhanced ML-02 Using PGM-MM-R/S - Palmer Hanson. In subsequent use of this program I found some inadvertent errors had been made in typing the user instructions on V8N6P8:

- Paragraph 4 should begin "Press B or R/S when you are ready to calculate the determinant. ...".

- Paragraph 5.a should begin "If the value of the determinant is not zero, press C or R/S. A "B" is printed to indicate the calculator is ready to accept vector elements. ..."

Digit Reverser Puzzle - Laurance Leeds questions the need for the IND immediately after LBL pi in this program from V8N1P5. My tests indicate no problems after deleting that instruction. Program author Myer Boland confirms that the instruction can be deleted.

ERRATA (cont)

The Use of RTN to Leave Fast Mode - V9N1P15 - George Thomson found that the demonstration program does not remove the calculator from TRACE mode at exit from fast mode. After some joint experimentation we have arrived at a better description of the effects. We will start with a demonstration program like that in V9N1P15:

```
050    LBL A Cms SBR 469 CLR Pause RCL 36 R/S
400    2 5 STO 36 Nop Nop Nop RTN
469    4 9 7 2 + N EE 1 2 = STF
```

where the N at location 474 may be an integer from 1 through 9. Press A to run the program. The calculator will stop with a flashing N. 12 in the display. Press 7 and then EE to enter fast mode at location 400 (49*8 + 7 + 1, from V8N4P15). The value 25 is stored in data memory 36. The RTN at location 407 causes exit from fast mode and a return to location 056, immediately after the subroutine call. You will see a flashed zero (the CLR at location 056 followed by the Pause at 057) followed by a steady 25 (recalled from register 36). But if you have a printer installed the reactions which follow depend on the value of N:

If N = 1, then flag 9 will be reset, and flag 4 will be set at fast mode entry (see V6N8P3). The calculator will stop with 25 in the display. There will be no printing.

If N = 2, then flag 9 will be set, and flag 4 will be reset at fast mode entry. The calculator will enter TRACE mode at fast mode exit, yielding a printout like that at the right. You can eliminate the TRACE mode by either pressing RST or INV Stflg 9 from the keyboard. Similarly, you can avoid the TRACE mode by resetting flag 9 in the program; for example, by replacing the three Nop's at locations 404 through 406 of the demonstration program with an INV STF 9 sequence.

```
2.5 01?
            CLR
 0.         RCL
             36
25.
            R/S
```

If N = 3 both flags 4 and 9 will be set at fast mode entry. Other than the setting of flag 4 the responses will be the same as for N = 2.

If N = 4 (the case in the demonstration program in v9N1P15) neither flag 4 nor flag 9 is set at fast mode entry. Even though flag 9 is not set, the calculator goes into TRACE at fast mode exit. Neither INV Stflg 9 from the keyboard nor an INV STF 9 in the program will release the calculator from the TRACE mode. Only a RST from the keyboard or in the program will do.

If N = 5 through 9 various combinations of set and reset of flags 4 and 9 occur as defined in V6N8P3. The response is the same as with N = 4. The TRACE mode persists at exit from fast mode, independent of the status of flag 9, and can only be cleared with a RST. Clearly, some other effect is causing the TRACE mode to occur. Since the effect does not occur with an N of less than 4, it seems likely that the result is somehow related to the 4's digit of the value of N.

Neither George nor I pretend to understand all of this. It does seem that the use of an N of 1 is to be preferred. Flag 4 is set, but flag 9 is not, so the user does not need to include an INV STF 9 in his program in order to avoid TRACE mode at fast mode exit. In addition, as discussed elsewhere in this issue, there is a very efficient way to make the entry constant which sets flag 4 and enters fast mode at location 001.

AN UNUSUAL TI-66 ERROR STATE - Dave Leising. If the program counter is set to the last step in the partition, and an SST is done from the keyboard no error state is indicated. But, now the TI-66 will not execute any keyboard command requiring a numeric address or an operand. A user-defined label command will not work either. Any such attempt displays an error indication, but also clears the error condition and subsequent operation is normal. Resetting the program pointer within bounds will clear the condition. CLR will not. The list of operations which will not work in this mode include:

- All register operations except CMs and CSR.
- GTO, SBR and all memory operations except CP.
- User defined labels A through E'.
- All Op commands.
- All flag operations.

Arithmetic, trigonometry, and register operations called by statistics routines seem to work properly.

TI-66 ERROR WHEN SUBROUTINE RETURN REGISTER IS FULL - Donald Wisander noticed that with the TI-66 the calculator stops and displays an error state if an additional subroutine is called with the return register full. This is consistent with page B-3 of the manual.

That reaction is different from that of the TI-58/59 where calling a subroutine beyond the sixth level does not cause an error indication, and does not store a new return address in the subroutine register. A RTN encountered at the seventh subroutine level causes the program to go to the return address stored for the sixth subroutine level. Strange results may bedevil the unsuspecting programmer who encounters this condition. A caution appears on page D-2 of Personal Programming. The effect was discussed in V6N6/7P30 and V6N9/10P9. A sample problem similar to that in V6N9/10P9 will illustrate the differences in response of the TI-66 and TI-58C/59:

```
LBL A  1  STO 00  LBL B  E  OP 20  B  LBL E  RCL 00  Pause  RTN
```

Press A on the TI-66. The display will count to 6 and stop with "Error" in the display. Press A on the TI-58C/59. The display will count to 7, flash the 7 six more times as the subroutine return register is cleared by the RTN's, and stop with a 7 in the display when the RTN is encountered with no return address available. No error condition is displayed.

The error indication with the TI-66 is a real improvement. The difference is not listed in the notes to the TI-58/58C/59 user in Appendix F of the TI-66 manual.

INSIDE THE TI-66 - D. Leising and K. Ward. We took one of our machines completely apart and made some interesting discoveries. The unit contains only two chips--the processor and a standard 1K x 4 CMOS RAM chip which is available from several manufacturers at a cost of only about six dollars each. It would be a very trivial matter for those with the knowhow to modify the TI-66 to receive homemade modules containing one of these chips, a few resistors and diodes, and two small batteries. The end result would be nonvolatile "CRAMS" like those which were going to be provided for the TI-88. I am going to do this and will furnish a schematic as soon as it is done. Perhaps one of the small companies selling TI-59 accessories would be interested in furnishing this modification to our members and others interested.

We also found that until fairly late in the design of the TI-66 there was going to be another key on the keyboard, located above the equals key. There is a cutout in the front cover for this key, and even a contact pad on the membrane, but no crosspoints on the circuit board.

TI-66 DATA REGISTER ARCHITECTURE - Dave Leising.

```
+--------+--------+--------+--------+--------+--------+--------+--------+
!  A  B  !  C  D  !  E  F  !  G  H  !  J  K  !  L  M  !  N  P  !  Q  R  !
+--------+--------+--------+--------+--------+--------+--------+--------+
```

Each data register in the TI-66 consists of eight bytes (sixteen nibbles) as identified from left to right in the figure above. The function of each nibble is indicated in the table at the right. A number of less than 13 digits can be placed anywhere in the mantissa field by synthetic programming and will be correctly interpreted by the ALU; however, any arithmetic operation on the register justifies the mantissa to the left.

Nibble	Function
A	Sign Control
B	Mantissa MSD
P	Mantissa LSD
Q	Exponent MSD
R	Exponent LSD

The sign control nibble logic is shown at the right. Bits 3 and 4 are "don't cares" during a read operation on the register. These bits are set to zero during any arithmetic operation on the register.

The exponent field is nibbles Q and R. An exponent value of 25 will appear as a R/S (code 25) in the last step of the equivalent program octet.

BITS				SIGN SIGNIFICANCE	
1	2	3	4	Mantissa	Exponent
0	0	X	X	+	+
0	1	X	X	-	+
1	0	X	X	+	-
1	1	X	X	-	-

NEWCOMER'S CORNER - THE GREAT CALENDAR RACE - Palmer Hanson

Several of the newer members have asked for some historical notes. For this issue I will discuss the calendar printing competition which started in the May 1978 issue of 52 Notes. The earliest TI-59 calendar program ran at about thirty minutes per year. But with such TI-59 giants as Lou Cargile, Jared Weinberger, Maurice Swinnen, Bill Skillman, and Richard Vanderburgh participating the time to print a full year was rapidly reduced. A 2 minutes 38 second program by Panos Galidas was published in the September 1978 issue of 52 Notes.

In early 1980 Maurice Swinnen for TI PPC Notes and Richard Nelson for the PPC Calculator Journal agreed on a set of competitions to demonstrate the relative capabilities of the TI and HP machines, as Maurice put it in V5N4/5P1, "now that the RPN programmers have a more seaworthy machine in the HP-41C." The calendar contest was resurrected. The TI-59 execution time for a full year still stood at 2 minutes 38 seconds. The best HP-41 time was reported to be in the six minute range. The competition was about to heat up. Roger Hill responded with an HP-41 program which would print a one year calendar in 2 minutes 18 seconds. The July/August 1980 issue of the PPC Calculator Journal published Roger's program and claimed the lead. But in mid-summer I had adapted Panos Galidas' program for use with the Martin Neef fast mode technique, yielding a one year printout in 1 minute 32 seconds. V5N7P7 of TI PPC Notes published those results in the first week of September, reclaiming the lead for the TI-59. Roger Hill responded quickly with another chapter of what he called the "~~all/out/war~~ friendly competition". The October 1980 issue of the PPC Calculator Journal V7N8P15) published his new HP-41 program with a printing time of only 1 minute 14 seconds -- very close to the theoretical limit for that device. Using ideas from Richard Snow the TI-59 execution time was reduced to 1 minute 26 seconds (V5N8P4), but the TI users had run into a version of the existence theorem: "If you doubt that a faster program can be witten, you won't try to write it."

The emphasis on speed for the TI-59 switched to the speedy factor finder problem, and there has been no more calendar effort reported in TI PPC Notes. A fresh attack on the calendar problem came from Patrick Acosta, who had already developed the fastest TI-59 Speedy Factor Finder using the Pgm-02-SBR-239 method of fast mode entry (V6N4/5P13). In July 1981 I received a calendar printing program which used the h12 method of fast mode entry. This was several months before publication of the landmark article on other methods of entering fast mode in V6N8P3/4. The use of the h12 technique permitted use of the ML-20 program for some of the preliminary calculations, releasing additional memory for program enhancement. The result was a program which will print the worst case calendar year (2000) in 1 minute 23 seconds, the best TI-59 time to date.

Of course some readers will wonder at all this discussion of calendars. As Maurice Swinnen observed in V7N7P7:

> "One of the members once admonished me not to run 'silly' calendar contests. 'I can get all the calendars I want at the bank' he wrote me. He obviously missed the point by more than a mile! The calendar is only the vehicle by which we try to get more out of our calculators."

A similar comment applies to the running of other programming problems. Elsewhwere in this issue you will find the results of the calendar printing problem applied to other calculators and computers.

A ONE MINUTE 23 SECOND CALENDAR - Patrick Acosta

Editor's Note: Patrick's original program included a demonstration of the RTN technique for leaving fast mode. Our common failure to remember the effect of setting flag 9 resulted in our deleting of the RTN technique to avoid the TRACE effect. Those problems led to the comment about returning in trace mode in V6N8P4. The program presented here has used a revised fast mode entry constant to re-establish the RTN technique.

User Instructions:

1. Store the program using the program listing from page 8 and the constants from the table at the right side of this page. Be sure to enter all thirteen digits of the constants where indicated. Record the four card sides.

2. To initialize the program enter all four card sides. Then generate the h12 at location 136 with the following keyboard sequence. Do not clear the flashing displays as they occur.

Keystrokes	Display
10-Op-17	159.99
GTO-136-CLR	0
Pgm-12-SBR-999	Flashing 0.
R/S	Flashing 0.00
DMS	Flashing 0
LRN	136 43
Ins	136 43
Ins	136.43
LRN-RST-CLR	0

Note that two inserts were required since the ones digit of the command at location 136 of the firmware was less than 4. You do not need to return the partitioning. The program will do that for you.

3. Enter the starting year (YYYY) and press A. See "YYYY." returned in the display.

4. Enter the number of months to be printed. For a two year printout you enter 24. Press R/S. See a "0." in the display.

5. Enter the starting month (1 to 12) and press R/S. After a few seconds the printout will begin. When the printout is complete the calculator will stop with a "0." in the display. Since the printout occurs in "fast mode" operation cannot be interrupted, even with R/S or RST.

6. For other calendars iterate steps 3 through 5.

0.	00
0.	01
0.	02
0.	03
0.	04
0.	05
0.	06
10.00000251331	07
8.000000211714	08
10.00000301335	09
9.000000133335	10
10.00000301345	11
9.000025413117	12
10.00025412745	13
10.00000134122	14
[illegible]	15
10.00000321537	16
9.000000313242	17
10.00000161715	18
1.000036000030	19
1.014300003700	20
1.000021000036	21
98.99000000000	22
2.010000000000	23
2.010000000000	24
2.010000000000	25
2.010000000000	26
2.010000000000	27
2.010000000002	28
2.010002000003	29
2.010003000004	30
2.010004000005	31
2.010005000006	32
2.010006000007	33
2.010007000010	34
2.010010000011	35
2.010011000012	36
2.010012000201	37
2.010201000202	38
2.010202000203	39
2.010203000204	40
2.010204000205	41
2.010205000206	42
2.010206000207	43
2.010207000210	44
2.010210000211	45
2.010211000212	46
2.010212000301	47
2.010301000302	48
2.010302000303	49
2.010303000304	50
2.010304000305	51
2.010305000306	52
2.010306000307	53
2.010307000310	54
2.010310000311	55
2.010311000312	56
2.010312000401	57
2.010401000402	58
1.000000000002	59
6.010311000000	60
6.010312000000	61
6.010401000000	62
6.010402000000	63
7.000000211714	64
0.	65
0.	66
0.	67
0.	68
0.	69

A One Minute 23 Second Calendar - (cont)

Program Listing - Banks 1 and 2

000	61	GTO
001	03	03
002	24	24
003	68	NOP
004	68	NOP
005	44	SUM
006	00	00
007	61	GTO
008	03	03
009	18	18
010	61	GTO
011	03	03
012	18	18
013	68	NOP
014	68	NOP
015	44	SUM
016	00	00
017	61	GTO
018	03	03
019	07	07
020	44	SUM
021	00	00
022	61	GTO
023	02	02
024	96	96
025	61	GTO
026	02	02
027	96	96
028	68	NOP
029	68	NOP
030	44	SUM
031	00	00
032	61	GTO
033	02	02
034	89	89
035	76	LBL
036	11	A
037	42	STO
038	03	03
039	42	STO
040	05	05
041	91	R/S
042	42	STO
043	06	06
044	07	7
045	69	OP
046	17	17
047	25	CLR
048	42	STO
049	02	02
050	91	R/S
051	42	STO
052	01	01
053	36	PGM
054	20	20
055	71	SBR
056	00	00
057	91	91
058	32	X:T
059	12	B
060	98	ADV
061	91	R/S
062	32	X:T
063	75	-
064	53	(
065	24	CE
066	55	÷
067	07	7
068	54	)
069	59	INT
070	65	×
071	07	7
072	95	=
073	42	STO
074	02	02
075	06	6
076	44	SUM
077	01	01
078	29	CP
079	43	RCL
080	64	64
081	42	STO
082	08	08
083	43	RCL
084	05	05
085	55	÷
086	04	4
087	55	÷
088	22	INV
089	59	INT
090	22	INV
091	67	EQ
092	03	03
093	91	91
094	69	OP
095	28	28
096	02	2
097	05	5
098	55	÷
099	22	INV
100	59	INT
101	22	INV
102	67	EQ
103	03	03
104	91	91
105	04	4
106	95	=
107	22	INV
108	59	INT
109	67	EQ
110	03	03
111	92	92
112	69	OP
113	38	38
114	61	GTO
115	03	03
116	92	92
117	69	OP
118	25	25
119	07	7
120	42	STO
121	01	01
122	61	GTO
123	00	00
124	78	78
125	98	ADV
126	25	CLR
127	92	RTN
128	76	LBL
129	12	B
130	58	FIX
131	00	00
132	60	DEG
133	43	RCL
134	59	59
135	86	STF
136	92	92
137	84	OP*
138	24	24
139	45	Y^x
140	38	SIN
141	88	DMS
142	69	OP
143	71	71
144	73	RC*
145	00	00
146	82	HIR
147	05	05
148	44	SUM
149	00	00
150	73	RC*
151	00	00
152	82	HIR
153	06	06
154	43	RCL
155	22	22
156	82	HIR
157	36	36
158	68	NOP
159	68	NOP

AFTER INITIALIZATION ⇓

128	76	LBL
129	12	B
130	58	FIX
131	00	00
132	60	DEG
133	43	RCL
134	59	59
135	86	STF
136	12	12
137	68	NOP
138	61	GTO
139	00	00
140	62	62
141	68	NOP
142	68	NOP
143	68	NOP
144	70	RAD
145	71	SBR
146	73	RC*
147	00	0
148	82	HIR
149	05	05
150	44	SUM
151	00	00
152	73	RC*
153	00	00
154	82	HIR
155	06	06
156	43	RCL
157	22	22
158	82	HIR
159	36	36

160	69	OP
161	20	20
162	73	RC*
163	00	00
164	44	SUM
165	00	00
166	22	INV
167	59	INT
168	82	HIR
169	07	07
170	73	RC*
171	00	00
172	82	HIR
173	08	08
174	69	OP
175	05	05
176	44	SUM
177	00	00
178	73	RC*
179	00	00
180	82	HIR
181	05	05
182	44	SUM
183	00	00
184	73	RC*
185	00	00
186	82	HIR
187	06	06
188	43	RCL
189	22	22
190	82	HIR
191	36	36
192	69	OP
193	20	20
194	73	RC*
195	00	00
196	44	SUM
197	00	00
198	22	INV
199	59	INT
200	82	HIR
201	07	07
202	73	RC*
203	00	00
204	82	HIR
205	08	08
206	69	OP
207	05	05
208	44	SUM
209	00	00
210	73	RC*
211	00	00
212	82	HIR
213	05	05
214	44	SUM
215	00	00
216	73	RC*
217	00	00
218	82	HIR
219	06	06
220	43	RCL
221	22	22
222	82	HIR
223	36	36
224	69	OP
225	20	20
226	73	RC*
227	00	00
228	44	SUM
229	00	00
230	22	INV
231	59	INT
232	82	HIR
233	07	07
234	73	RC*
235	00	00
236	82	HIR
237	08	08
238	69	OP
239	05	05
240	44	SUM
241	00	00
242	73	RC*
243	00	00
244	82	HIR
245	05	05
246	44	SUM
247	00	00
248	73	RC*
249	00	00
250	82	HIR
251	06	06
252	43	RCL
253	22	22
254	82	HIR
255	36	36
256	69	OP
257	20	20
258	73	RC*
259	00	00
260	44	SUM
261	00	00
262	22	INV
263	59	INT
264	82	HIR
265	07	07
266	73	RC*
267	00	00
268	82	HIR
269	08	08
270	69	OP
271	05	05
272	97	DSZ
273	04	04
274	02	02
275	40	40
276	43	RCL
277	02	02
278	44	SUM
279	00	00
280	42	STO
281	65	65
282	05	5
283	49	PRD
284	65	65
285	69	OP
286	00	00
287	83	GO*
288	65	65
289	73	RC*
290	00	00
291	22	INV
292	44	SUM
293	00	00
294	82	HIR
295	08	08
296	73	RC*
297	00	00
298	22	INV
299	44	SUM
300	00	00
301	22	INV
302	59	INT
303	82	HIR
304	07	07
305	69	OP
306	20	20
307	73	RC*
308	00	00
309	22	INV
310	44	SUM
311	00	00
312	82	HIR
313	06	06
314	43	RCL
315	22	22
316	82	HIR
317	36	36
318	73	RC*
319	00	00
320	82	HIR
321	05	05
322	69	OP
323	05	05
324	22	INV
325	97	DSZ
326	06	06
327	01	01
328	25	25
329	43	RCL
330	01	01
331	67	EQ
332	01	01
333	17	17
334	69	OP
335	21	21
336	02	2
337	09	9
338	42	STO
339	00	00
340	98	ADV
341	43	RCL
342	02	02
343	22	INV
344	44	SUM
345	00	00
346	85	+
347	73	RC*
348	01	01
349	82	HIR
350	08	08
351	59	INT
352	75	-
353	53	(
354	24	CE
355	55	÷
356	07	7
357	54	)
358	59	INT
359	42	STO
360	04	04
361	65	×
362	07	7
363	95	=
364	42	STO
365	02	02
366	43	RCL
367	05	05
368	69	OP
369	06	06
370	43	RCL
371	19	19
372	82	HIR
373	05	05
374	43	RCL
375	20	20
376	82	HIR
377	07	07
378	22	INV
379	59	INT
380	82	HIR
381	06	06
382	43	RCL
383	21	21
384	82	HIR
385	08	08
386	69	OP
387	05	05
388	61	GTO
389	01	01
390	46	46
391	25	CLR
392	01	1
393	08	8
394	32	X:T
395	61	GTO
396	03	03
397	36	36
398	00	0
399	00	0

A CALENDAR PRINTING PROGRAM FOR THE TI-66/PC-200 - Dave Leising

Due to the 16 column limitation of the PC-200 a calendar in the conventional US format cannot be obtained. Therefore, the European format with weeks aligned vertically was used. Extensive utilization of digit-serial operations on registers was necessary to overcome the memory limitations of the TI-66. As a result the program is very slow. A one year printout requires one hour and forty-six seconds.

The program listing appears on the next page. Tricks to watch out for when entering the program are: (1) the short form addressing used at locations 008, 029, 067, 073, 079 and 435 (see V9N1P8 for a discussion of short form addressing); and (2) the 13-digit constants in registers 07 and 08 which are:

Register 07 0.4243434443434

Register 08 0.2012212421032

The program runs in 439.08 partitioning. To execute the program key in a year YYYY and press A. Then sit back and wait. A sample full-size one year calendar assembled from the PC-200 printout follows.

```
     1.1984            4.1984            7.1984            10.1984
S  1  8 15 22 29   S  1  8 15 22 29   S  1  8 15 22 29   S     7 14 21 28
M  2  9 16 23 30   M  2  9 16 23 30   M  2  9 16 23 30   M  1  8 15 22 29
T  3 10 17 24 31   T  3 10 17 24      T  3 10 17 24 31   T  2  9 16 23 30
W  4 11 18 25      W  4 11 18 25      W  4 11 18 25      W  3 10 17 24 31
T  5 12 19 26      T  5 12 19 26      T  5 12 19 26      T  4 11 18 25
F  6 13 20 27      F  6 13 20 27      F  6 13 20 27      F  5 12 19 26
S  7 14 21 28      S  7 14 21 28      S  7 14 21 28      S  6 13 20 27

     2.1984            5.1984            8.1984            11.1984
S     5 12 19 26   S     6 13 20 27   S     5 12 19 26   S     4 11 18 25
M     6 13 20 27   M     7 14 21 28   M     6 13 20 27   M     5 12 19 26
T     7 14 21 28   T  1  8 15 22 29   T     7 14 21 28   T     6 13 20 27
W  1  8 15 22 29   W  2  9 16 23 30   W  1  8 15 22 29   W     7 14 21 28
T  2  9 16 23      T  3 10 17 24 31   T  2  9 16 23 30   T  1  8 15 22 29
F  3 10 17 24      F  4 11 18 25      F  3 10 17 24 31   F  2  9 16 23 30
S  4 11 18 25      S  5 12 19 26      S  4 11 18 25      S  3 10 17 24

     3.1984            6.1984            9.1984            12.1984
S     4 11 18 25   S     3 10 17 24   S  2  9 16 23 30   S  2  9 16 23 30
M     5 12 19 26   M     4 11 18 25   M  3 10 17 24      M  3 10 17 24 31
T     6 13 20 27   T     5 12 19 26   T  4 11 18 25      T  4 11 18 25
W     7 14 21 28   W     6 13 20 27   W  5 12 19 26      W  5 12 19 26
T  1  8 15 22 29   T     7 14 21 28   T  6 13 20 27      T  6 13 20 27
F  2  9 16 23 30   F  1  8 15 22 29   F  7 14 21 28      F  7 14 21 28
S  3 10 17 24 31   S  2  9 16 23 30   S1 8 15 22 29      S1 8 15 22 29
```

HELP WANTED - Information as to where I might obtain the Math/ Utilities and EE modules, and the Programming Aids Specialty Pakette. Write to:

Ken Farr
732 Garden City Drive
Monroeville, PA 15146

A Calendar Printing Program for the TI-66/PC-200 - (cont)

Program Listing:

000	LBL	075	D	150	STO	225	INTG	300	-	375	E'
001	A'	076	X=T	151	03	226	INV	301	X⇄T	376	D'
002	D'	077	84	152	RCL	227	SUM	302	=	377	A'
003	INV	078	.	153	04	228	04	303	STO	378	D'
004	STF	079	2	154	÷	229	×	304	05	379	CLR
005	01	080	5	155	C	230	RETN	305	OP	380	E'
006	CP	081	D	156	1	231	(	306	35	381	A'
007	X=T	082	X=T	157	0	232	.	307	OP	382	E
008	13	083	89	158	0	233	4	308	25	383	OP
009	STF	084	.	159	=	234	×	309	CLR	384	01
010	01	085	0	160	X⇄T	235	RCL	310	STO	385	A'
011	C'	086	1	161	RCL	236	04	311	03	386	B'
012	LBL	087	+/-	162	04	237	+	312	6	387	CLR
013	CLR	088	LBL	163	3	238	2	313	STO	388	E'
014	E'	089	CLR	164	6	239	.	314	04	389	E
015	RETN	090	STO	165	5	240	3	315	RCL	390	OP
016	LBL	091	02	166	×	241	)	316	05	391	02
017	B'	092	RCL	167	RCL	242	INTG	317	STO	392	A'
018	D'	093	07	168	03	243	+/-	318	06	393	B'
019	C'	094	SUM	169	+	244	+	319	RCL	394	CLR
020	E'	095	02	170	X⇄T	245	(	320	06	395	E'
021	RETN	096	4	171	+	246	RCL	321	X⇄T	396	B'
022	LBL	097	INV	172	3	247	03	322	RCL	397	E
023	C'	098	LOG	173	1	248	÷	323	02	398	OP
024	X⇄T	099	INV	174	×	249	4	324	INTG	399	03
025	8	100	PROD	175	RCL	250	)	325	INV	400	B'
026	X⇄T	101	00	176	04	251	INTG	326	X≥T	401	CLR
027	INV	102	RCL	177	-	252	-	327	03	402	E'
028	X≥T	103	08	178	3	253	(	328	37	403	A'
029	32	104	STO	179	1	254	.	329	X⇄T	404	D'
030	+	105	01	180	+	255	7	330	CP	405	CP
031	2	106	1	181	3	256	5	331	INV	406	X=T
032	+	107	0	182	X⇄T	257	+	332	X≥T	407	04
033	4	108	PROD	183	RCL	258	(	333	03	408	14
034	0	109	02	184	04	259	RCL	334	37	409	C'
035	=	110	RCL	185	X≥T	260	03	335	X⇄T	410	E'
036	RETN	111	02	186	02	261	÷	336	LBL	411	GTO
037	LBL	112	INTG	187	31	262	1	337	CP	412	04
038	E'	113	INV	188	OP	263	0	338	X⇄T	413	23
039	X⇄T	114	SUM	189	33	264	0	339	×	414	INV
040	1	115	02	190	GTO	265	)	340	.	415	IFF
041	0	116	CP	191	02	266	INTG	341	0	416	01
042	0	117	X=T	192	45	267	×	342	1	417	04
043	PROD	118	56	193	LBL	268	.	343	PROD	418	21
044	04	119	+	194	D'	269	7	344	03	419	C'
045	CLR	120	2	195	1	270	5	345	=	420	LBL
046	X⇄T	121	7	196	0	271	)	346	SUM	421	CLR
047	SUM	122	=	197	PROD	272	INTG	347	03	422	E'
048	04	123	SUM	198	03	273	=	348	7	423	E
049	RETN	124	02	199	RCL	274	STO	349	INV	424	OP
050	LBL	125	X⇄T	200	03	275	04	350	SUM	425	04
051	E	126	OP	201	INTG	276	+/-	351	06	426	OP
052	CLR	127	20	202	INV	277	÷	352	DSZ	427	05
053	EXC	128	RCL	203	SUM	278	7	353	04	428	GTO
054	04	129	00	204	03	279	=	354	03	429	03
055	RETN	130	ADV	205	RETN	280	INTG	355	19	430	07
056	ADV	131	FIX	206	LBL	281	×	356	E	431	RCL
057	ADV	132	04	207	D	282	7	357	1	432	02
058	ADV	133	PRINT	208	×	283	+	358	0	433	INTG
059	R/S	134	INV	209	RCL	284	RCL	359	0	434	INV
060	LBL	135	FIX	210	00	285	04	360	PROD	435	SUM
061	A	136	INV	211	=	286	=	361	01	436	02
062	STO	137	INTG	212	INV	287	CP	362	RCL	437	GTO
063	00	138	+	213	INTG	288	INV	363	01	438	01
064	.	139	X⇄T	214	CP	289	X=T	364	INTG	439	02
065	0	140	+	215	RETN	290	02	365	CP		
066	0	141	RCL	216	LBL	291	93	366	X=T		
067	2	142	00	217	C	292	7	367	04		
068	5	143	INTG	218	1	293	-	368	31		
069	D	144	×	219	0	294	1	369	INV		
070	X=T	145	C	220	0	295	=	370	SUM		
071	89	146	4	221	=	296	X⇄T	371	01		
072	.	147	INV	222	STO	297	RCL	372	+		
073	0	148	LOG	223	04	298	02	373	3		
074	1	149	=	224	INV	299	INTG	374	=		

0.	00
0.	01
0.	02
0.	03
0.	04
0.	05
0.	06
.42434344	07
.20122124	08

ANOTHER TEST OF PRECISION - Fred Gruenberber's "Computer Recreations" in the April 1984 issue of Scientific American states: "If you have a calculator with a key for squaring a number, try this: enter the number 1.0000001 and press the square key 27 times. The procedure is equivalent to raising the initial number to the 134,217,728th power. ... The problem is designed to reveal the precision level of the machine." A table is presented for an assortment of calculators and computers. None of the machines listed got even 7 digits correct.

Editor"s Note: For nearly every machine there are several ways to solve the problem. For example, in BASIC language the problem may be solved as:

Method A

```
10 A = 1.0000001
20 FOR I = 1 TO 27
30 A = A^2
40 NEXT I
50 PRINT A
60 END
```

Method B

```
10 A = 1.0000001
20 FOR I = 1 TO 27
30 A = A*A
40 NEXT I
50 PRINT A
60 END
```

Method C

```
10 A = 1.0000001
20 B = 134217728
30 A = A^B
40 PRINT A
50 END
```

On most calculators the equivalent of methods A and B will yield identical results, but the y^x function which is equivalent to method C will return a different, and usually more accurate result. On many computers the results vary widely with method; for example, the BASIC on the Commodore 64 and Apple II+. Results from the three methods from a representative set of machines are:

Machine	Method A	Method B	Method C
Exact	674530.4707...	674630.4707...	674530.4707...
Model 100	674529.41305068	674529.41305068	674530.47074049
TI-66	674520.6067381	674520.6067381	674530.4707400
HP-11, etc.	674494.0561	674494.0561	674530.4707
CC-40	674530.31804225	674530.31804225	674621.4634954
BA-55	674432.82060	674432.82060	674530.92317
TI-57	674432.8204	674432.8204	674530.9232
TI-59	674520.6052712	674520.6052712	674530.9234109
Color Computer	713658.879	643571.305	665348.188
C-64/Apple	728339.418	22723.9709	665348.189
TI-55II/57LCD	674432.8206	674432.8206	660003.2248

POWERS & ROOTS FOR TI-66 AND TI-58/59 - Myer Boland discovered that the TI-66 obtains values for powers and roots which are slightly different from those obtained from the TI-58C/59. A single example will illustrate the differences. Key in the following sequence:

2 Y^x 3 = INV INT

Both the TI-66 and TI-59 calculators will display an 8 at the equals sign. The TI-59 will display 1. -12 at the INV INT. The TI-66 will display a one. The difference is that at the equal sign the TI-59 value is slightly greater than eight, while at the same point the TI-66 value is slightly less by 1.2 -11.

SIN(A) = COS(90-A) ON THE TI-66 - Palmer Hanson. I have found that the TI-66 delivers identical values for sin(A) and cos(90-A). That is not the case for the TI-58C/59, a phenomenon that was discussed at length on V8N3P15 and on page C-1 of Personal Programming. The TI-55II and the TI-57LCD show the same consistency between sin(A) and cos(90-A) as the TI-66.

MORE TRIGONOMETRIC ANOMALIES FOR THE TI-66 - Palmer Hanson. In V2N5P4 of 52 Notes Karl Hoppe reported trigonometric anomalies in the SR-52 for very small arguments:

> "... attempts to take the sin, cos, or tan of a number whose absolute value is between zero and 3.6 D-97 in degree mode (or 6.283185307180 D-99 in radian mode) result in an error condition. This may be due to the creation of underflowed intermediate results by the trig firmware."

The same anomalies occur with the TI-58C/TI-59. My tests show that for the argument 3.6E-97 in the degree mode, or for 6.283185307180E-99 = (2π)E-99 in radian mode, either the sine or the tangent return the value (2π)E-99 and the cosine returns a "1", all without an error indication. With either input argument reduced by 1 in the least significant digit, that is 3.599999999999E-97 in degree mode or 6.283185307179E-99 in radian mode, the sine or tangent still return the value (2π)E-99 and the cosine returns a "1", but now the display is flashing. Exactly the same results including the flashing display will occur for smaller arguments all the way down to 1EE-99.

The TI-66 has a similar anomaly. In degree mode with an argument of 4.5E-97 the sine and tangent will return the value ($\pi/4$)E-98 . With any smaller argument the sine and tangent return an "Error" when called from the keyboard. In a program with the smaller arguments the sine and tangent return 0. 00 with an error indication, but program execution does not stop. The cosine function on the TI-66 returns a value of one with no error for even the smallest possible argument of 1E-99.

My HP-11 does not generate error conditions when the trigonometric functions are called with very small arguments. With the argument ($180/\pi$) E-99 and any smaller arguments the sine and tangent functions return a zero. With an argument of ($180/\pi$ + 0.00000001)E-99 the sine and tangent return 1E-99.

ROBERT PRINS BRAINTEASER - Starting from the turnon condition you are to create a flashing one (1) in the display. The last key you press from the keyboard should be CLR, but you are not allowed to use GTO or SBR before it. To make things a little harder, the only functions or keys you may use are 2nd, INV, CLR, LRN, RCL, STO, SUM, PRD, GTO, and R/S, but you may use R/S only once.

Editor's Note: Robert added "The problem has a solution. You can find it at the end of this letter". But at the end of the letter he wrote "I am not going to give you the solution before the problem has been published". So far my only solution involves some "sea-lawyering" as to the definition of "before". Happy puzzling.

PRINTER PAPER - Lem Matteson writes: "When I started through my stack of old programs I found out why you had asked about sources of good printing paper. I found that many of my newer program listings had sections that had faded out. Late in 1979 I had bought ten packs of TI paper for $70.00. That lasted me a long time. Programs recorded on that paper are brownish yellow now but the printing has not faded at all ..."

Editor's Note: My experience has been the same. Printouts obtained in 1979/1980 with old style paper which had a brownish cast are still readable, and the paper has not been attacked by the glue used to attach the printouts to program forms. But some newer paper which showed somewhat better contrast at the time of printing has faded in a year or so, and the glue seems to have caused discoloration. That is why I was so pleased to find a source of the old style paper (V9N1P2).

CARD READER CLEANING - Paul Sperry reports that use of the CCL-144 cleaning strip was successful in clearing problems with reading and writing magnetic cards. He identified the solvent as alcohol. He forwarded a brochure from Texwipe in which a solution of 91% isopropyl alcohol and 9% ionized water is used for cleaning magnetic components. Paul and others have suggested that the cleaning strip may be reused by supplying your own solution.

Editors Note: So far I have provided seventeen cleaning strips to members who reported magnetic card read/write problems. I have had no reports of failure to clear the problem. At the $2.00 member price that is a low cost repair.

PRINT HEAD CLEANING - Palmer Hanson. I have one PC-100 which has had long-standing print head problems. No amount of cleaning using the procedure on page VI-12 of Personal Programming or the one in V5N3P3 would clear the problem. I had tried running cleaning cards saturated in alcohol through the printer with the ADV key with little success. One day when I was particularly frustrated with the problem I remembered the "burnishing" effect claimed for the CCL-144 cleaning card. I found a piece of crocus cloth, cut a strip to the width of the printer paper, and passed it through the printer using the ADV key. No more print head problems!!! I would need more experience with this process before recommending it as a general cure-all. I discussed this with Maurice Swinnen. He says others have had success with a strip of the material used for polishing defects in Plexiglas. He also indicated that HP sells a cleaning card for use with their printers. I will report further developments in this area in future issues.

A TAPE DRIVE FOR THE CC-40 ?

An article on page 10 of the February 27, 1984 issue of Electronic News reports that

> "Texas Instruments is contemplating a second effort at integrating Entrepo's Wafer Tape Drive systems into its CC-40 LCD-style portable computer. ..."

The "Inside Track" column on page 112 of the March 26, 1984 issue of Infoworld says we should

> "Look for TI to bring out a new version of its hand-held computer. This time it will work with cassettes rather than the defunct wafer tape it promised buyers. ..."

I have received no releases from TI. We will just have to wait and see.

BACK ISSUES OF 52-NOTES - Richard Vanderburgh reports that he will continue to provide copies of back issues of 52 Notes at a price of $1.50 each in the US, and $2.00 each abroad. Write to:

Richard Vanderburgh
9459 Taylorsville Road
Huber Heights OH 45424

The issues from Volume 2 Number 6 through Volume 4 Number 3, a total of 22 issues and 130 typewritten pages, are of primary interest to 58/59 users. The earlier issues make interesting reading, and some of the material is applicable to the TI-58C/59; e.g., the trigonometric anomaly for small arguments discussed on page 12 was reported for the SR-52 in Volume 2 Number 5, just one issue before coverage of the TI-59 started. The anomaly was not discussed relative to the TI-59 in subsequent issues of 52 Notes, and has not been previously discussed in TI PPC Notes.

SOLID STATE SOFTWARE MODULE AND MAGNETIC CARD AVAILABILITY

In V9N2P4 member J. M. Gallego offered magnetic cards and Solid State Software modules for sale. He was able to obtain some more material. His inventory as of April 1 was:

- 149 Boxes of 40 Blank Magnetic Cards with Carrying Case
- 10 Navigation modules
- 12 Aviation modules
- 5 Leisure modules
- 6 Real Estate and Investment modules
- 15 Securities Analysis modules

He will sell them for sixteen dollars ($16.00) for each module, and eight dollars ($8.00) for each box of magnetic cards while they last. Shipping is included. U. S. members should send money orders only to:

Q. Jose M. Gallego
250 Quintard Avenue, Apt. 96
Chula Vista CA 92011/4924.

Members from other countries should write to make appropriate arrangements.

MULTIPLICATION MAY NOT BE COMMUTATIVE WITH THE TI-59 - George Thomson

This quirk is described in W. Kahan's paper "Mathematics Written in Sand ..." at the foot of page 14 in Proceedings of the Statistical Computing Section, American Statistical Association, 1983, pp. 12-26:

> "Multiplication is neither commutative nor monotonic on the TI-59. Try e x π - π x e = "

You will see 2.8E-11 in the display, not the expected zero. The problem is not unique to that pair of values. Kahan did not further elaborate so I made a brief investigation and discovered what was going on but not the reason. Keep in mind that multiplier x multiplicand gives the product. The exact products below were obtained by hand calculation from the stored data.

> RULE: When two 13-digit numbers, A and B, are multiplied on the TI-59 the 13-digit output will be within a few units in the 13th digit of the exact answer to the product of A and B', where B' is equal to the multiplicand (B) truncated to 12-digits. All 13-digits of the multiplier are used.

In other words, in most cases the result corresponds to multiplying together a 13-digit number by a twelve-digit number. If we calculate A x B and B x A then either one or both or neither of the two answers will be correct to within a unit in the 13th-digit. Try this easy example:

2 SQRT STO 02	1.4142 1386 2373	A in Register 02
- 3 EE 12 +/- = STO 12	1.4142 1386 2370	A' in Register 12
3 SQRT STO 03	1.7320 5080 7568	B in Register 03
- 8 EE 12 +/- = STO 13	1.7320 5080 7560	B' in Register 13

The exact products of the stored data as calculated by hand are

A x B =	2.4494 8974 2781 77...
A x B' =	2.4494 8974 2770 45...
B x A' =	2.4494 8974 2776 57...

On the TI-59: RCL 02 x RCL 03 = 2.4494 8974 2770 which is the same as A x B' but not A x B.

On the TI-59: RCL 03 x RCL 02 = 2.4494 8974 2776 which is the same as B x A' but not B x A.

Neither answer is the correct value of 2.4494 8974 2782 .

(For Kahan's e x π case, e x π is within 1 in the 13th place since the 13th digit of π on the TI-59 is a zero, but π x e is wrong.)

Multiplication May Not Be Commutative with the TI-59 - (cont)

While it is true that TI only promises 10-digit accuracy and calls the other three digits guard digits, small errors in arithmetic ramify in an alarming way in repetitive calculations such as matrix inversions. To be blunt, I think that this quirk plays hell with all of our discussions about squeezing high accuracy from the TI-59. At the moment we do not know the rules governing the multiplication process. The basic patent, U. S. Patent 4,153,937, May 8, 1979 is of no help since it refers all references to the arithmetic unit back to another TI patent: U.S. Patent 3,900,722, August 19, 1975...Maybe some electronically minded reader can tell us all about it.

Editors Note: George Thomson writes "...this quirk plays hell with all of our discussions about squeezing high accuracy from the TI-59." I am not quite so sure. I suspect that the TI-59 will still reign supreme where chain calculations are concerned. There is even a workaround which holds the promise of delivering better TI-59 performance than we have seen in the past.

Kahan also reported that on the HP-15C the elements of the inverse of the 8 x 8 Hilbert are good to "roughly three significant decimals". That is not very good with respect to the results from ML-02 on the TI-59 which are good to roughly 5 - 6 significant decimals. My tests show that multiplication is commutative on the HP-11. Consider the intermediate results for e x :

Exact Product		8.53973 42226 73986
TI-59	e x	8.53973 42226 73
TI-59	x e	8.53973 42226 45
HP-11	e x π = π x e	8.53973 4222

An error has already appeared in the least sigificant digit of the HP-11 solution -- it should have been a 3. Even the poorer of the two TI-59 results is 24 times closer to the exact result. As I noted in V8N2P3 TI-59 owners who want to make additional comparisons but do not have a ten digit RPN calculator can obtain the equivalent ten digit results through judicious use of EE-INV-EE to round intermediate results to the display and discard the guard digits. Thus the sequence

Pi EE INV EE x 1 INV Ln EE INV EE =

will yield the same result as the HP-11, and the result will be the same for e x π. The HP-41 results are typically the same as the HP-11.

This "non-commutative multiply" quirk seems to be inherent in all the TI calculators manufactured at about the same time as the TI-59 was released. The good news is that the quirk does not seem to be present in the later TI calculators which use LCD displays such as the TI-35, BA-55, TI-55II, TI-57LCD and TI-66. For the TI-66:

e x π = π x e = 8.53973 42226 61

The error is a factor of 56 smaller than the HP-11 error. Cursory checks show commutative multiplication on the CC-40, the Model 100 and the Color Computer.

A WORKAROUND FOR THE NON-COMMUTATIVE MULTIPLY ON THE TI-59 - Palmer Hanson

With many others I had not thought to test the commutative multiply property for the TI-59; but when working on the 13 Digit Speedy Factor Finder problem (V8N4P15) I had tested the divide process quite carefully. What if we were to replace the non-commutative multiply function with two divisions? In George Thomsons example with $\sqrt{2}$ and $\sqrt{3}$:

Exact product	2.4494 8974 2781 77 ...
RCL 02 x RCL 03 =	2.4494 8974 2770
RCL 03 x RCL 02 =	2.4494 8974 2776
RCL 02 / RCL 03 1/x =	2.4494 8974 2781
RCL 03 / RCL 02 1/X =	2.4494 8974 2781

Using the double divide for that problem the "multiplication" is commutative and correct in the truncated sense. For the e x π problem:

Exact Product	8.53973 42226 73986 ...
Pi / 1 INV Ln 1/x =	8.53973 42226 73
1 INV Ln / Pi 1/x =	8.53973 42226 74

where one answer is correct in the truncated sense and the other is correct in the rounded sense, and the results are not commutative by 1 in the thirteenth digit. So far I have not found a "product" using two divides which fails to be commutative by more than 2 in the thirteenth digit. In any case the double divide technique seems to substantially reduce the error in the "product".

If the double divide technique is really superior then an appropriate test might be George Thomson's 7 x 7 sub-Hilbert problem (V8N6P18). You will recall that I was bothered with the TI-59 results. The CC-40 solution was more than two orders of magnitude better, even though the documentation indicates that the CC-40 method and the TI-59 ML-02 method were the same. (Tests show that the ML-02 and CC-40 methods are not equivalent--that will be discussed in a later issue). But TI-59 solutions not using ML-02 did little better. Of course one must download ML-02 to replace multiplies with double divides. Instead I modified two of the other solutions for simultaneous equations, the Nick/Ristanovic "Gauss" method from V7N6P13, and Robert Prins' improvement on the Probrambiten method from V9N1P16. Each program had only two multiplies to replace with double divides. A CE was added after each 1/X to avoid error indications when a zero is involved. Then adjustment of the absolute addresses yielded working programs. The instructions for use are not changed except that the 9 Op 17 partitioning must be set from the keyboard, and the Nick/Ristanovic program requires a Cms from the keyboard before starting.

The results were nothing short of astounding. For the 7 x 7 sub-Hilbert problem the results from the programs using double divides were two orders of magnitude better than from the same programs with multiplies.

A Workaround for the Non-commutative Multiply on the TI-59 - (cont)

In the results presented in V8N6P19 I truncated the results at the level which defined the relative error. That led to confusion on the part of several readers. In the following table all digits of the results are presented.

	With Normal Multiply		With Double Divide	
Exact Solution	Prins/ Programbiten	Nick/ Ristanovic	Prins/ Programbiten	Nick/ Ristanovic
56.	55.92331553138	56.00755936628	55.99920734492	56.00006963441
-1512.	-1510.227621269	-1512.173201797	-1511.980680403	-1512.000924520
12600.	12587.09108515	12601.25359143	12599.85456132	12600.00305157
-46200.	-46157.96724918	-46204.06225137	-46199.51558684	-46200.00068443
83160.	83091.96322716	83166.55034060	83159.20278400	83159.98916928
-72072.	-72018.43333244	-72077.14114652	-72071.36429214	-72071.98379276
24024.	24007.64247141	24025.56586139	24023.80389959	24023.99309676
Max Error	1.37E-3	1.35E-4	1.42E-5	1.24E-6

James A. Walters of Smyrna, Georgia, who uses HP products and is a subscriber to the PPC Calculator Journal, has proposed another method for evaluating the quality of the solution. If the solution vector is multiplied by the original matrix, then the result should be the original vector, in this case the unity vector. Any difference is a measure of the error in the solution. This is the same method by which we evaluate the quality of a least squares fit. I wrote a short program to check the solution. The results, vertically in-line with each corresponding solution above, are:

1.	0.999999996	0.999999943	1.000000028	0.999999945
1.000000003	0.999999981	0.999999957	1.000000021	0.999999961
1.	0.999999993	0.999999977	1.000000026	0.999999971
1.	0.999999997	0.999999982	1.000000012	0.999999968
1.000000001	0.999999892	0.999999999	1.000000007	0.999999974
1.	0.999999993	0.999999993	1.000000002	0.999999977
1.	1.000000013	0.99999999	0.999999997	0.999999979
Max error	1.9E-8	5.7E-8	2.8E-8	5.5E-8
RMS error	1.0E-8	2.9E-8	1.7E-8	3.4E-8

Due to truncation in the calculator, even the exact solution yields some error. This particular check was obtained for each row of the matrix by dividing each element of the solution vector by the exact integer inverse of the corresponding element from the matrix. Somewhat different results can be obtained by multiplying by the reciprocals, or by a double divide process. The order of magnitude of the errors stays the same. The relative error of the elements of the solution vector when compared with the exact solution decreases by two orders of magnitude through use of the double divide technique, there is only a slight increase in the error as measured against the input vector--at least for this particular test.

Clearly we need more investigation. But results so far suggest that Mr. Kahan's report of the non-commutative multiply anomaly may have led directly to the discovery of a technique for obtaining even better results from our TI-58's and TI-59's.

A Workaround for the Non-commutative Multiply on the TI-59 - (cont)

Listing for the Prins/Programbiten Program with Double Divides

000	92	RTN	040	06	06	080	05	05	120	22	INV	160	43	RCL	200	69	OP
001	24	CE	041	35	1/X	081	69	OP	121	97	DSZ	161	02	02	201	27	27
002	69	OP	042	54	)	082	22	22	122	05	05	162	22	INV	202	09	9
003	37	37	043	24	CE	083	93	.	123	02	02	163	44	SUM	203	42	STO
004	09	9	044	74	SM*	084	00	0	124	00	00	164	03	03	204	06	06
005	42	STO	045	03	03	085	01	1	125	08	8	165	69	OP	205	97	DSZ
006	03	03	046	69	OP	086	44	SUM	126	42	STO	166	26	26	206	01	01
007	42	STO	047	23	23	087	04	04	127	03	03	167	97	DSZ	207	00	00
008	06	06	048	69	OP	088	43	RCL	128	43	RCL	168	08	08	208	09	09
009	43	RCL	049	26	26	089	04	04	129	01	01	169	01	01	209	69	OP
010	07	07	050	32	X:T	090	99	PRT	130	42	STO	170	40	40	210	34	34
011	42	STO	051	97	DSZ	091	91	R/S	131	08	08	171	43	RCL	211	73	RC*
012	02	02	052	02	02	092	99	PRT	132	69	OP	172	01	01	212	06	06
013	43	RCL	053	00	00	093	72	ST*	133	23	23	173	44	SUM	213	99	PRT
014	04	04	054	37	37	094	03	03	134	69	OP	174	06	06	214	91	R/S
015	59	INT	055	43	RCL	095	69	OP	135	32	32	175	42	STO	215	69	OP
016	42	STO	056	07	07	096	23	23	136	00	0	176	08	08	216	26	26
017	04	04	057	42	STO	097	00	0	137	63	EX*	177	97	DSZ	217	97	DSZ
018	42	STO	058	02	02	098	63	EX*	138	03	03		05	05	218	04	04
019	05	05	059	44	SUM	099	03	03	139	32	X:T		01	01	219	02	02
020	69	OP	060	03	03	100	69	OP	140	69	OP	180	32	32	220	11	11
021	24	24	061	97	DSZ	101	33	33	141	23	23	181	43	RCL	221	00	0
022	29	CP	062	05	05	102	74	SM*	142	00	0	182	02	02	222	81	RST
023	67	EQ	063	00	00	103	03	03	143	63	EX*	183	44	SUM	223	00	0
024	00	00	064	26	26	104	43	RCL	144	03	03	184	03	03	224	00	0
025	65	65	065	93	.	105	05	05	145	75	-	185	69	OP	225	00	0
026	93	.	066	00	0	106	22	INV	146	43	RCL	186	23	23	226	76	LBL
027	00	0	067	01	1	107	64	PD*	147	02	02	187	00	0	227	11	A
028	01	1	068	44	SUM	108	03	03	148	44	SUM	188	63	EX*	228	94	+/-
029	44	SUM	069	04	04	109	69	OP	149	03	03	189	06	06	229	47	CMS
030	04	04	070	43	RCL	110	23	23	150	32	X:T	190	69	OP	230	42	STO
031	43	RCL	071	04	04	111	97	DSZ	151	55	÷	191	26	26	231	01	01
032	04	04	072	99	PRT	112	02	02	152	32	X:T	192	72	ST*	232	42	STO
033	99	PRT	073	91	R/S	113	00	00	153	73	RC*	193	03	03	233	07	07
034	91	R/S	074	99	PRT	114	83	83	154	06	06	194	69	OP	234	93	.
035	99	PRT	075	42	STO	115	43	RCL	155	35	1/X	195	23	23	235	01	1
036	94	+/-	076	05	05	116	04	04	156	54	)	196	97	DSZ	236	34	√X
037	55	÷	077	73	RC*	117	59	INT	157	24	CE	197	08	08	237	33	X²
038	32	X:T	078	03	03	118	42	STO	158	72	ST*	198	01	01	238	35	1/X
039	73	RC*	079	44	SUM	119	05	05	159	03	03	199	87	87	239	86	STF

Listing for the Nick/Ristanovic Program with Double Divides

000	76	LBL	040	29	CP	080	58	58	120	43	RCL	160	55	÷	200	91	R/S
001	10	E'	041	69	OP	081	61	GTO	121	03	03	161	82	HIR	201	69	OP
002	42	STO	042	20	20	082	01	01	122	10	E'	162	16	16	202	21	21
003	03	03	043	43	RCL	083	84	84	123	43	RCL	163	35	1/X	203	06	6
004	75	-	044	00	00	084	19	D'	124	01	01	164	95	=	204	75	-
005	01	1	045	42	STO	085	43	RCL	125	22	INV	165	24	CE	205	43	RCL
006	95	=	046	01	01	086	00	00	126	64	PD*	166	32	X:T	206	01	01
007	55	÷	047	42	STO	087	10	E'	127	05	05	167	43	RCL	207	95	=
008	82	HIR	048	04	04	088	73	RC*	128	69	OP	168	01	01	208	55	÷
009	17	17	049	10	E'	089	05	05	129	35	35	169	10	E'	209	82	HIR
010	35	1/X	050	25	CLR	090	32	X:T	130	97	DSZ	170	32	X:T	210	17	17
011	85	+	051	73	RC*	091	43	RCL	131	02	02	171	22	INV	211	65	×
012	24	CE	052	05	05	092	01	01	132	01	01	172	74	SM*	212	53	(
013	43	RCL	053	52	EE	093	10	E'	133	23	23	173	05	05	213	22	INV
014	04	04	054	22	INV	094	32	X:T	134	82	HIR	174	69	OP	214	59	INT
015	85	+	055	67	EQ	095	63	EX*	135	18	18	175	34	34	215	69	OP
016	05	5	056	01	01	096	05	05	136	42	STO	176	97	DSZ	216	10	10
017	95	=	057	10	10	097	32	X:T	137	01	01	177	02	02	217	85	+
018	42	STO	058	29	CP	098	[illegible]	RCL	138	43	RCL	178	01	01	218	01	1
019	05	05	059	25	CLR	099	[illegible]	00	139	01	01	179	55	55	219	95	=
020	92	RTN	060	73	RC*	100	10	E'	140	32	X:T	180	97	DSZ	220	22	INV
021	76	LBL	061	05	05	101	32	X:T	141	43	RCL	181	01	01	221	44	SUM
022	19	D'	062	52	EE	102	72	ST*	142	00	00	182	01	01	222	01	01
023	82	HIR	063	22	INV	103	05	05	143	67	EQ	183	38	38	223	61	GTO
024	17	17	064	67	EQ	104	69	OP	144	01	01	184	82	HIR	224	01	01
025	42	STO	065	00	00	105	34	34	145	30	80	185	18	18	225	98	98
026	04	04	066	84	84	106	97	DSZ	146	42	STO	186	32	X:T	226	76	LBL
027	75	-	067	82	HIR	107	02	02	147	04	04	187	43	RCL	227	11	A
028	43	RCL	068	17	17	108	00	00	148	32	X:T	188	00	00	228	37	P/R
029	00	00	069	44	SUM	109	85	85	149	10	E'	189	22	INV	229	06	6
030	95	=	070	05	05	110	43	RCL	150	73	RC*	190	77	GE	230	48	EXC
031	42	STO	071	69	OP	111	00	00	151	05	05	191	12	B	231	01	01
032	02	02	072	21	21	112	42	STO	152	82	HIR	192	22	INV	232	91	R/S
033	69	OP	073	43	RCL	113	04	04	153	06	06	193	58	FIX	233	72	ST*
034	22	22	074	01	01	114	10	E'	154	19	D'	194	06	6	234	01	01
035	92	RTN	075	32	X:T	115	73	RC*	155	43	RCL	195	42	STO	235	69	OP
036	76	LBL	076	82	HIR	116	05	05	156	00	00	196	01	01	236	21	21
037	12	B	077	18	18	117	42	STO	157	10	E'	197	25	CLR	237	61	GTO
038	58	FIX	078	77	GE	118	01	01	158	73	RC*	198	73	RC*	238	02	02
039	08	08	079	00	00	119	19	D'	159	05	05	199	01	01	239	32	32

POLYNOMIAL REGRESSION WITH VARIANCE - Gene Friel. This program is an extension of Thomas Wysmuller's Polynomial Regression Program which appeared in V4N6P6/7 of PPX Exchange. The program provides a least squares polynomial curve fit of the form

$$Y = a_0 + a_1 X + a_2 X^2 \ldots a_n X^n$$

where the highest degree may be selected to be from 1 to 7. The modifications provide for magnetic card entry of the data used for headings and annotation, and an additional program which provides for calculation of variance, but only by reentering the data pairs. The variance is calculated using the formula

$$\sigma^2 = \frac{\sum_{1}^{N} (Y_i - a_0 - a_1 X_i - a_2 X_i^2 - \ldots - a_n X_i^n)}{(N - n - 1)}$$

where N is the number of data pairs and n is the degree of polynomial selected.

Record the four banks of program A on magnetic cards using the startup partitioning. The program changes to the running partitioning at initialization. Record bank 1 of program B using partitioning 10 Op 17.

User Instructions:

1. With the Master Library module installed, enter all four card sides of program A.

2. Enter the highest degree of polynomial you expect to evaluate (from 1 to 7) and press E'. The printer will document that decision with the notation "HIGHEST PERMISSIBLE DEGREE OF REGRESSION" followed by the degree selected. If you do not perform this step the program will select degree 7 as the default mode. Selection of the lowest possible degree will save operator time.

3. Proceed to enter data.

a. Enter the independent variable (X) and press A. The entered value is printed with the annotation "X".

b. Enter the dependent variable (Y) and press B. The entered value is printed with the annotation "Y". The calculator will run for some time as all the sums required for the least squares solution are updated, and then print the total number of data pairs entered so far with the annotation "N", and stop with the same value in the display. The approximate run times versus highest degree of polynomial selected in step 2 are:

Degree	Run Time (seconds)
1	6
2	10
3	13
4	16
5	20
6	25
7	29

Polynomial Regression with Variance - (cont)

From the table you can see that it really is important to select the lowest degree possible to save operator time.

c. Repeat steps 3.a and 3.b until all data pairs have been entered. The number of data pairs should not exceed 35 if you plan to use program B to find the variance.

4. Error Correction:

a. To delete the pair just entered, press C. The printer will document the decision with the annotation "LAST PAIR DELETED" and a printout of the reduced number of data pairs.

b. To delete a pair that had been entered earlier, enter the X value to be deleted and press A, enter the Y value to be deleted and press B. The printout will be the same as for a normal entry. When the calculator stops press C. This deletes the erroneous data pair which was just entered on purpose. The printout will be the same as for a normal deletion (step 4.a). When the calculator stops, press C again. This will delete the erroneous data entered earlier. The decision will be documented with the printout "LAST PAIR DELETED", not an accurate description of what hsppened, and a printout of the reduced number of data pairs.

5. Record the accumulated sums for future use by writing banks 3 and 4 on a blank magnetic card. You only need to do this step if you are planning to try more than one degree of polynomial. This will avoid re-entry of all the data to accumulate the sums again.

6. Solve: Enter the degree of polynomial desired and press E. The calculator will print "DEGREE OF REGRESSION" followed by the degree selected. The printout will include the accumulated sums and the determinant, followed by the regression coefficients. There is no annotation of the regression coefficients. You recognize them as the last group of values printed, with a_0 printed first.

7. To find an estimated Y' from the polynomial using the regression coefficients, enter the X' value and press A'. Repeat with other X' values as often as you like.

8. If you wish to calculate the residuals and variance enter bank 1 of program B. The partitioning will already be at 10 Op 17.

a. Press C to initialize.

b. Enter an x value and press A. The entered value and the register where it is stored are printed.

c. Enter the corresponding Y value and press B. The Y value and the register where it is stored are printed, then the number of pairs entered so far is printed and displayed.

d. Repeat steps 8.b and 8.c for up to 35 pairs.

e. Correct entry errors by using the STO command together with the printed indication of storage location.

Polynomial Regression with Variance - (cont)

Banks 1 and 2 of Program A

```
000 69 OP
001 17 17
002 73 RC*
003 00 00
004 72 ST*
005 01 01
006 69 OP
007 20 20
008 69 OP
009 31 31
010 43 RCL
011 00 00
012 22 INV
013 67 EQ
014 00 00
015 02 02
016 07 7
017 42 STO
018 01 01
019 08 8
020 09 9
021 42 STO
022 99 99
023 43 RCL
024 07 07
025 42 STO
026 00 00
027 73 RC*
028 99 99
029 69 OP
030 21 21
031 72 ST*
032 01 01
033 97 DSZ
034 99 99
035 51 BST
036 97 DSZ
037 00 00
038 00 00
039 27 27
040 43 RCL
041 07 07
042 75 -
043 01 1
044 95 =
045 44 SUM
046 99 99
047 97 DSZ
048 05 05
049 00 00
050 23 23
051 36 PGM
052 02 02
053 13 C
054 01 1
055 42 STO
056 05 05
057 43 RCL
058 07 07
059 42 STO
060 00 00
061 09 9
062 08 8
063 42 STO
064 99 99
065 73 RC*
066 99 99
067 36 PGM
068 02 02
069 71 SBR
070 03 03
071 55 55
072 97 DSZ
073 99 99
074 51 BST
075 97 DSZ
076 00 00
077 00 00
078 65 65
079 43 RCL
080 07 07
081 42 STO
082 00 00
083 42 STO
084 99 99
085 05 5
086 42 STO
087 88 88
088 36 PGM
089 02 02
090 15 E
091 36 PGM
092 02 02
093 16 A'
094 52 EE
095 36 PGM
096 02 02
097 91 R/S
098 72 ST*
099 88 88
100 68 NOP
101 01 1
102 44 SUM
103 88 88
104 97 DSZ
105 00 00
106 00 00
107 95 95
108 43 RCL
109 99 99
110 75 -
111 01 1
112 95 =
113 42 STO
114 04 04
115 43 RCL
116 89 89
117 42 STO
118 13 13
119 25 CLR
120 98 ADV
121 98 ADV
122 98 ADV
123 92 RTN
124 76 LBL
125 17 B'
126 22 INV
127 76 LBL
128 18 C'
129 86 STF
130 01 01
131 42 STO
132 99 99
133 98 ADV
134 99 PRT
135 43 RCL
136 99 99
137 85 +
138 43 RCL
139 03 03
140 95 =
141 61 GTO
142 01 01
143 49 49
144 76 LBL
145 16 A'
146 22 INV
147 86 STF
148 01 01
149 36 PGM
150 07 07
151 13 C
152 87 IFF
153 01 01
154 01 01
155 35 35
156 92 RTN
157 61 GTO
158 01 01
159 35 35
160 76 LBL
161 12 B
162 61 GTO
163 03 03
164 16 16
165 76 LBL
166 11 A
167 42 STO
168 01 01
169 43 RCL
170 05 05
171 69 OP
172 04 04
173 43 RCL
174 01 01
175 69 OP
176 06 06
177 00 0
178 92 RTN
179 76 LBL
180 10 E'
181 61 GTO
182 03 03
183 68 68
184 76 LBL
185 13 C
186 43 RCL
187 07 07
188 69 OP
189 01 01
190 43 RCL
191 08 08
192 69 OP
193 02 02
194 43 RCL
195 09 09
196 69 OP
197 03 03
198 43 RCL
199 10 10
200 71 SBR
201 04 04
202 28 28
203 43 RCL
204 04 04
205 22 INV
206 44 SUM
207 11 11
208 33 X²
209 22 INV
210 44 SUM
211 19 19
212 02 2
213 22 INV
214 44 SUM
215 20 20
216 22 INV
217 86 STF
218 01 01
219 61 GTO
220 03 03
221 33 33
222 76 LBL
223 14 D
224 43 RCL
225 11 11
226 42 STO
227 01 01
228 43 RCL
229 19 19
230 42 STO
231 02 02
232 43 RCL
233 20 20
234 42 STO
235 03 03
236 43 RCL
237 21 21
238 42 STO
239 04 04
240 43 RCL
241 22 22
242 42 STO
243 05 05
244 43 RCL
245 12 12
246 42 STO
247 06 06
248 00 0
249 92 RTN
250 76 LBL
251 15 E
252 42 STO
253 46 46
254 32 X:T
255 01 1
256 32 X:T
257 77 GE
258 02 02
259 62 62
260 00 0
261 92 RTN
262 32 X:T
263 43 RCL
264 41 41
265 22 INV
266 77 GE
267 02 02
268 60 60
269 98 ADV
270 71 SBR
271 04 04
272 14 14
273 32 X:T
274 99 PRT
275 85 +
276 01 1
277 95 =
278 42 STO
279 05 05
280 42 STO
281 07 07
282 65 ×
283 02 2
284 85 +
285 01 1
286 09 9
287 42 STO
288 00 00
289 95 =
290 32 X:T
291 98 ADV
292 09 9
293 08 8
294 42 STO
295 01 01
296 73 RC*
297 00 00
298 99 PRT
299 69 OP
300 20 20
301 43 RCL
302 00 00
303 77 GE
304 03 03
305 09 09
306 61 GTO
307 02 02
308 96 96
309 01 1
310 01 1
311 42 STO
312 00 00
313 01 1
314 00 0
315 81 RST
316 42 STO
317 04 04
318 86 STF
319 01 01
320 43 RCL
321 06 06
322 69 OP
323 04 04
324 43 RCL
325 04 04
326 69 OP
327 06 06
328 44 SUM
329 11 11
330 33 X²
331 44 SUM
332 19 19
333 02 2
334 01 1
335 42 STO
336 02 02
337 43 RCL
338 39 39
339 32 X:T
340 01 1
341 44 SUM
342 20 20
343 65 ×
344 43 RCL
345 01 01
346 95 =
347 42 STO
348 03 03
349 87 IFF
350 01 01
351 03 03
352 54 54
353 94 +/-
354 74 SM*
355 02 02
356 69 OP
357 22 22
358 43 RCL
359 02 02
360 77 GE
361 04 04
362 33 33
363 43 RCL
364 03 03
365 61 GTO
366 03 03
367 43 43
368 42 STO
369 41 41
370 98 ADV
371 43 RCL
372 42 42
373 69 OP
374 01 01
375 43 RCL
376 43 43
377 69 OP
378 02 02
379 43 RCL
380 44 44
381 69 OP
382 03 03
383 43 RCL
384 45 45
385 71 SBR
386 04 04
387 28 28
388 71 SBR
389 04 04
390 14 14
391 43 RCL
392 41 41
393 65 ×
394 02 2
395 85 +
396 02 2
397 01 1
398 95 =
399 42 STO
400 39 39
401 85 +
402 03 3
403 95 =
404 55 ÷
405 02 2
406 95 =
407 42 STO
408 40 40
409 43 RCL
410 41 41
411 99 PRT
412 98 ADV
413 92 RTN
414 43 RCL
415 35 35
416 69 OP
417 01 01
418 43 RCL
419 36 36
420 69 OP
421 02 02
422 43 RCL
423 37 37
424 69 OP
425 03 03
426 43 RCL
427 38 38
428 69 OP
429 04 04
430 69 OP
431 05 05
432 92 RTN
433 43 RCL
434 40 40
435 32 X:T
436 01 1
437 02 2
438 42 STO
439 02 02
440 01 1
441 65 ×
442 43 RCL
443 01 01
444 95 =
445 42 STO
446 03 03
447 65 ×
448 43 RCL
449 04 04
450 95 =
451 87 IFF
452 01 01
453 04 04
454 56 56
455 94 +/-
456 74 SM*
457 02 02
458 69 OP
459 22 22
460 43 RCL
461 02 02
462 77 GE
463 04 04
464 70 70
465 43 RCL
466 03 03
467 61 GTO
468 04 04
469 41 41
470 43 RCL
471 00 00
472 69 OP
473 04 04
474 43 RCL
475 20 20
476 69 OP
477 06 06
478 98 ADV
479 92 RTN
```

Polynomial Regression with Variance - (cont)

Banks 3 and 4 of Progrsm A:

31000000.	00	0.	20	19.	40
0.	01	0.	21	7.	41
0.	02	0.	22	2324222317.	42
0.	03	0.	23	3637003317.	43
0.	04	0.	24	3530243636.	44
44000000.	05	0.	25	2414271700.	45
45000000.	06	0.	26	0.	46
27133637.	07	0.	27	0.	47
33132435.	08	0.	28	0.	48
16172717.	09	0.	29	0.	49
3717160000.	10	0.	30	0.	50
0.	11	0.	31	0.	51
0.	12	0.	32	0.	52
0.	13	0.	33	0.	53
0.	14	0.	34	0.	54
0.	15	1617223517.	35	0.	55
0.	16	1700322100.	36	0.	56
0.	17	3517223517.	37	0.	57
0.	18	3636243231.	38	0.	58
0.	19	35.	39	0.	59

Bank 1 of Program B:

000	76	LBL	027	43	RCL	054	17	17	081	17	B'	108	42	STO	135	06	6
001	16	A'	028	16	16	055	01	1	082	00	0	109	00	00	136	69	OP
002	53	(	029	16	A'	056	44	SUM	083	98	ADV	110	43	RCL	137	17	17
003	24	CE	030	65	×	057	16	16	084	92	RTN	111	13	13	138	43	RCL
004	85	+	031	01	1	058	92	RTN	085	76	LBL	112	42	STO	139	15	15
005	22	INV	032	00	0	059	76	LBL	086	13	C	113	14	14	140	55	÷
006	23	LNX	033	00	0	060	11	A	087	25	CLR	114	73	RC*	141	53	(
007	55	÷	034	85	+	061	72	ST*	088	42	STO	115	00	00	142	43	RCL
008	03	3	035	43	RCL	062	00	00	089	19	19	116	36	PGM	143	13	13
009	04	4	036	17	17	063	32	X:T	090	42	STO	117	07	07	144	75	-
010	00	0	037	16	A'	064	18	C'	091	17	17	118	13	C	145	43	RCL
011	00	0	038	95	=	065	01	1	092	03	3	119	75	-	146	04	04
012	85	+	039	17	B'	066	44	SUM	093	42	STO	120	69	OP	147	75	-
013	01	1	040	69	OP	067	19	19	094	16	16	121	20	20	148	01	1
014	54	)	041	20	20	068	00	0	095	03	3	122	73	RC*	149	95	=
015	59	INT	042	01	1	069	92	RTN	096	00	0	123	00	00	150	42	STO
016	92	RTN	043	44	SUM	070	76	LBL	097	42	STO	124	95	=	151	18	18
017	76	LBL	044	17	17	071	12	B	098	00	00	125	33	X^2	152	92	RTN
018	17	B'	045	43	RCL	072	72	ST*	099	00	0	126	44	SUM	153	98	ADV
019	69	OP	046	17	17	073	00	00	100	92	RTN	127	15	15	154	32	X:T
020	04	04	047	32	X:T	074	32	X:T	101	76	LBL	128	69	OP	155	04	4
021	32	X:T	048	09	9	075	18	C'	102	14	D	129	20	20	156	02	2
022	69	OP	049	77	GE	076	43	RCL	103	25	CLR	130	68	NOP	157	17	B'
023	06	06	050	00	00	077	19	19	104	42	STO	131	97	DSZ	158	98	ADV
024	92	RTN	051	58	58	078	32	X:T	105	15	15	132	14	14	159	92	RTN
025	76	LBL	052	00	0	079	03	3	106	03	3	133	01	01			
026	18	C'	053	42	STO	080	01	1	107	00	0	134	14	14			

Editor's Note: This is a very versatile program for polynomial regression, but it does involve a lot of magnetic card manipulation. Users who can limit the highest degree to 4, and the number of data pairs to 20 might wish to use my polynomial curve fit program. The data points are entered as rapidly as you can key them in, with no wait for accumulation of sums. The entered data pairs may be edited in a manner similar to step 8.e of the Friel program. Then one option accumulates the sums, automatically loads the second half of the program using INV Write, and prints out the complete solutions for degrees 1 through 4 including residuals and the standard error (the square root of variance) without any additional user action.
