* TI PPC NOTES *

v5N7, 1980

NEWSLETTER OF THE TI PERSONAL PROGRAMMABLE CALCULATOR CLUB.

9213 Lanham Severn Road, Lanham, Maryland, 20801.

IN MEMORIAM: JAMES DALLAS EGBERT III,(384) died on August 16, 1980. Old timers will remember Dallas Egbert as the youngest member of the 52-User's Club (14 years old) and as one of the most active contributors of SR-52 routines. Dallas died in Dayton Ohio before his eighteenth birthday from an apparently self-inflicted gunshot wound.

Don O'Grady, who wrote, among other software, the programs for the M/U and the RPN mods , surprised me with a last-minute solution to the "brain teasers" on page 5.
His suggestion is : INV Σ+ RAD INV COS + X:T GRAD INV COS + P/R 1/x INT =
Follow this routine by either an INT or an INV INT, as needed.
He also sent me an SR-56 Mastermind program. As I had stated "Do not send me any more TI-58/59 Mastermind programs", this program, however, is an animal of a different kind. I will be happy to publish it in one of the future issues.

The September issue of Popular Science in the Calcu-letter column by Darrell Huff mentions the existence of our club. Thanks to this plug I am swamped once again with tens of letters asking for information. But I am not complaining. Thanks Darrell!

I received a long letter and all the newsletters that appeared since 1977 from the two editors of the Danish Programmeringsklub '77, Morton Jørgensen and Ib-Michael Martinsen. Their address is: Kanehaven 32, 8240 Risskov, Denmark. These fellas tell me I can write them in either"Danish, English, German, French, Norwegian, Swedish or anything, but unfortunately not Flemish, sorry." That is a real disappointment, friends. I will have to make English do in that case and continue to write in Flemish only to Tom Coppens in Kapellen, Belgium! Many thanks for the material! Luckily I found an able translator for Swedish and Danish in Dan Lufkin, a physicist working for the US government and living about 10 miles from Lanham in Rockville, Maryland.

Several members found a glitch in the BOWLING program of v5n6p15. The cure for it is
1. Write RCL 39 in step 196.
2. Load reg 44 with .1 (NOT .0001) Record both changes permanently on the mag cards.

Copying the display into the t-reg (v5n6p4) illicited quite a lot of comment. Jeff Rosedale has this gem, which runs faster, since the Y^x is rather time-consuming: ..+ X:T 0) Morton Matthew says:".. actually, doesn't X x:t 1 do it? In fact, if t is holding a 0, how about just + x:t ? I want to add = to all of these, but depending on what follows, no need maybe."

Harald M. Otto, Lerchenweg 6, D-4502 Bad Rothenfelde, West Germany, the author of several very good compilations of TI-59 Tricks, (see also v5n4/5p2) in my opinion, has become one of the most imaginative practioners of TI-59 programming. His latest "tricks" book is 112 pages long and so chock full of good programs and routines, that it will take me weeks to digest everything. I will try to bring you some of his jewels in the next issue.

A special message to our friends of the HP PPC JOURNAL: <u>the heat is on</u>. See page 7 of this issue. Palmer Hansen has re-written Panos Galidas' 2 min 38.5 sec calendar program, such that it may be used in Martin Neef's FAST MODE. It now runs at the incredible speed of 1 min 32 sec per calendar year!!!!

Maurice E. T. Swinnen

TABLE OF CONTENTS

CONVERSION TO ABSOLUTE ADDRESSES

The program will determine and list the locations of program steps needed to absolute address a program. The program is also suitable for determining the correct addresses for programs that have been modified.

1. List the program to be absolute addressed [list].

2. List the labels of the program to be absolute addressed [OP 8].

3. Clear the calculator and enter the conversion to absolute address program (2 sides)

4. Press "E'" to initialize. RST is printed.

5. Enter the locations of each label to be deleted exactly as each appears in the OP 8 listing, pressing "A" for each label. Both the location and "LBL" are printed. [see note 1]

6. Enter the value of each absolute address in the program (if any), and press "B". Both the location and "ADR" are printed. [see note 2]

7. Review the listing of the program to be absolute addressed and determine any locations where steps must be either inserted or deleted.

8. Where the insertion of ...

   a) a single step is required, enter that location and press "E". Both the location and "INS" are printed.

   b) 2 - 9 steps is required, enter the first location (xxx) along with the number of steps (y) as xxx.y and press "E". Both the number of steps and "INS" are printed.

   c) more than 9 steps is required, use multiple "E" entries, with the same location for each entry. [see note 3]

9. Deletions are handled as insertions, except that "D" is pressed instead of "E". Both the location and "DEL" are printed. [see note 4]

10. Press "C" to compute and list the new locations. The new listing includes two numbers separated by a decimal point for each label or address entered. The three digits to the right of the decimal represent the location initially entered. The three digits to the left of the decimal represent the computed absolute address corresponding to that location.

11. "C" may be pressed again to relist the locations, if desired.

12. Steps 8 - 10 may be repeated as needed to enter and compute all the required instructions.

13. Reenter the program to be absolute addressed, and, starting at the end of the program, replace the label calls with the absolute addresses indicated. Delete the labels which have been replaced and perform any other indicated INS or DEL operations.

NOTES

1. The program can store up to 65 labels and 65 instructions. If either of these limits is reached, the calculator will return a flashing display whenever additional entries are attempted. Every label entry automatically includes an instruction to delete two steps. For the program to perform its function, all label and absolute address entries must be made before Step 10 (computation and listing) is begun by pressing "C". However, additional INS and/or DEL instructions may be made and computed following a previous computation. Therefore if the total number of instructions could conceivable exceed 65, all LBL entries should be made before any of the INS or DEL instructions.

2. As a check, it is often convenient to enter the last step of the program as an address. After tabulation and modification of the program, it should coincide with the last step of the new absolute addressed program.

3. To insert 25 steps following step 100 in a program
   a. enter 100.9 and press "E".
   b. enter 100.9 and press "E"
   c. enter 100.7 and press "E".

4. If an error is made in the entry of labels, addresses or instructions, it may be corrected in one of several ways;
   a. If the error was the last entry and...
      1) was a label entry, decrement memories 00, 01,02 & 03 by 1.
      2) was an address entry, decrement memories 00 & 02 by 1.
      3) was an insert or delete instruction, decrement memories 01 & 03 by 1.

   b. If the error was not the last entry,
      1) make additional entries which cancel the erroneous insert or delete instructions (or press "E'" and reenter all data). An INS error at a particular location, for example, may be corrected by a DEL instruction at that same location.

      2) but involved LBL, that label address call may be ignored. However, the two steps in the program which are automatically deleted by the label operation must be restored by an insertion of two steps at the location of the label.

      3) but involved only ADR, that address may simply be ignored.

CONVERSION TO ABSOLUTE ADDRESSES, *John Worthington, Bowie, Maryland.*

CONVERSION TO
ABSOLUTE ADDRESSES

```
000  92 RTN
001  76 LBL
002  18 C'
003  01  1
004  00  0
005  00  0
006  00  0
007  54  )
008  92 RTN
009  22 INV
010  58 FIX
011  25 CLR
012  43 RCL
013  04  04
014  69 OP
015  06  06
016  81 RST
017  76 LBL
018  11  A
019  32 X:T
020  86 STF
021  03  03
022  02  2
023  07  7
024  01  1
025  04  4
026  02  2
027  07  7
028  61 GTO
029  00  00
030  40  40
031  76 LBL
032  12  B
033  32 X:T
034  01  1
035  03  3
036  01  1
037  06  6
038  03  3
039  05  5
040  69 OP
041  04  04
042  32 X:T
043  50 I×I
044  59 INT
045  42 STO
046  04  04
047  85  +
048  32 X:T
049  53  (
050  18 C'
051  22 INV
052  77  GE
053  01  01
054  70  70
055  53  (
056  43 RCL
057  04  04
058  85  +
059  43 RCL

060  00  0
061  71 SBR
062  03  03
063  63  63
064  22 INV
065  87 IFF
066  03  03
067  00  00
068  73  73
069  71 SBR
070  03  03
071  61  61
072  01  1
073  93  .
074  54  )
075  69 OP
076  20  20
077  69 OP
078  22  22
079  65  ×
080  18 C'
081  85  +
082  73 RC*
083  00  00
084  22 INV
085  59 INT
086  65  ×
087  29 CP
088  77  GE
089  00  00
090  95  95
091  01  1
092  94 +/-
093  82 HIR
094  41  41
095  01  1
096  95  =
097  72 ST*
098  00  00
099  93  .
100  02  2
101  85  +
102  87 IFF
103  03  03
104  01  01
105  44  44
106  61 GTO
107  00  00
108  09  09
109  76 LBL
110  15  E
111  86 STF
112  01  01
113  76 LBL
114  14  D
115  42 STO
116  04  04
117  71 SBR
118  03  03
119  61  61

120  18 C'
121  22 INV
122  77  GE
123  01  01
124  70  70
125  02  2
126  04  4
127  03  3
128  01  1
129  03  3
130  06  6
131  87 IFF
132  01  01
133  01  01
134  42  42
135  75  -
136  08  8
137  01  1
138  04  4
139  00  0
140  09  9
141  95  =
142  69 OP
143  04  04
144  43 RCL
145  04  04
146  95  =
147  50 I×I
148  69 OP
149  21  21
150  69 OP
151  23  23
152  55  ÷
153  18 C'
154  85  +
155  73 RC*
156  01  01
157  59 INT
158  50 I×I
159  95  =
160  87 IFF
161  01  01
162  01  01
163  65  65
164  94 +/-
165  72 ST*
166  01  01
167  61 GTO
168  00  00
169  09  09.
170  25 CLR
171  35 1/X
172  43 RCL
173  04  04
174  81 RST
175  76 LBL
176  13  C
177  98 ADV
178  69 OP
179  00  00

180  03  3
181  01  1
182  01  1
183  07  7
184  04  4
185  03  3
186  04  4
187  00  0
188  03  3
189  02  2
190  69 OP
191  02  02
192  01  1
193  00  0
194  00  0
195  02  2
196  07  7
197  01  1
198  06  6
199  82 HIR
200  07  07
201  69 OP
202  05  05
203  69 OP
204  00  00
205  04  4
206  42 STO
207  01  01
208  69 OP
209  23  23
210  22 INV
211  97 DSZ
212  03  03
213  02  02
214  98  98
215  69 OP
216  21  21
217  04  4
218  48 EXC
219  00  00
220  75  -
221  03  3
222  95  =
223  42 STO
224  02  02
225  73 RC*
226  01  01
227  22 INV
228  59 INT
229  65  ×
230  18 C'
231  50 I×I
232  75  -
233  22 INV
234  59 INT
235  82 HIR
236  07  07
237  95  =
238  66 PAU
239  32 X:T

240  01  1
241  00  0
242  82 HIR
243  47  47
244  22 INV
245  97 DSZ
246  02  02
247  02  02
248  10  10
249  69 OP
250  20  20
251  73 RC*
252  00  00
253  59 INT
254  55  ÷
255  18 C'
256  50 I×I
257  22 INV
258  59 INT
259  65  ×
260  18 C'
261  22 INV
262  77  GE
263  02  02
264  44  44
265  00  0
266  32 X:T
267  42 STO
268  04  04
269  82 HIR
270  17  17
271  59 INT
272  22 INV
273  67  EQ
274  02  02
275  77  77
276  01  1
277  65  ×
278  18 C'
279  65  ×
280  53  (
281  73 RC*
282  00  00
283  65  ×
284  73 RC*
285  01  01
286  54  )
287  69 OP
288  10  10
289  95  =
290  74 SM*
291  00  00
292  43 RCL
293  04  04
294  32 X:T
295  61 GTO
296  02  02
297  44  44
298  43 RCL
299  00  00

300  75  -
301  04  4
302  42 STO
303  01  01
304  95  =
305  48 EXC
306  02  02
307  42 STO
308  03  03
309  58 FIX
310  03  03
311  05  5
312  42 STO
313  00  00
314  43 RCL
315  02  02
316  42 STO
317  04  04
318  73 RC*
319  00  00
320  69 OP
321  20  20
322  50 I×I
323  59 INT
324  55  ÷
325  18 C'
326  99 PRT
327  97 DSZ
328  04  04
329  03  03
330  18  18
331  43 RCL
332  02  02
333  85  +
334  04  4
335  95  =
336  42 STO
337  00  00
338  25 CLR
339  42 STO
340  03  03
341  98 ADV
342  22 INV
343  58 FIX
344  81 RST
345  76 LBL
346  10 E'
347  22 INV
348  58 FIX
349  07  7
350  69 OP
351  17  17
352  47 CMS
353  04  4
354  42 STO
355  00  00
356  48 EXC
357  01  01
358  86 STF
359  09  09

360  81 RST
361  43 RCL
362  01  01
363  32 X:T
364  06  6
365  08  8
366  22 INV
367  77  GE
368  01  01
369  70  70
370  92 RTN

     LABELS
002  18 C'
018  11  A
032  12  B
110  15  E
114  14  D
176  13  C
346  10 E'
```

HEATING, PIPING, AIR CONDITIONING.- Three 1980 issues of the journal by that name contain some good TI-59 programs. Each program is accompanied by extensive documentation:
March 1980: Heat Gain Calculations for Programmable Calculators.Harry C.Sutch, PE.
April 1980: Indoor Swimming Pool Design Program. M.D. Syed & D.W. Strang, PE.
July 1980: Duct Weight Estimated by Programmable Calculator, H.C. Sutch, P.E.

NEW PPX PROGRAMS.- Terrence O'Brien, one of our more prolific programming members (he is also the head of the Department of Marketing at Kansas State U) has sent me copies of some of his latest contributions to PPX:
Register Roll, simulates INV LIST without the printer. All 100 registers can be dumped sequentially into display to quickly check their contents.
Packed Data,provides standardized, compact storage for data bases, 13 digits/reg.
Bivariate Unpacking; this program is compatible with Packed Data.
Good programming combined with clear and concise documentation. I wish all the PPX submissions were done this way. Be on the look out for these programs in the next addendum of the PPX software catalog.

SOLAR ENERGY SOFTWARE.- The Sunshine Power Co, 1018 Lancer Drive, SanJose CA 95129, sells software with solar energy application for the HP-67/97 and the TI-59. For the 59, the programs are well written and are contained on prerecorded magnetic cards. The documentation contains instructions, formulas, definitions and references. The entire solar engineering library costs $ 250.00 US, while individual programs costs between $ 30.00 and $ 60.00 US.

OIL DRILLING TECHNOLOGY.- Imco Services, a division of Halliburton Co, P.O Box 22605, Houston, TX 77027, sells a special module to be used with the TI-58/59 and intended to compute drilling parameters. One of the pricipal programmers is David Engle, a member of the TI PPC club. The module, known under the name of IMCO DRILLMOD, sells for $ 150.00 US plus tax. It comes with extensive documentation. David tells me that he is working on a second module, intended to complement the first one.Drillers refer to their science as "mud technology."

FAST FOURIER TRANSFORM- Robert E. Harrison, one of our PPC members, who lives in Rancho Palos Verdes, California, had a program by that name in Electronics, May 10, 1980, p 223. If you have seen this program and wondered why it didn't work, blaim printer's error for your disappointment. Robert has these corrections:
Step 152 should be 78 and not 76
Step 165 should be STO and not STD
Step 204 should be C and 0.
Incidentally, the program runs in RAD mode, but is should not be necessary to use the RAD command explicitly. The calculator is left in the RAD mode after the first pass through step 204.
Bill Skillman sent me an enhanced version of this program. If you send me a SASE I will send you a copy of this three-page enhancement. It contains only the program listing. For instructions how to use the program you still have to refer to Robert's original program in Electronics, which I am not allowed to copy. (copyrighted!)

TI-59 MOS FET PROGRAM AIDS LSI DESIGNERS.- Ray Pinkham, Texas Instruments, Houston, TX Electronics, July 17, 1980, pp 137-139.
I don't know Ray Pinkham, and I don't want to criticize his engineering ability. It is sound. His program works. But why he has 59 NOPs in a total of 453 steps is beyond me. And why magazines like Electronics continue publishing such obviously "not reviewed" programs is also a mistery to me. I hope enough members protest to those magazines and journals every time they see a glaring "blasphemy" of this kind. At least I do each time. Maybe we can get Ray Pinckham a free programming course in the TIPPP division of his own company.

PROGRAMMING PUZZLES- George Vogel is the author of these delightful puzzles.I will publish your solutions, provided they work. You have about twelve weeks to think about it, because I plan to give George's and anybody else's solution in issue 9. I don't intend to entertain long letters and other correspondence about it: either you have a workable solution or you don't. It is that simple. OK?

1. You design a program which stops in the middle of a subroutine and according to the value displayed, you plan to use your judgement to decide whether to press A or B to complete the computation. Let's say that labels A and B are both within the subroutine, e.g.,....R/S LBL A 2 + LBL B 1 = RTN.... But it doesn't work: the program always stops at the RTN and will not return to the main program. What is wrong? How can you make your idea work?
2. You have a program with a number of OP 06 comments. You want to keep your choice of FIX open for each calculation. You don't want to use FIX IND, or assemble codes via HIR 8. Can you think of a simple way of avoiding distortion of the comments, regardless of the FIX chosen?
3. Write a histogram program that will accumulate in separate bins (registers) the number of times x occurs in the following steps-of-five ranges: x ≤60, 60≤x<65, 65≤x<70, etc. Program should have 15 program steps or less.
4. Write a program of 14 steps or less that will store from cold start (CMS, RST, 0 X:T, CLR, R/S) say 0.7, 1.7, 2.7, 3.7,....8.7 in registers 0 through 8.
5. Try this on the keyboard: 9.999 EE +/- 87 STO 01 STO 02 1 EE +/- 99 SUM 02 0.1 PRD 01 PRD 02 . You now have two small numbers in the two registers. Then, still from the keyboard, do RCL 01 X:T RCL 02 X=T . Why the flashing? And why do you have to press CE twice to stop the flashing?

BRAIN TEASERS. Re-v5n4/5p13. The object of Stuart Cox's contribution was to produce in the display the numbers 197 and .1415926536 by means of a routine or key stroke sequence of 13 steps or less. All keys, except the numerical ones, could be used. There seems to be some confusion as to what constitutes a numerical key. Many members sent in routines for the second puzzle, running something like : π INV INT .... *That is a little too obvious, don't you think so?*

So, here are the two routines proposed by Stuart:

For .141: INV LNX INV LNX INV LNX Y^X LNX X^2 = LNX LNX RAD COS +/- INV COS

For 197: INV LNX INV LNX INV LNX Y^X LNX X^2 = LNX LNX GRAD COS +/- INV COS

Can this be done in less steps?

CASIO FX-501P/502P CALCULATOR CLUB.- A new calculator club for the Casio programmable calculators has been formed. Six issues of the newsletter are planned. $ 5.00 US and $ 7.00 for those living outside the US. For more information write: Kiyoshi Akima, Casio Calculator Club, 2880 20th Street, Boulder, CO 80302, USA.

CALCULATOR PROGRAM SOLVES GAS EQUATIONS.- Philip R. Rowley, James M. Brown Ltd. Chemical Engineering, February 11, 1980, pp 97-100. This program calculates variables in the ideal and van der Waals equations of state on the TI-58/59. Accelerated iteration speeds up the computation of number of gas moles and volume. Philip R. Rowley, who is a Development Chemist in the Pigment Division of James M. Brown Ltd, Stoke-on-Trent, Great Britain is also the well known editor of the British TI-Users Club newsletter. A fine piece of programming.

TELEPHONE RATE TIMER.- Re-v5n4/5p24. Bob Baldassano suggests the following enhancement of the instructions:

"When you use the special rate and key D, the first minute usually carries a rate different than subsequent minutes. Thus, enter the starting rate and press D. Then enter the subsequent rate and press R/S. Now press E. Rates are always entered in CENTS."

The M/U MODULE.- Re-v5n4/5p11. Insert a CP between LBL B and INV EQ if you feel uneasy about the possiblity of something in the t-register. (In the program almost at the bottom of page 11) Suggestion by Bob Baldassano.

<u>THE GREAT "SPEEDY FACTOR FINDER" CONTROVERSY.-</u> In v5n6 we had a routine by that name, written by Bill Skillman. Several members wrote in to tell me that the routine was not one hundred percent "kosher." It failed to test for factors 11, 31, 41, etc. What was worse, it called prime numbers such as 341 (11 x 31, 737 (11 x 67) or 4387 (41 x 107).

Several members proposed solutions, all involving rewriting the program starting with step 049. The most prolific was Nello Coda, who even gave two different solutions! I am sorry to say, even those proposed remedies did not cure the sick routine. Still some factors were skipped.

But the whole exercise showed that many members understood and had fun with the sieve of Eratosthenes, although some of the solutions produced an exorbitantly long sieve.

So, what do you do in a case like this? You go directly to the horse's mouth, of course. The equine orifice in this case lives in nearby Baltimore, so consultation was easy. Bill revised the whole routine and offers the one reproduced below. It works, I can assure you! Just try the product 2 x 3 x 5 x 7 x 11 x 13 x 17 x 19 x 23 x 29 x 31 = . It will do the proper factors. And it is fast.

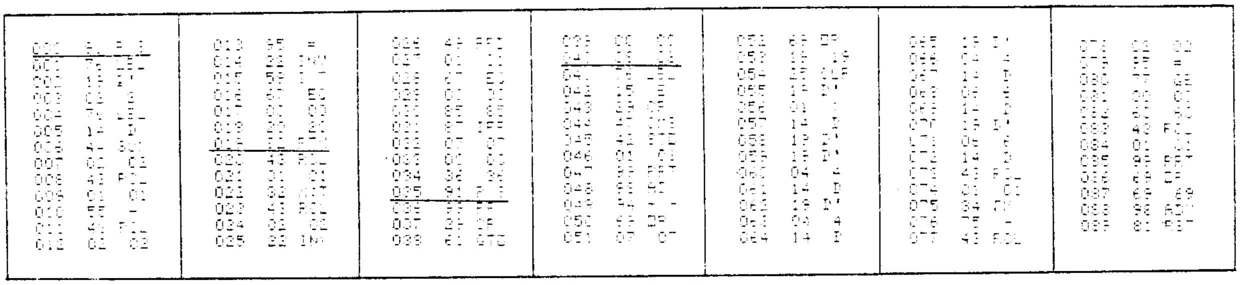
The instructions are still:

1. Enter program, bank 1.

2. Enter N, press E. Factors will be printed, if PC100 is connected.
Last factor flashes. If handheld, use R/S as needed.

Examples: 987654321 = 3, 3, 17, 17, 379721 in 2 min 26 sec. MU-09 finds this in 3 min 30 sec in crom and in 7 min 37 sec downloaded.
103569859 = 463, 467, 479 in 1 min 53 sec.

Only registers 01 and 02 are used. Flag 7 is used in the printer sensing routine. As was stated in v5 n6, this program was originally written for the SR-52 by Mike Louder and modified by Dick Vanderburgh for the TI-59 (v3n11p4)

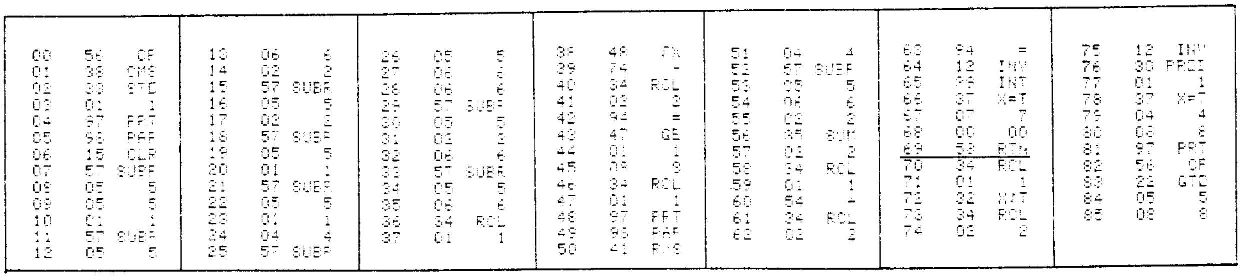


```
000  [illegible]
001  76  LBL
002  15  E
003  03  3
004  76  LBL
005  14  D
006  44  SUM
007  02  02
008  43  RCL
009  01  01
010  55  ÷
011  43  RCL
012  02  02

013  95  =
014  22  INV
015  59  INT
016  67  EQ
017  00  00
018  20  20
019  92  RTN
020  43  RCL
021  01  01
022  32  X:T
023  43  RCL
024  02  02
025  22  INV

026  [illegible]
027  01  [illegible]
028  67  EQ
029  01  01
030  65  [illegible]
031  87  IFF
032  07  07
033  00  00
034  36  36
035  91  R/S
036  [illegible]
037  29  CP
038  61  GTO

039  00  00
040  [illegible]
041  76  LBL
042  15  E
043  29  CP
044  47  CMS
045  42  STO
046  01  01
047  99  PRT
048  98  ADV
049  94  +/-
050  69  OP
051  07  07

052  69  OP
053  [illegible]
054  25  CLR
055  19  D'
056  01  1
057  14  D
058  19  D'
059  19  D'
060  04  4
061  14  D
062  19  D'
063  04  4
064  14  D

065  19  D'
066  04  4
067  14  D
068  [illegible]
069  14  D
070  19  D'
071  05  5
072  14  D
073  43  RCL
074  01  01
075  34  [illegible]
076  75  -
077  43  RCL

078  02  02
079  95  =
080  77  GE
081  00  00
082  [illegible]
083  43  RCL
084  01  01
085  99  PRT
086  69  OP
087  69  69
088  98  ADV
089  81  RST
```

Then Bill reworked the routine for the speedy Gonsalez among calculators, the SR-56. It does 103569859 in 1 min 31 sec and 987654321 in only 1 min 55 sec !!!

This routine uses also registers 1 and 2. If you don't want to use it on the printer you may replace the PRT commands by R/S. The most elegant way would be to replace step 81 by an R/S and steps 48 and 49 by NOP +/- respectively. That will result in displaying the last factor as a negative number.

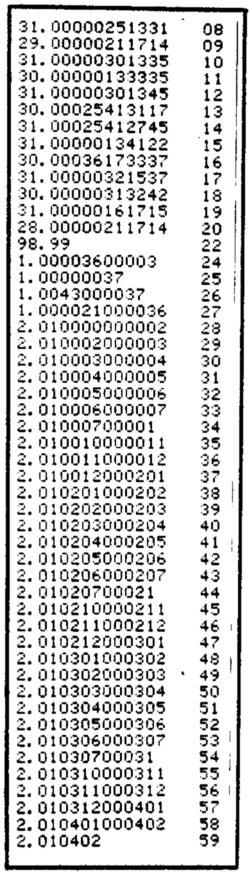
The instructions are: enter N, press RST, R/S. When the first factor is displayed, copy it down and press R/S, and so on, until the last factor shows up negative.



```
00  56  CP
01  38  CMS
02  33  STO
03  01  1
04  97  PRT
05  98  PAP
06  15  CLR
07  57  SUBR
08  05  5
09  05  5
10  01  1
11  57  SUBR
12  05  5

13  06  6
14  02  2
15  57  SUBR
16  05  5
17  02  2
18  57  SUBR
19  05  5
20  01  1
21  57  SUBR
22  05  5
23  01  1
24  04  4
25  57  SUBR

26  05  5
27  06  6
28  06  6
29  57  SUBR
30  05  5
31  03  3
32  06  6
33  57  SUBR
34  05  5
35  06  6
36  34  RCL
37  01  1

38  48  EX
39  74  -
40  34  RCL
41  02  2
42  94  =
43  47  GE
44  01  1
45  09  9
46  34  RCL
47  01  1
48  97  PRT
49  98  PAP
50  41  R/S

51  04  4
52  57  SUBR
53  05  5
54  06  6
55  02  2
56  35  SUM
57  02  2
58  34  RCL
59  01  1
60  54  ÷
61  34  RCL
62  02  2

63  94  =
64  12  INV
65  29  INT
66  37  X=T
67  07  7
68  00  00
69  58  RTN
70  34  RCL
71  01  1
72  32  X⇄T
73  34  RCL
74  02  2

75  12  INV
76  30  PROD
77  01  1
78  37  X=T
79  04  4
80  08  8
81  97  PRT
82  56  CP
83  22  GTO
84  05  5
85  08  8
```

Bill also wrote a program called SR-56 LISTING ON A TI-59, which was used to produce the above. I will publish it in one of the future issues.

FAST-MODE CALENDAR.- Palmer O. Hansen of Largo, Florida, has seen fit to combine Panos Galidas' winning calendar program (v3n9p3,4 of 52-Notes) and Martin Neef's FAST MODE (v5n6p4) into a challenging entry for the HP vs TI calendar contest announced in v5n4/5pl. If any RPN programmer ever had illusions of equaling Panos' time of 2 min 43 sec to print out a full year calendar, it is going to be much more difficult from now on: Palmer's program does it in the impossible speed of 1 min 32 sec !!!!! The PC100 really moves out at this speed!



```
31.00000251331   08
29.00000211714   09
31.00000301335   10
30.00000133335   11
31.00000301345   12
30.00025413117   13
31.00025412745   14
31.00000134122   15
30.00036173337   16
31.00000321537   17
30.00000313242   18
31.00000161715   19
28.00000211714   20
98.99            22
1.00003600003    24
1.00000037       25
1.0043000037     26
1.000021000036   27
2.010000000002   28
2.010002000003   29
2.010003000004   30
2.010004000005   31
2.010005000006   32
2.010006000007   33
2.01000700001    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
2.01020700021    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
2.01030700031    54
2.010310000311   55
2.010311000312   56
2.010312000401   57
2.010401000402   58
2.010402         59
```

I once sent one of my "also ran" calendar programs, about 2 min 59 sec, to Display, were it was promtly published. In one of the subsequent issues I found out that one of the members had given me the accolade of "der Amerikanische Geschwindigkeitsfanatiker." (the American speed nut) I hate to find out what Palmer is going to be called!

This program is really an example of "friendliness towards the user". It can be run in FAST or in NORMAL mode. You can load the registers by means of long numbers or by kev strokes. And you enter the year and the number of years as one, single number: YYYY.n

Instructions to key in the program:

1. In 6 OP 17 partition, key in the program up to step 477.
2. Load registers 08 through 59 with the values shown at right.
   As a reminder, reg 08 is loaded as:
   2.51331 EE 06 +/- + 31 = STO 08.
3. Record all four sides on two mag cards.

OR, IF YOU PREFER THIS ALTERNATE METHOD:

1. In 0 OP 17, key in steps 000 through 895.
   Some of the codes have to be synthesized, of course, such as 21 or 31. The way to do it is STO 21 BST BST DEL SST....
2. In 6 OP 17 partition, record all four sides on two mag cards.

RUNNING THE PROGRAM IN FAST MODE: Instructions:(be very careful to follow these instructions, as any other sequence might cause the calculator to revert to NORMAL mode. Do not press RST nor CLR once the FAST mode has been acquired)

1. Enter bank 1. Display shows 1. Press RST R/S
   You are now in FAST mode.
2. Enter banks 1, 2, 3 and 4 in sequence. The printer will print each time the number of the bank entered. The display will always show a zero. So, DO NOT PRESS CLR before entering a bank.
3. To print out a calendar starting with January, enter YYYY.n with n equal to or smaller than 9. Press R/S.(n= number of years)
4. Print additional years after first printing is finished. Enter n and press R/S.
5. To print out a single month, enter M and press SBR 456. Display shows zero. Enter year YYYY and press R/S.
6. To print out a calendar starting at a month other than January, enter month M and press SBR 463. Enter first year and number of years as YYYY.n and press R/S.
7. To print out a calendar starting at January after steps 4, 5 or 6 have been used with n = < 9, enter YYYY.n and press SBR 051.

RUNNING THE PROGRAM IN NORMAL MODE: Instructions:

1. Load banks 1, 2, 3 and 4 in standard manner.
2. Clear any risidual modes such as FIX, ENG or EE and/or print registers. (OP 00) Initialize by pressing A. Printer prints a 1, display shows a zero.
3. To print out a calendar starting with January, enter YYYY.n and press B.
4. To print out a single month, enter month M and press C.
   Enter year YYYY and press R/S.
5. To print out a calendar starting with a month other than January, enter month M and press D. Enter first year and number of years as YYYY.n and press R/S.
   NOTE: in all the instructions above, n has to be equal or smaller than 9.

SOUR NOTE: 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.

```
000 00 0
001 00 0
002 00 0
003 00 0
004 00 0
005 36 PGM
006 02 02
007 71 SBR
008 02 02
009 39 39
010 09 9
011 00 0
012 76 LBL
013 11 A
014 69 OP
015 00 00
016 22 INV
017 58 FIX
018 22 INV
019 57 ENG
020 01 1
021 99 PRT
022 25 CLR
023 91 R/S
024 99 PRT
025 25 CLR
026 91 R/S
027 99 PRT
028 25 CLR
029 91 R/S
030 99 PRT
031 98 ADV
032 04 4
033 05 5
034 01 1
035 07 7
036 01 1
037 03 3
038 03 3
039 05 5
040 07 7
041 01 1
042 69 OP
043 03 03
044 69 OP
045 05 05
046 98 ADV
047 25 CLR
048 91 R/S
049 76 LBL
050 12 B
051 98 ADV
052 22 INV
053 86 STF
054 01 01
055 32 X:T
056 01 1
057 42 STO
058 01 01
059 32 X:T
060 42 STO
061 05 05
062 59 INT
063 42 STO
064 03 03
065 42 STO
066 06 06
067 22 INV
068 44 SUM
069 05 05
070 01 1
071 00 0
072 49 PRD
073 05 05

074 00 0
075 42 STO
076 02 02
077 53 (
078 03 3
079 06 6
080 05 5
081 65 ×
082 43 RCL
083 06 06
084 85 +
085 43 RCL
086 02 02
087 85 +
088 03 3
089 01 1
090 65 ×
091 43 RCL
092 01 01
093 75 -
094 03 3
095 01 1
096 85 +
097 03 3
098 32 X:T
099 43 RCL
100 01 01
101 77 GE
102 01 01
103 09 09
104 69 OP
105 36 36
106 61 GTO
107 01 01
108 23 23
109 53 (
110 93 .
111 04 4
112 65 ×
113 43 RCL
114 01 01
115 85 +
116 02 2
117 93 .
118 03 3
119 54 )
120 59 INT
121 94 +/-
122 85 +
123 53 (
124 43 RCL
125 06 06
126 55 ÷
127 04 4
128 54 )
129 59 INT
130 75 -
131 53 (
132 93 .
133 07 7
134 05 5
135 85 +
136 53 (
137 43 RCL
138 06 06
139 55 ÷
140 01 1
141 00 0
142 00 0
143 54 )
144 59 INT
145 65 ×
146 93 .
147 07 7

148 05 5
149 54 )
150 59 INT
151 54 )
152 75 -
153 53 (
154 24 CE
155 55 ÷
156 07 7
157 54 )
158 59 INT
159 65 ×
160 07 7
161 95 =
162 42 STO
163 02 02
164 07 7
165 44 SUM
166 01 01
167 43 RCL
168 20 20
169 42 STO
170 09 09
171 43 RCL
172 03 03
173 55 ÷
174 04 4
175 95 =
176 42 STO
177 04 04
178 22 INV
179 59 INT
180 29 CP
181 22 INV
182 67 EQ
183 03 03
184 52 52
185 69 OP
186 29 29
187 43 RCL
188 04 04
189 55 ÷
190 02 2
191 05 5
192 95 =
193 42 STO
194 04 04
195 22 INV
196 59 INT
197 22 INV
198 67 EQ
199 03 03
200 52 52
201 43 RCL
202 04 04
203 55 ÷
204 04 4
205 95 =
206 22 INV
207 59 INT
208 67 EQ
209 03 03
210 52 52
211 69 OP
212 39 39
213 61 GTO
214 03 03
215 52 52
216 44 SUM
217 00 00
218 73 RC*
219 00 00
220 82 HIR
221 05 05

222 44 SUM
223 00 00
224 73 RC*
225 00 00
226 82 HIR
227 06 06
228 43 RCL
229 22 22
230 82 HIR
231 36 36
232 69 OP
233 20 20
234 73 RC*
235 00 00
236 44 SUM
237 00 00
238 22 INV
239 59 INT
240 82 HIR
241 07 07
242 73 RC*
243 00 00
244 82 HIR
245 08 08
246 69 OP
247 05 05
248 97 DSZ
249 04 04
250 02 02
251 16 16
252 43 RCL
253 02 02
254 29 CP
255 67 EQ
256 03 03
257 37 37
258 44 SUM
259 00 00
260 69 OP
261 00 00
262 32 X:T
263 02 2
264 67 EQ
265 03 03
266 30 30
267 05 5
268 67 EQ
269 03 03
270 11 11
271 69 OP
272 20 20
273 06 6
274 67 EQ
275 03 03
276 03 03
277 01 1
278 67 EQ
279 02 02
280 97 97
281 03 3
282 67 EQ
283 02 02
284 91 91
285 73 RC*
286 00 00
287 52 EE
288 61 GTO
289 03 03
290 13 13
291 73 RC*
292 00 00
293 52 EE
294 61 GTO
295 03 03

296 21 21
297 73 RC*
298 00 00
299 52 EE
300 61 GTO
301 03 03
302 32 32
303 73 RC*
304 00 00
305 22 INV
306 44 SUM
307 00 00
308 52 EE
309 82 HIR
310 08 08
311 73 RC*
312 00 00
313 22 INV
314 59 INT
315 82 HIR
316 07 07
317 69 OP
318 30 30
319 73 RC*
320 00 00
321 22 INV
322 44 SUM
323 00 00
324 85 +
325 43 RCL
326 22 22
327 95 =
328 82 HIR
329 06 06
330 73 RC*
331 00 00
332 82 HIR
333 05 05
334 69 OP
335 05 05
336 25 CLR
337 98 ADV
338 87 IFF
339 01 01
340 04 04
341 73 73
342 43 RCL
343 01 01
344 32 X:T
345 01 1
346 09 9
347 67 EQ
348 04 04
349 38 38
350 69 OP
351 21 21
352 02 2
353 09 9
354 42 STO
355 00 00
356 73 RC*
357 01 01
358 59 INT
359 42 STO
360 07 07
361 85 +
362 43 RCL
363 02 02
364 75 -
365 53 (
366 24 CE
367 55 ÷
368 07 7
369 54 )

370 59 INT
371 42 STO
372 04 04
373 65 ×
374 07 7
375 95 =
376 32 X:T
377 73 RC*
378 01 01
379 82 HIR
380 08 08
381 43 RCL
382 03 03
383 69 OP
384 06 06
385 43 RCL
386 24 24
387 82 HIR
388 05 05
389 43 RCL
390 25 25
391 82 HIR
392 06 06
393 43 RCL
394 26 26
395 82 HIR
396 07 07
397 43 RCL
398 27 27
399 82 HIR
400 08 08
401 69 OP
402 05 05
403 69 OP
404 00 00
405 32 X:T
406 48 EXC
407 02 02
408 32 X:T
409 05 5
410 67 EQ
411 02 02
412 42 42
413 03 3
414 67 EQ
415 02 02
416 34 34
417 02 2
418 67 EQ
419 02 02
420 24 24
421 00 0
422 67 EQ
423 02 02
424 18 18
425 69 OP
426 30 30
427 06 6
428 67 EQ
429 02 02
430 42 42
431 04 4
432 67 EQ
433 02 02
434 34 34
435 61 GTO
436 02 02
437 18 18
438 69 OP
439 23 23
440 08 8
441 42 STO
442 01 01
443 97 DSZ

444 05 05
445 01 01
446 67 67
447 25 CLR
448 91 R/S
449 42 STO
450 05 05
451 61 GTO
452 01 01
453 67 67
454 76 LBL
455 13 C
456 86 STF
457 01 01
458 61 GTO
459 04 04
460 66 66
461 76 LBL
462 14 D
463 22 INV
464 86 STF
465 01 01
466 42 STO
467 01 01
468 25 CLR
469 91 R/S
470 61 GTO
471 00 00
472 60 60
473 22 INV
474 86 STF
475 01 01
476 25 CLR
477 91 R/S
478 00 0
479 00 0
480 00 0
481 00 0
482 00 0
483 00 0
484 20 CLR
485 40 IND
486 10 E'
487 20 CLR
488 00 0
489 20 CLR
490 40 IND
491 00 0
492 10 E'
493 40 IND
494 10 E'
495 20 CLR
496 00 0
497 10 E'
498 40 IND
499 00 0
500 20 CLR
501 31 LRN
502 10 E'
503 20 CLR
504 00 0
505 20 CLR
506 31 LRN
507 00 0
508 10 E'
509 31 LRN
510 10 E'
511 20 CLR
512 00 0
513 10 E'
514 31 LRN
515 00 0
516 00 0
517 31 LRN

518 10 E'
519 20 CLR
520 00 0
521 00 0
522 31 LRN
523 00 0
524 70 RAD
525 30 TAN
526 10 E'
527 20 CLR
528 00 0
529 70 RAD
530 30 TAN
531 00 0
532 60 DEG
533 30 TAN
534 10 E'
535 20 CLR
536 00 0
537 60 DEG
538 30 TAN
539 00 0
540 50 |X|
541 30 TAN
542 10 E'
543 20 CLR
544 00 0
545 50 |X|
546 30 TAN
547 00 0
548 40 IND
549 30 TAN
550 10 E'
551 20 CLR
552 00 0
553 40 IND
554 30 TAN
555 00 0
556 30 TAN
557 30 TAN
558 10 E'
559 20 CLR
560 00 0
561 30 TAN
562 30 TAN
563 00 0
564 20 CLR
565 30 TAN
566 10 E'
567 20 CLR
568 00 0
569 20 CLR
570 30 TAN
571 00 0
572 10 E'
573 30 TAN
574 10 E'
575 20 CLR
576 00 0
577 10 E'
578 30 TAN
579 00 0
580 20 CLR
581 21 2ND
582 10 E'
583 20 CLR
584 00 0
585 20 CLR
586 21 2ND
587 00 0
588 10 E'
589 21 2ND
590 10 E'

591 20 CLR
592 00 0
593 10 E'
594 21 2ND
595 00 0
596 00 0
597 21 2ND
598 10 E'
599 20 CLR
600 00 0
601 00 0
602 21 2ND
603 00 0
604 70 RAD
605 20 CLR
606 10 E'
607 20 CLR
608 00 0
609 70 RAD
610 20 CLR
611 00 0
612 60 DEG
613 20 CLR
614 10 E'
615 20 CLR
616 00 0
617 60 DEG
618 20 CLR
619 00 0
620 50 |X|
621 20 CLR
622 10 E'
623 20 CLR
624 00 0
625 50 |X|
626 20 CLR
627 00 0
628 40 IND
629 20 CLR
630 10 E'
631 20 CLR
632 00 0
633 40 IND
634 20 CLR
635 00 0
636 30 TAN
637 20 CLR
638 10 E'
639 20 CLR
640 00 0
641 30 TAN
642 20 CLR
643 00 0
644 20 CLR
645 20 CLR
646 10 E'
647 20 CLR
648 00 0
649 20 CLR
650 20 CLR
651 00 0
652 10 E'
653 20 CLR
654 10 E'
655 20 CLR
656 00 0
657 10 E'
658 20 CLR
659 00 0
660 20 CLR
661 01 1
662 10 E'
663 20 CLR

664 00 0
665 20 CLR
666 01 1
667 00 0
668 10 E'
669 01 1
670 10 E'
671 20 CLR
672 00 0
673 10 E'
674 01 1
675 00 0
676 00 0
677 01 1
678 10 E'
679 20 CLR
680 00 0
681 00 0
682 01 1
683 00 0
684 70 RAD
685 00 0
686 10 E'
687 20 CLR
688 00 0
689 70 RAD
690 00 0
691 00 0
692 60 DEG
693 00 0
694 10 E'
695 20 CLR
696 00 0
697 60 DEG
698 00 0
699 00 0
700 50 |X|
701 00 0
702 10 E'
703 20 CLR
704 00 0
705 50 |X|
706 00 0
707 00 0
708 40 IND
709 00 0
710 10 E'
711 20 CLR
712 00 0
713 40 IND
714 00 0
715 00 0
716 30 TAN
717 00 0
718 10 E'
719 20 CLR
720 00 0
721 30 TAN
722 00 0
723 00 0
724 20 CLR
725 00 0
726 10 E'
727 20 CLR
728 00 0
729 20 CLR
730 00 0
731 00 0
732 00 0
733 00 0
734 10 E'
735 20 CLR
736 00 0

737 60 DEG
738 03 3
739 00 0
740 10 E'
741 02 2
742 00 0
743 10 E'
744 00 0
745 00 0
746 70 RAD
747 03 3
748 00 0
749 30 TAN
750 04 4
751 10 E'
752 00 0
753 00 0
754 00 0
755 70 RAD
756 03 3
757 00 0
758 00 0
759 10 E'
760 00 0
761 00 0
762 03 3
763 00 0
764 60 DEG
765 03 3
766 00 0
767 10 E'
768 00 0
769 00 0
770 00 0
771 00 0
772 00 0
773 00 0
774 00 0
775 00 0
776 10 E'
777 00 0
778 00 0
779 00 0
780 00 0
781 00 0
782 99 PRT
783 98 ADV
784 00 0
785 00 0
786 00 0
787 00 0
788 00 0
789 00 0
790 00 0
791 00 0
792 10 E'
793 40 IND
794 71 SBR
795 11 A
796 02 2
797 00 0
798 00 0
799 28 LOG
800 10 E'
801 50 |X|
802 71 SBR
803 61 GTO
804 01 1
805 00 0
806 00 0
807 31 LRN
808 10 E'
809 20 CLR
810 24 CE

811 13 C
812 03 3
813 00 0
814 00 0
815 30 TAN
816 10 E'
817 70 RAD
818 53 (
819 21 2ND
820 03 3
821 00 0
822 00 0
823 31 LRN
824 10 E'
825 70 RAD
826 33 X²
827 73 RC*
828 61 61
829 03 3
830 00 0
831 30 TAN
832 10 E'
833 20 CLR
834 12 B
835 34 √X
836 01 1
837 00 0
838 00 0
839 31 LRN
840 10 E'
841 50 |X|
842 74 SM*
843 12 12
844 54 )
845 02 2
846 00 0
847 31 LRN
848 10 E'
849 70 RAD
850 11 A
851 13 C
852 54 )
853 02 2
854 00 0
855 30 TAN
856 10 E'
857 50 |X|
858 34 √X
859 01 1
860 03 3
861 00 0
862 00 0
863 31 LRN
864 10 E'
865 50 |X|
866 33 X²
867 33 X²
868 01 1
869 00 0
870 00 0
871 30 TAN
872 10 E'
873 50 |X|
874 33 X²
875 01 1
876 03 3
877 00 0
878 00 0
879 31 LRN
880 10 E'
881 40 IND
882 71 SBR
883 11 A
884 02 2

885 00 0
886 00 0
887 29 CP
888 10 E'
889 10 E'
890 33 X²
891 51 BST
892 02 2
893 00 0
894 00 0
895 31 LRN
896 00 0
897 00 0
898 00 0
899 00 0
900 00 0
901 00 0
902 00 0
903 00 0
904 00 0
905 00 0
906 00 0
907 00 0
908 00 0
909 00 0
910 00 0
911 00 0
912 00 0
913 00 0
914 00 0
915 00 0
916 00 0
917 00 0
918 00 0
919 00 0
920 00 0
921 00 0
922 00 0
923 00 0
924 00 0
925 00 0
926 00 0
927 00 0
928 00 0
929 00 0
930 00 0
931 00 0
932 00 0
933 00 0
934 00 0
935 00 0
936 00 0
937 00 0
938 00 0
939 00 0
940 00 0
941 00 0
942 00 0
943 00 0
944 00 0
945 00 0
946 00 0
947 00 0
948 00 0
949 00 0
950 00 0
951 00 0
952 00 0
953 00 0
954 00 0
955 00 0
956 00 0
957 00 0
958 00 0
959 00 0
```

OP 40.- Did you know that the TI-58C has an extra OP-code, OP 40 ? It sets flag 7 on printer. It is mentioned on page V-27 of the latest manual.

PROGRAM LISTING WITHOUT PAPER.- Martin Neef (ZEPRA, West Germany) passes along this useful trick:

With a program in banks 1 and/or 2 only, and with the ML-module installed, press:

8 STO 53 PGM 19 SBR 172 Then within the next second HOLD DOWN the GTO key.

The program will be listed in the display window, without it being printed. The listing speed is such that an experienced programmer may easily scan his/her own program and detect errors.
The attached printer will always print step 000, but nothing more. This trick will not work without the printer being attached, however.
Martin further says that, as a side quirk, this trick always deposits a code 85 (+) in step 000 of your program. I confirm this. But Martin also claims that a code 43 (RCL) will be deposited on the step the calculator will start the listing, that is at the number that happened to be in the display. That part I could not make my calculator do. It left the rest of my program entirely intact.
I found that, if you want to start the listing at the lowest step possible, it is wise to press a CLR after the "53" in the routine above. Then, enter the routine with your right hand, while holding one finger of your left hand on the GTO key, ready to depress that key right after entering the "2" in your routine.
South paws, please disregard my bias and read "right hand" where I say "left hand" and vice versa.
On steps 172-173 of PGM 19 there is an OP IND 53. By entering an "8", you actually perform an OP 08. But how that misterious ML-module interprets OP 08 (=list labels) to do a normal listing is beyond me!
Don't expect any mnemonics in this type of listing, of course.
One question: Is it possible to find something similar in other modules? Please let me know if you find something worthwhile.

TANH ANYONE? - Re-v5n4/5p25. Paul Berg says that, even it is true that the calculator will be in radian mode when done, the answer will NOT be in radians unless entry was also in radians.

MATH-UTILITY MODULE- Re-v5n4/5p11. Paul Berg did some more sleuthing and found the following:

PGM 02 SBR 313	prints the word	FLOW (and NOT SLOW as I stated)
PGM 02 SBR 109	....	R/S
PGM 02 SBR 144	....	READY READY (in two lines)
PGM 02 SBR 156	....	REPEAT REPEAT
PGM 02 SBR 172	....	RESULT RESULT
PGM 02 SBR 198	....	OPTION OPTION
PGM 02 SBR 224	....	BAD COMMAND BAD COMMAND
PGM 02 SBR 283	....	UNDERFLOW UNDERFLOW
PGM 02 SBR 297	....	OVERFLOW OVERFLOW
PGM 02 SBR 242	....	BAD MA
PGM 02 SBR 262	....	BAD
PGM 02 SBR 087	....	PRESS RST R/S

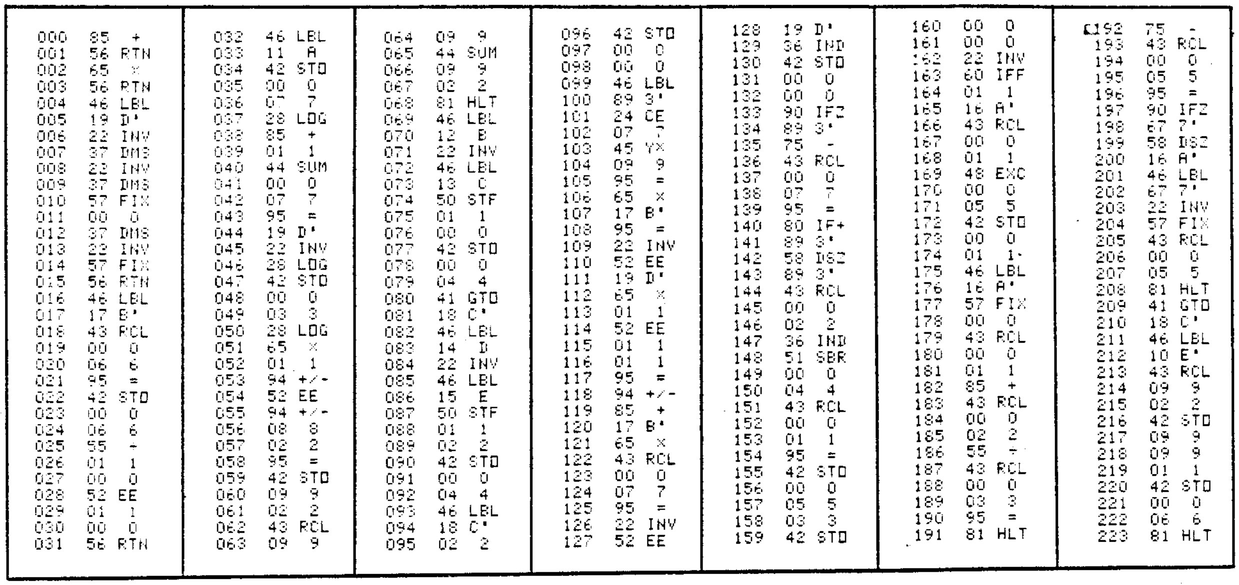
If the desired code, up to max 5 characters, is in the display at the time of call of PGM 02 SBR 262, you can print a two-word message, of which the first word is BAD. For example, enter 16171327 and press PGM 02 SBR 262, to see BAD DEAL printed.

Fred Fish, whom you will remember as the author who unraveled all the misteries of the ML-module and wrote "the bible" on it, declined to become a member of our club. He says he will be very busy persuing a degree at Arizona State University. Maybe someone else among our large membership might want to write the bible on the M/U module? I am convinced he/she will become famous and make a lot of money in the process. Count on me to tell everybody about it, once it is written.

ARITHMETIC EXERCISES on the SR-52.- This program by Dean Athans allows you to exercise your arithmetic, by supplying random problems. It uses numbers in a range you can predetermine and it allows you to prescribe also the function:addition, subtraction, multiplication or division. The program gives you three chances to produce the correct answer, after which it judges you to be too dumb anyway and supplies the correct answer itself!!!
This program demonstrates the use of dynamic code. That means, storing a number in a register (here 92) which is part of the user's memory. Each separate digit or pair of digits will then be executed as program step commands.
The instructions are:

1. Initialize, press 2nd E'.
2. Enter the max value of n, the number you want to exercise with and press A.
   You will now see -4.385010043-71 in the display. Ignore this number.
3. For addition press B
   subtraction C
   multiplication D
   division E
   You will now see the problem in the display in the form m.n
   This means "add m to n" or "multiply m with n", etc. depending on if you pressed B, C, D or E.
4. Enter your anwer and press RUN.
   If your answer is INCORRECT, the calculator will display m.n up to three times.
   After the third attempt it will display the correct answer itself.
5. To see the next problem, press RUN.

NOTES: The max value n (step 2) and/or the problem type (step3) may be changed only after the calculator displays the correct answer.(step 4)
The program needs to be initialized only once. DO NOT initialize again during or after an exercise. Only after reading in the program at the beginning.



```
000 85 +    032 46 LBL  064 09 9    096 42 STO  128 19 D'   160 00 0    192 75 -
001 56 RTN  033 11 A    065 44 SUM  097 00 0    129 36 IND  161 00 0    193 43 RCL
002 65 ×    034 42 STO  066 09 9    098 00 0    130 42 STO  162 22 INV  194 00 0
003 56 RTN  035 00 0    067 02 2    099 46 LBL  131 00 0    163 60 IFF  195 05 5
004 46 LBL  036 07 7    068 81 HLT  100 89 3'   132 00 0    164 01 1    196 95 =
005 19 D'   037 28 LOG  069 46 LBL  101 24 CE   133 90 IFZ  165 16 A'   197 90 IFZ
006 22 INV  038 85 +    070 12 B    102 07 7    134 89 3'   166 43 RCL  198 67 7'
007 37 DMS  039 01 1    071 22 INV  103 45 YX   135 75 -    167 00 0    199 58 DSZ
008 22 INV  040 44 SUM  072 46 LBL  104 09 9    136 43 RCL  168 01 1    200 16 A'
009 37 DMS  041 00 0    073 13 C    105 95 =    137 00 0    169 48 EXC  201 46 LBL
010 57 FIX  042 07 7    074 50 STF  106 65 ×    138 07 7    170 00 0    202 67 7'
011 00 0    043 95 =    075 01 1    107 17 B'   139 95 =    171 05 5    203 22 INV
012 37 DMS  044 19 D'   076 00 0    108 95 =    140 80 IF+  172 42 STO  204 57 FIX
013 22 INV  045 22 INV  077 42 STO  109 22 INV  141 89 3'   173 00 0    205 43 RCL
014 57 FIX  046 28 LOG  078 00 0    110 52 EE   142 58 DSZ  174 01 1    206 00 0
015 56 RTN  047 42 STO  079 04 4    111 19 D'   143 89 3'   175 46 LBL  207 05 5
016 46 LBL  048 00 0    080 41 GTO  112 65 ×    144 43 RCL  176 16 A'   208 81 HLT
017 17 B'   049 03 3    081 18 C'   113 01 1    145 00 0    177 57 FIX  209 41 GTO
018 43 RCL  050 28 LOG  082 46 LBL  114 52 EE   146 02 2    178 00 0    210 18 C'
019 00 0    051 65 ×    083 14 D    115 01 1    147 36 IND  179 43 RCL  211 46 LBL
020 06 6    052 01 1    084 22 INV  116 01 1    148 51 SBR  180 00 0    212 10 E'
021 95 =    053 94 +/-  085 46 LBL  117 95 =    149 00 0    181 01 1    213 43 RCL
022 42 STO  054 52 EE   086 15 E    118 94 +/-  150 04 4    182 85 +    214 09 9
023 00 0    055 94 +/-  087 50 STF  119 85 +    151 43 RCL  183 43 RCL  215 02 2
024 06 6    056 08 8    088 01 1    120 17 B'   152 00 0    184 00 0    216 42 STO
025 55 ÷    057 02 2    089 02 2    121 65 ×    153 01 1    185 02 2    217 09 9
026 01 1    058 95 =    090 42 STO  122 43 RCL  154 95 =    186 55 ÷    218 09 9
027 00 0    059 42 STO  091 00 0    123 00 0    155 42 STO  187 43 RCL  219 01 1
028 52 EE   060 09 9    092 04 4    124 07 7    156 00 0    188 00 0    220 42 STO
029 01 1    061 02 2    093 46 LBL  125 95 =    157 05 5    189 03 3    221 00 0
030 00 0    062 43 RCL  094 18 C'   126 22 INV  158 03 3    190 95 =    222 06 6
031 56 RTN  063 09 9    095 02 2    127 52 EE   159 42 STO  191 81 HLT  223 81 HLT
```

BUILDING_SYSTEMS_DESIGN_WITH_PROGRAMMABLE_CALCULATORS.- Sital Daryanani, Mc. Graw-Hill, New York NY, 10020. USA.
This book contains 22 programs that address commenly encountered problems in design of building systems including air, liquid and gas piping, lighting, solar shading and radiation systems. Available for $ 29.50 US from the publisher.
Alan Morris, of Bethesda, MD, brought this book to my attention.

PRESSURE_VESSEL_PROGRAMS.- Prentex, P.O. Box 223583, Dallas TX. 75222, tel.214/748-7837, produced a series of programs applicable to the design and construction of pressure vessels, such as, e.g. Radiography and Joint Efficiency, Size of Welds, nozzle Reinforcement, and others. The programs come recorded on mag cards. One of our very active members, Clyde Durbin, collaborated on the development. Prices range from $ 30.00 to $ 60.00 US per individual program.

BRANCHING FROM THE KEY BOARD DURING PROGRAM EXECUTION.- Martin Neef, Zepra-club in West Germany, discovered this very useful trick. Here follows then a translation of what he writes. I have kept the wording as litteral as possible, at the risk of sounding too Germanic once in a while:

« It is very well known that the R/S command will normally start and stop a program in user memory, except when you call a module program from the key board, when the next R/S command will start the module program, rather than the user program. When during program execution a call to a module with PGM xx is encountered, according to the manual, you have to have a SBR call with A through E', or a SBR N or SBR nnn. If you call anything else, the program goes automatically to PGM 00, that is the program in user memory.

One exception to this rule here is the R/S command, which provokes a starting of the module program, beginning at the step the program pointer happens to be at. These R/S calls have to be placed in the program immediately after the program call PGM xx.

By means of a trick it is now possible to avoid that the call to PGM xx would be cancelled by the subsequent commands. That means that the call to a SBR or R/S does not have to follow immediately after the PGM xx call. This also means that an R/S from the key board, during program execution, does not constitute a program interruption anymore of the user program, but it means the start of a module program.

The trick I am alluding to consists of placing immediately after the PGM xx call one of the codes 21, 41 or 51, with 51 being preferred for its lack of side effects. (21 is the code for 2nd 2nd, 41=SST and 51=BST, ed)

As long as after this call you have neither a SBR nor a RST, the R/S call from the key board, during program execution, will start the module program.

A practical application results from this sequence:

```
0 STO 00  PGM 01  SBR 098  PGM 01  51 ......LBL A......(ML module)
```

After execution of ...PGM 01 51 ... you may branch to LBL A with a R/S. That means that you can make decisions without stopping the program, which constitutes a rather "friendly" style of programming. This trick would allow you to write a confortable reaction test program. Note, however, that each call opens up two more SBR levels.

It is clear that this method has so many new possiblities that it would be rather difficult to mention them all. I will leave it to the reader to do some further exploration. Happy searching.(Martin Neef) »

And this is exactly what Richard Snow did, searching. He writes:

« The results can be quite frustrating if you don't know what to expect. Please note the following:

1. Any subroutine call in user memory (SBR A through E') will call the subroutine in the library module.
2. If R/S is held down until the module routine is finished, then the user program will stop.
3. A RTN in user memory will cancel the PGM xx 51 call.
4. A R/S in the user program will not stop the program, but will start the module program instead.

As an example, steps 547 through 567 of ML-19 can be used to set a flag. (flags 1, 2, 3, 4, ed.) A user program can then be made to branch after the R/S key is pressed and the corresponding flag is checked. The problem, however, is that LBLs A' through D' (on steps 547 through 567, ed) are such short routines, that it is difficult not to stop the user program when R/S is pressed. »

Richard then encloses an enhanced version of his brother's TIME BOMBS game. This program appeared originally as PPX # 918029B. The key board branching trick was added to reduce the possiblity of cheating. It should be noted that this program is intended only as a demonstration of the use of the key board branching trick. Press R/S to stop the program. Then change your mind about cutting a particular wire and press R/S again to continue. Unfortunately, if you hold down R/S a little too long, it stops the program for good. Perhaps a longer routine than LBL D' PGM 19 BST (see steps 028-032 of TIME BOMBS) can be found in one of the library modules which will do the job more reliably.

You will find TIME BOMBS somewhere is this issue.

```
         TIME BOMBS

    RED
  BLACK
  BLACK WIRE CUT
             BANG!

         TRY AGAIN!
  BLACK
    RED
  GREEN
  GREEN WIRE CUT
  BROWN
  WHITE
  WHITE WIRE CUT
               DUD
     2 BOMBS LEFT

  BROWN
  BROWN
  BROWN WIRE CUT
       BOMB IS DEFUSED
     1 BOMB LEFT

  GREEN
  BLACK
   PINK
  BROWN
  BROWN WIRE CUT
             BANG!

         TRY AGAIN!
```

```
   ALPHA REG LIST:

     NUMERIC  REG ALPHA
  4323243717.  60 WHITE
    33243126.  61  PINK
  1435324331.  62 BROWN
    14274117.  63  BLUE
  1427131526.  64 BLACK
  2235171731.  65 GREEN
      351716.  66   RED
    27172137.  67  LEFT
    15413700.  68   CUT
    43243517.  69  WIRE
```

```
000  29 CP
001  43 RCL
002  09  09
003  67  EQ
004  00  00
005  75  75
006  61 GTO
007  00  00
008  82  82
009  76 LBL
010  17 B'
011  69 OP
012  03  03
013  69 OP
014  05  05
015  69 OP
016  00  00
```

START

PRINTER

```
017  92 RTN
018  76 LBL
019  18 C'
020  05  5
021  16 A'
022  73 RC*
023  21  21
024  22 INV
025  67  EQ
026  18 C'
027  92 RTN
028  76 LBL
029  19 D'
030  36 PGM
031  19  19
032  51 BST
033  42 STO
```

WIRE CONTROL

ST* 4 IF R/S

```
034  08  08
035  42 STO
036  22  22
037  06  6
038  00  0
039  44 SUM
040  22  22
041  73 RC*
042  22  22
043  69 OP
044  01  01
045  69 OP
046  05  05
047  43 RCL
048  07  07
049  22 INV
050  97 DSZ
051  07  07
052  00  00
053  74  74
054  66 PAU
055  87 IFF
056  04  04
057  13 C
058  97 DSZ
059  08  08
060  00  00
061  47  47
062  07  7
063  76 LBL
064  16 A'
065  82 HIR
066  44  44
067  82 HIR
068  14  14
069  59 INT
070  82 HIR
071  54  54
072  42 STO
073  21  21
074  92 RTN
075  05  5
076  06  6
077  42 STO
078  07  07
079  03  3
080  42 STO
081  09  09
082  04  4
083  36 PGM
084  01  01
085  71 SBR
086  00  00
087  12  12
088  42 STO
089  00  00
090  05  5
091  16 A'
092  22 INV
093  67  EQ
094  00  00
095  98  98
096  86 STF
097  01  01
098  05  5
099  16 A'
100  01  1
101  72 ST*
102  21  21
103  18 C'
104  02  2
105  72 ST*
106  21  21
```

SECS

PRINT COLOR OF WIRE

CHECK DISPLAY TIME REMAINING

SECS BTW WIRES / CUT

GENERATOR

TOT. SECS

TOT. BOMBS

CLEAN OUT REGS 0→4

MAKE DUD

EXPLODE

DEFUSE

```
107  18 C'
108  03  3
109  72 ST*
110  21  21
111  36 PGM
112  19  19
113  71 SBR
114  05  05
115  62  62
116  07  7
117  16 A'
118  19 D'
119  87 IFF
120  04  04
121  13 C
122  97 DSZ
123  07  07
124  01  01
125  18  18
126  69 OP
127  00  00
128  87 IFF
129  01  01
130  02  02
131  86  86
132  01  1
133  04  4
134  01  1
135  03  3
136  03  3
137  01  1
138  02  2
139  02  2
140  07  7
141  03  3
142  17 B'
143  05  5
144  42 STO
145  09  09
146  00  0
147  35 1/X
148  66 PAU
149  97 DSZ
150  09  09
151  01  01
152  48  48
153  25 CLR
154  98 ADV
155  03  3
156  07  7
157  03  3
158  05  5
159  04  4
160  05  5
161  00  0
162  00  0
163  01  1
164  03  3
165  69 OP
166  02  02
167  02  2
168  02  2
169  01  1
170  03  3
171  02  2
172  04  4
173  03  3
174  01  1
175  07  7
176  03  3
177  17 B'
178  81 RST
179  76 LBL
```

DO NOTHING

INIT. PGM 19

CUT WIRE? FIRST WIRE

TIME REMAINING

BANG!

FLASHER

TRY AGAIN!

```
180  13  C
181  22 INV
182  86 STF
183  04  04
184  43 RCL
185  69  69
186  69 OP
187  02  02
188  43 RCL
189  68  68
190  17 B'
191  43 RCL
192  07  07
193  39 COS
194  82 HIR
195  34  34
196  82 HIR
197  14  14
198  59 INT
199  82 HIR
200  54  54
201  73 RC*
202  21  21
203  67  EQ
204  02  02
205  33  33
206  32 X:T
207  01  1
208  67  EQ
209  01  01
210  28  28
211  02  2
212  67  EQ
213  02  02
214  53  53
215  03  3
216  67  EQ
217  02  02
218  25  25
219  87  IFF
220  02  02
221  02  02
222  53  53
223  86 STF
224  03  03
225  29 CP
226  04  4
227  22 INV
228  44 SUM
229  07  07
230  61 GTO
231  01  01
232  11  11
233  87 IFF
234  03  03
235  01  01
236  28  28
237  86 STF
238  02  02
239  43 RCL
240  68  68
241  69 OP
242  02  02
243  43 RCL
244  69  69
245  69 OP
246  04  04
247  43 RCL
248  65  65
249  17 B'
250  61 GTO
251  02  02
252  25  25
```

WIRE CUT

RANDOMIZE

PRIMARY WIRE CUT?

EXPLODE

DEFUSE

DO NOTHING

LOGIC FOR GREEN WIRE

SUBTRACT 4 SECS

CUT GREEN WIRE

```
253  43 RCL
254  58  58
255  75  -
256  03  3
257  06  6
258  95  =
259  69 OP
260  02  02
261  02  2
262  01  1
263  04  4
264  01  1
265  03  3
266  06  6
267  01  1
268  07  7
269  01  1
270  06  6
271  69 OP
272  04  04
273  02  2
274  04  4
275  03  3
276  06  6
277  00  0
278  00  0
279  01  1
280  06  6
281  01  1
282  07  7
283  61 GTO
284  02  02
285  92  92
286  01  1
287  06  6
288  04  4
289  01  1
290  01  1
291  06  6
292  17 B'
293  43 RCL
294  09  09
295  97 DSZ
296  09  09
297  03  03
298  26  26
299  43 RCL
300  59  59
301  69 OP
302  02  02
303  43 RCL
304  67  67
305  17 B'
306  03  3
307  06  6
308  01  1
309  07  7
310  01  1
311  05  5
312  04  4
313  00  0
314  69 OP
315  04  04
316  43 RCL
317  07  07
318  50 I×I
319  69 OP
320  06  06
321  69 OP
322  00  00
323  61 GTO
324  01  01
325  54  54
```

BOMB IS DEFUSED

DUD

NEXT BOMB

TIME LEFT

SEC.

PRINT SECS LEFT

```
326  65  ×
327  32 X:T
328  01  1
329  00  0
330  00  0
331  95  =
332  69 OP
333  01  01
334  02  2
335  22 INV
336  77  GE
337  03  03
338  44  44
339  58 FIX
340  02  02
341  03  3
342  06  6
343  75  -
344  43 RCL
345  58  58
346  95  =
347  69 OP
348  02  02
349  43 RCL
350  67  67
351  17 B'
352  22 INV
353  58 FIX
354  98 ADV
355  81 RST
356  76 LBL
357  11  A
358  60 DEG
359  04  4
360  42 STO
361  05  05
362  01  1
363  42 STO
364  06  06
365  07  7
366  69 OP
367  17  17
368  39 COS
369  82 HIR
370  04  04
371  03  3
372  07  7
373  02  2
374  04  4
375  03  3
376  00  0
377  01  1
378  07  7
379  00  0
380  00  0
381  42 STO
382  59  59
383  69 OP
384  02  02
385  01  1
386  04  4
387  03  3
388  02  2
389  03  3
390  00  0
391  01  1
392  04  4
393  03  3
394  06  6
395  42 STO
396  58  58
397  17 B'
398  98 ADV
399  81 RST
```

PRINT NUMBER OF BOMBS LEFT

TIME

BOMBS

BOOK REVIEW- I recently read THE MICRO MILLENIUM by Christoffer Evans, The Viking Press, New York, 1980. It left me simply fascinated. This book is written by a pschycologists who had a lot of computer experience. The book is about the future. Not some distant future we can conveniently ignore, but about one which is imminent. It will involve a transformation of world society at all levels. And while taking place slowly at first, it will gather pace with sudden force. It is a future molded by a single piece of startling development in technology, the microcomputer. The author leads you through the past, the present, suddenly into the very near future. You, who have been in the thick of this revolution and have seen all the signs on the wall, will still, be fascinated by what is in store. I highly recommend this book as THE ONE to curl up with on one of those rainy autumn evenings. You will not be able to put it down before you finish it, from cover to cover. It is just mind boggling!

NEWCOMER'S CORNER.- POSTAL RATES.- A nice vehicle to learn programming is trying to mechanize the tables of the Post Office.
We will attempt to put together a program on one mag card that computes:

1. Priority mail or first class, internal US, Canada and Mexico, max 12 ounces.
2. Third class (printed matter) US, Canada and Mexico, max 16 ounces.
3. Air mail first class, overseas, two different categories.
4. So-called AO mail, i.e. third class airmail, three different categories.

If we get all that on one single card, it will be quite an achievement. It is, of course, also possible to expand each type, like first class for example, and also cover weights larger than 12 ounces. This way you could make a special card for each type. In any case, this attempt will serve only as a basis to learn how to attack such a problem.

First class or priority mail.- The Post Office defines this type as weighing 12 ounces or less, except postal or post cards. The fee is $ 0.15 for the first ounce or fraction of an ounce and $ 0.13 for each additional or fraction of an ounce.

The algorithm for it would sound about like this:

1. Test to see if the entered weight in ounces is larger than 12. If YES, print an appropriate message. If NO, go to 2 below.
2. Print the number of ounces in FIX 2 format.
3. Always assume that there is at least one ounce or a fraction of an ounce. Thus charge at least $ 0.15.
4. The total number of ounces to be charged is always equal to the integer part of the number of ounces entered plus one. Thus, 1.7 = 1 + 1 = 2 and 0.35 = 0 + 1 = 1. As we took already care of the one we added, by assuming at least one ounce and charging a minimum of $ 0.15 in 3, above, it suffices to multiply the integer part of the number of ounces by $ 0.13 and add this to the $ 0.15 already charged, to obtain the complete charge.
5. Print charge with appropriate descriptor in the margin.
   In case the printer is not used, make sure that the final charge is in the display when the program halts, so that it may be read out.

A first attempt to program this algorithm could look as follows:

```
LBL B  CMS  X:T  12  INV GE  PRT  {  X:T  32 46  OP 04  X:T  STO 00 } { FIX 2  OP
06 }.15 { STO 01  RCL 00} INT  X  .13  {  =  SUM 01  16 27  OP 04  RCL 01} { OP
06} R/S  LBL PRT  0  1/X  0  R/S
```

The special brackets { and } merely indicate subroutines used in the final program and can be ignored for now.
If you try this sequence you will see that it works, except for one border value: whole numbers. Thus, if we enter, for example 0.99 ounces it will compute correctly $ 0.15. But if we enter 1.00 ounces we still should get $ 0.15, because the rules say: first ounce or fraction...We can skirt around this difficulty easily by subtracting from the weight register,R00,a very small value, say 10^{-9}. This can be done in the program within the SBR brackets STO 01 RCL 00 - 9 INV LOG 1/x = on line two. This way is shorter than writing - 1 EE 9 +/- = INV EE and surely shorter than 0.000 000 001 !!! So, press GTO 029 LRN and make 6 spaces by pressing INS 6 times. Then key in: - 9 INV LOG 1/x =

Third class or printed matter.- The rates for this class are $ 0.20 per two ounces up to four ounces. Then, with a max of sixteen ounces,$ 0.13 each additional two ounces.

So, the first thing we have to do is, to test if it is over 16 ounces, the same way we did for first class over 12 ounces. Next we print the number of ounces, in FIX 2 format, with OZ in the margin. Again, we can assume that the charge will always be at least $ 0.20 for the first two ounces. So, we store that value in R01 and subtract 2 ounces from the weight in R00 by saying 2 INV SUM 00. Then we divide the number of remaining ounces in R00 by 2, add 1 to it and take the integer value. This will give us the correct whole "two-ounces" remaining to be charged at $ 0.13 per two ounces.

(over)

Note that again we run in the trouble that whole number of ounces entered make the program skip to the next higher charge, when in reality the Post Office defined that this should not be the case, until the entered number exceeds the whole number, even by a slight margin. So, we resort again to the trick of subtracting a very small amount by keying : - 9 INV LOG 1/x = . The following sequence does the trick. You will note that again I have used the special brackets to denote repeating sequences which are candidates for subroutines. Note also that we can make subroutines not only repeating sequences from this routine but also from first class routine B, so that the subroutines may be shared by both routines and possibly others following.

```
LBL E  CMS  X:T  16 INV GE  PRT  { 32 46 OP 04 X:T  STO 00}  { FIX 2  OP 06 }
{  - 9 INV LOG 1/X = DIV 2 + 1 =INT STO 00 2 {- 9 INV LOG 1/X =} X:T  RCL 00
GE  GTO  X  .2  { = SUM 01  16 27  OP 04   RCL 01} {OP 06 } ADV  R/S
LBL GTO  2 INV SUM 00  .4  SUM 01  RCL 00  X .13 {  = SUM 01  16 27  OP 04  RCL 01}
{  OP 06 }   ADV  R/S  LBL PRT 0 1/X  0  R/S
```

Air mail.- The Post Office defines two categories here: letters to Central America, the Caribbian Islands and from Guam to the Philipines pay $ 0.25 for the first half ounce through 2 ounces. Then $ 0.21 for each additional half ounce. To all other countries, that is all the countries in Europe, to Africa, Australia, etc. the rate is $ 0.31 for each half ounce up to 2 ounces, with $ 0.26 for each additional half ounce above two ounces.

As you may observe, there is no maximum limit, so we can dispense with this test. For the rest, the sequence is going to be very similar to routine E above, with one exception: instead of counting in two-ounce steps we count in half-ounce steps. So, instead of DIVIDING by two,we MULTIPLY by two on line 2 of routine E. We also insert the proper numbers (.25, .21, .31 and .26) in the correct places. This way we would end up with a separate routine for each AIR-1 and AIR-2. But is also possible to set a flag when pressing one key and not when pressing the other. That way we can avoid a tremendous amount of duplication, as both routines are going to be identical, save for the numbers .25 and .21 or .31 and .26.

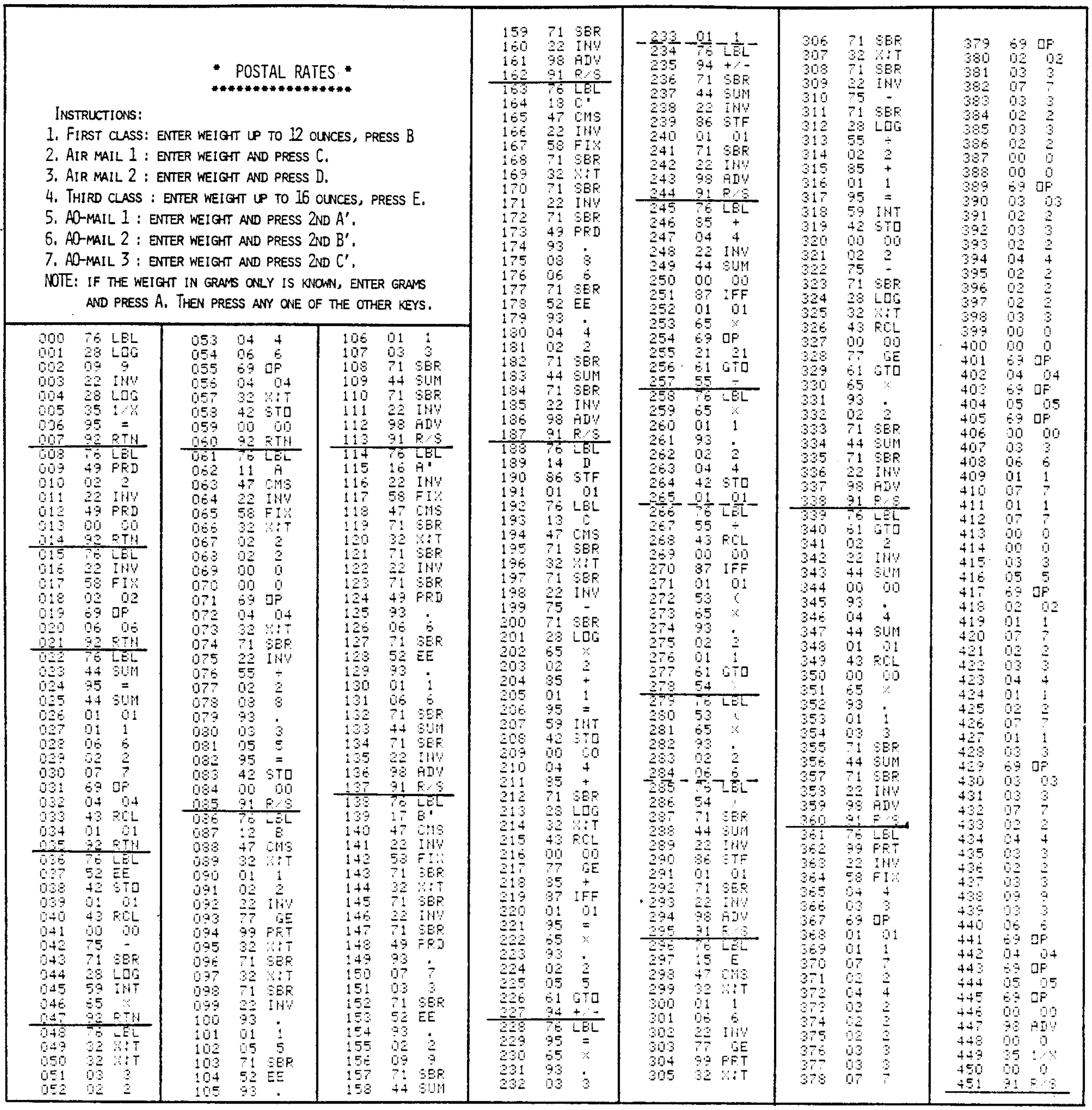
I am not going to list the routine here, as it would become unduly long without resorting to the use of subroutines. You know by now enough about the technique to be able to follow routines C and D in the finished program. LBL D sets flag 1, while LBL C doesn't. Then each time flag 1 is checked and the needed branch is made to insert the correct value. At the end of each routine and branch, flag 1 is reset.

All subroutines are conveniently grouped at the top of the program. This is not done out of an inherent sense of neatness, usually ascribed to programmers. The reason is speed: each time a subroutine is called, the program counter resets to location 000 and starts its search from there for the required LBL. Thus, by placing all the subroutine LBLs at the top we make this search short and speedy. If we later decide to take all the SBR labels out and replace them with direct addresses, this placing the labels at the top loses its advantage. Direct addresses do what their name implies: they reset the program counter DIRECTLY to the address indicated. That is the reason why they are so much speedier than labels.

AO-MAIL.- This is a special third class air mail service. It consists of three different categories, rates depending on the distance mail is carried. The least expensive category is mail going to Central America: $ 0.60 for the first two ounces or a fraction thereof and $ 0.29 for each additional ounce. The next category comprises most of the countries in Europe and the charges are $ 0.73 and $ 0.29 respectively. Then, the last category is the most expensive: to Autralia, Asia, the Middle East, Africa, to the Sowjet Union, etc. with charges of $ 0.86 and $ 0.42.

It should be easy by now to device your own routines for these three categories. I wrote separate routines for each one and finding them almost identical I thought I could save lots of steps by setting two different flags, one for each of two categories and none for the third one. But I found out that the overhead of checking two flags is so enormous that it takes a total of 95 steps, even using all the subroutines available. Three separate routines on the other hand need, by coincidence, exactly 95 steps and each one executes much faster, without having to check all those flags.

Maurice E.T. Swinnen



* POSTAL RATES *

****

INSTRUCTIONS:

1. FIRST CLASS: ENTER WEIGHT UP TO 12 OUNCES, PRESS B
2. AIR MAIL 1 : ENTER WEIGHT AND PRESS C.
3. AIR MAIL 2 : ENTER WEIGHT AND PRESS D.
4. THIRD CLASS : ENTER WEIGHT UP TO 16 OUNCES, PRESS E.
5. AO-MAIL 1 : ENTER WEIGHT AND PRESS 2ND A'.
6. AO-MAIL 2 : ENTER WEIGHT AND PRESS 2ND B'.
7. AO-MAIL 3 : ENTER WEIGHT AND PRESS 2ND C'.

NOTE: IF THE WEIGHT IN GRAMS ONLY IS KNOWN, ENTER GRAMS AND PRESS A. THEN PRESS ANY ONE OF THE OTHER KEYS.

```
000  76 LBL
001  28 LOG
002  09  9
003  22 INV
004  28 LOG
005  35 1/X
006  95  =
007  92 RTN
008  76 LBL
009  49 PRD
010  02  2
011  22 INV
012  49 PRD
013  00  00
014  92 RTN
015  76 LBL
016  22 INV
017  58 FIX
018  02  02
019  69 OP
020  06  06
021  92 RTN
022  76 LBL
023  44 SUM
024  95  =
025  44 SUM
026  01  01
027  01  1
028  06  6
029  02  2
030  07  7
031  69 OP
032  04  04
033  43 RCL
034  01  01
035  92 RTN
036  76 LBL
037  52 EE
038  42 STO
039  01  01
040  43 RCL
041  00  00
042  75  -
043  71 SBR
044  28 LOG
045  59 INT
046  65  ×
047  92 RTN
048  76 LBL
049  32 X:T
050  32 X:T
051  03  3
052  02  2
053  04  4
054  06  6
055  69 OP
056  04  04
057  32 X:T
058  42 STO
059  00  00
060  92 RTN
061  76 LBL
062  11  A
063  47 CMS
064  22 INV
065  58 FIX
066  32 X:T
067  02  2
068  02  2
069  00  0
070  00  0
071  69 OP
072  04  04
073  32 X:T
074  71 SBR
075  22 INV
076  55  ÷
077  02  2
078  08  8
079  93  .
080  03  3
081  05  5
082  95  =
083  42 STO
084  00  00
085  91 R/S
086  76 LBL
087  12  B
088  47 CMS
089  32 X:T
090  01  1
091  02  2
092  22 INV
093  77  GE
094  99 PRT
095  32 X:T
096  71 SBR
097  32 X:T
098  71 SBR
099  22 INV
100  93  .
101  01  1
102  05  5
103  71 SBR
104  52 EE
105  93  .
106  01  1
107  03  3
108  71 SBR
109  44 SUM
110  71 SBR
111  22 INV
112  98 ADV
113  91 R/S
114  76 LBL
115  16 A'
116  22 INV
117  58 FIX
118  47 CMS
119  71 SBR
120  32 X:T
121  71 SBR
122  22 INV
123  71 SBR
124  49 PRD
125  93  .
126  06  6
127  71 SBR
128  52 EE
129  93  .
130  01  1
131  06  6
132  71 SBR
133  44 SUM
134  71 SBR
135  22 INV
136  98 ADV
137  91 R/S
138  76 LBL
139  17 B'
140  47 CMS
141  22 INV
142  58 FIX
143  71 SBR
144  32 X:T
145  71 SBR
146  22 INV
147  71 SBR
148  49 PRD
149  93  .
150  07  7
151  03  3
152  71 SBR
153  52 EE
154  93  .
155  02  2
156  09  9
157  71 SBR
158  44 SUM
159  71 SBR
160  22 INV
161  98 ADV
162  91 R/S
163  76 LBL
164  18 C'
165  47 CMS
166  22 INV
167  58 FIX
168  71 SBR
169  32 X:T
170  71 SBR
171  22 INV
172  71 SBR
173  49 PRD
174  93  .
175  08  8
176  06  6
177  71 SBR
178  52 EE
179  93  .
180  04  4
181  02  2
182  71 SBR
183  44 SUM
184  71 SBR
185  22 INV
186  98 ADV
187  91 R/S
188  76 LBL
189  14  D
190  86 STF
191  01  01
192  76 LBL
193  13  C
194  47 CMS
195  71 SBR
196  32 X:T
197  71 SBR
198  22 INV
199  75  -
200  71 SBR
201  28 LOG
202  65  ×
203  02  2
204  85  +
205  01  1
206  95  =
207  59 INT
208  42 STO
209  00  00
210  04  4
211  85  +
212  71 SBR
213  28 LOG
214  32 X:T
215  43 RCL
216  00  00
217  77  GE
218  85  +
219  87 IFF
220  01  01
221  95  =
222  65  ×
223  93  .
224  02  2
225  05  5
226  61 GTO
227  94 +/-
228  76 LBL
229  95  =
230  65  ×
231  93  .
232  03  3
233  01  1
234  76 LBL
235  94 +/-
236  71 SBR
237  44 SUM
238  22 INV
239  86 STF
240  01  01
241  71 SBR
242  22 INV
243  98 ADV
244  91 R/S
245  76 LBL
246  85  +
247  04  4
248  22 INV
249  44 SUM
250  00  00
251  87 IFF
252  01  01
253  65  ×
254  69 OP
255  21  21
256  61 GTO
257  55  ÷
258  76 LBL
259  65  ×
260  01  1
261  93  .
262  02  2
263  04  4
264  42 STO
265  01  01
266  76 LBL
267  55  ÷
268  43 RCL
269  00  00
270  87 IFF
271  01  01
272  53  (
273  65  ×
274  93  .
275  02  2
276  01  1
277  61 GTO
278  54  )
279  76 LBL
280  53  (
281  65  ×
282  93  .
283  02  2
284  06  6
285  76 LBL
286  54  )
287  71 SBR
288  44 SUM
289  22 INV
290  86 STF
291  01  01
292  71 SBR
293  22 INV
294  98 ADV
295  91 R/S
296  76 LBL
297  15  E
298  47 CMS
299  32 X:T
300  01  1
301  06  6
302  22 INV
303  77  GE
304  99 PRT
305  32 X:T
306  71 SBR
307  32 X:T
308  71 SBR
309  22 INV
310  75  -
311  71 SBR
312  28 LOG
313  55  ÷
314  02  2
315  85  +
316  01  1
317  95  =
318  59 INT
319  42 STO
320  00  00
321  02  2
322  75  -
323  71 SBR
324  28 LOG
325  32 X:T
326  43 RCL
327  00  00
328  77  GE
329  61 GTO
330  65  ×
331  93  .
332  02  2
333  71 SBR
334  44 SUM
335  71 SBR
336  22 INV
337  98 ADV
338  91 R/S
339  76 LBL
340  61 GTO
341  02  2
342  22 INV
343  44 SUM
344  00  00
345  93  .
346  04  4
347  44 SUM
348  01  01
349  43 RCL
350  00  00
351  65  ×
352  93  .
353  01  1
354  03  3
355  71 SBR
356  44 SUM
357  71 SBR
358  22 INV
359  98 ADV
360  91 R/S
361  76 LBL
362  99 PRT
363  22 INV
364  58 FIX
365  04  4
366  03  3
367  69 OP
368  01  01
369  01  1
370  07  7
371  02  2
372  04  4
373  02  2
374  02  2
375  02  2
376  03  3
377  03  3
378  07  7
379  69 OP
380  02  02
381  03  3
382  07  7
383  03  3
384  02  2
385  03  3
386  02  2
387  00  0
388  00  0
389  69 OP
390  03  03
391  02  2
392  03  3
393  02  2
394  04  4
395  02  2
396  02  2
397  02  2
398  03  3
399  00  0
400  00  0
401  69 OP
402  04  04
403  69 OP
404  05  05
405  69 OP
406  00  00
407  03  3
408  06  6
409  01  1
410  07  7
411  01  1
412  07  7
413  00  0
414  00  0
415  03  3
416  05  5
417  69 OP
418  02  02
419  01  1
420  07  7
421  02  2
422  03  3
423  04  4
424  01  1
425  02  2
426  07  7
427  01  1
428  03  3
429  69 OP
430  03  03
431  03  3
432  07  7
433  02  2
434  04  4
435  03  3
436  02  2
437  03  3
438  09  9
439  03  3
440  06  6
441  69 OP
442  04  04
443  69 OP
444  05  05
445  69 OP
446  00  00
447  98 ADV
448  00  0
449  35 1/X
450  00  0
451  91 R/S
```

REMAINDER ROUTINE.- With respect to the remainder routine on v5n4/5p13, Bob Patton, Arlington, TX, writes that he used a similar one: REMAINDER (N/D) = N - INT(N/D)·D

One of the requirements for the program routine was, that no register, except t, be used. The routine he presents here seems bulkier than he would like, but it is the shortest one he could come up with. I think it is a rather elegant solution, though. Call the routine with N in the t-reg and D in the x-reg (display):

LBL E' (X:T - (CE DIV X:T (CP + X:T)) INT X X:T) RTN

QUADRATIC SOLUTION.- Re v5n4/5p13, Richard Vanderburgh says that he never saw such a solution in the PPC Journal, but he recalls being introduced to such a mechanization for the first programmable RPN, the HP-65, by John Herro. Richard wrote a similar one for the SR-56 in v2n2p2 of 52-Notes.

SORTING WITH DESCRIPTORS.- From Programbiten (Sweden) 79-3/4-p38/39 comes this unique sorting program written by Henrik Tjernberg. It permits the entry of up to 46 items consisting of a number with max 9 digits and a descriptor with up to max 4 characters, i.e. max 8 code digits. Then it will sort and print out the entered items with their respective descriptors in ascending order. You may pre-determine the number of decimals in the entries, so long as the sum of the digits in the integer part plus the fractional part is not larger than 9.

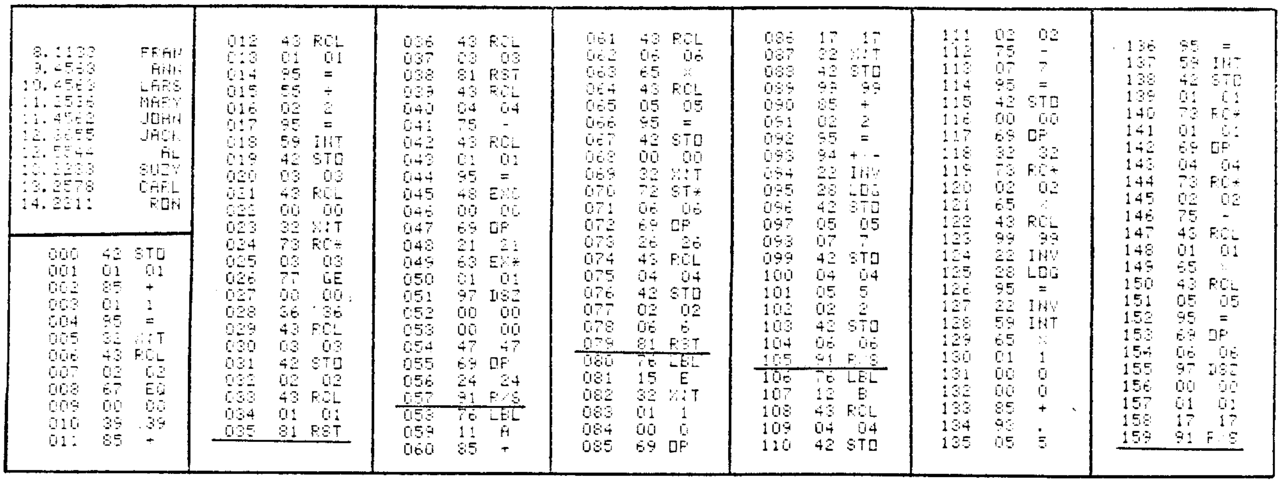
To record the program on one side of a mag card, key in the program in 6 OP 17 partition and record the one side in that partition. During execution of the program that partition is automatically changed to 10 OP 17.

Instructions to run the program:

1. *Enter the number of digits in the fractional part of the numbers you intend to enter and press E. In the original program the entries were "times" in sport events. The descriptors were the names of the participants.*
2. *Enter the alpha code, max 8 digits, of the descriptor and press X:T.*
3. *Enter the corresponding number and press A.*
   REPEAT X:T AND A AS MANY TIMES AS NEEDED, MAX 46 TIMES.
4. *To sort in ascending order and print out, press B.*
   See example of print out at the beginning of the program listing.

For those interested how it is done, take a peek at the registers (CLR INV LIST) and see, for example 1.572654. Suppose you asked for 4 decimals. The 1.5726 is your entry, the 54 is the number of the register it is stored in. According to the author, sorting is done by means of the "intervallhalveringsmetoden." No translation needed here.

Note: It is possible to obtain partial results. For example, you may press B after only a few entries and obtain a preliminary sorting, after which you enter more items and obtain a final listing. I suspect the special module used at Lake Placid last winter must have contained such a routine. I have not yet been able to lay my hands on one and I am not even sure that I will be able to download it. It might be a protected one.



```
 8.1133      FRAN
 9.4563       ANN
10.4562      LARS
11.1536      MARY
11.4562      JOHN
12.3655      JACK
12.5544        AL
13.1233      SUZY
13.2578      CARL
14.2211       RON

000  42  STO     012  43  RCL     036  43  RCL     061  43  RCL     086  17   17     111  02   02     136  95   =
001  01   01     013  01   01     037  03   03     062  06   06     087  32  X:T     112  75   -      137  59  INT
002  85   +      014  95   =      038  81  RST     063  65   ×      088  42  STO     113  07   7      138  42  STO
003  01   1      015  55   ÷      039  43  RCL     064  43  RCL     089  99   99     114  95   =      139  01   01
004  95   =      016  02   2      040  04   04     065  05   05     090  85   +      115  42  STO     140  73  RC*
005  32  X:T     017  95   =      041  75   -      066  95   =      091  02   2      116  00   00     141  01   01
006  43  RCL     018  59  INT     042  43  RCL     067  42  STO     092  95   =      117  69  OP      142  69  OP
007  02   02     019  42  STO     043  01   01     068  00   00     093  94  +/-     118  32   32     143  04   04
008  67  EQ      020  03   03     044  95   =      069  32  X:T     094  22  INV     119  73  RC*     144  73  RC*
009  00   00     021  43  RCL     045  48  EXC     070  72  ST*     095  28  LOG     120  02   02     145  02   02
010  39   39     022  00   00     046  00   00     071  06   06     096  42  STO     121  65   ×      146  75   -
011  85   +      023  32  X:T     047  69  OP      072  69  OP      097  05   05     122  43  RCL     147  43  RCL
                 024  73  RC*     048  21   21     073  26   26     098  07   7      123  99   99     148  01   01
                 025  03   03     049  63  EX*     074  43  RCL     099  42  STO     124  22  INV     149  65   ×
                 026  77  GE      050  01   01     075  04   04     100  04   04     125  28  LOG     150  43  RCL
                 027  00   00     051  97  DSZ     076  42  STO     101  05   5      126  95   =      151  05   05
                 028  36   36     052  00   00     077  02   02     102  02   2      127  22  INV     152  95   =
                 029  43  RCL     053  00   00     078  06   6      103  42  STO     128  59  INT     153  69  OP
                 030  03   03     054  47   47     079  81  RST     104  06   06     129  65   ×      154  06   06
                 031  42  STO     055  69  OP      080  76  LBL     105  91  R/S     130  01   1      155  97  DSZ
                 032  02   02     056  24   24     081  15   E      106  76  LBL     131  00   0      156  00   00
                 033  43  RCL     057  91  R/S     082  32  X:T     107  12   B      132  00   0      157  01   01
                 034  01   01     058  76  LBL     083  01   1      108  43  RCL     133  85   +      158  17   17
                 035  81  RST     059  11   A      084  00   0      109  04   04     134  93   .      159  91  R/S
                                  060  85   +      085  69  OP      110  42  STO     135  05   5
```

TI-59 KEY CODES.- Re Jared Weinberger's routine in v5n4/5p8 and Karl-Josef Meusch's one in v5n6p12: There seems to be always one more way to skin that proverbial cat. Bill Skillman proves it by this routine:

```
LBL A  100 STO 01  9200.760869  STO 00  LBL A'  0 OP 17  SBR  955  6  OP 17  OP 20
DSZ 1  A'  R/S
```
(last step = 036)

Press A to have the printer produce all possible key codes and their respective mnemonics at step 958.

See you next time, Maurice