* T I P P C N O T E S *

****

v5N9-10 , 1980

NEWSLETTER OF THE TI PROGRAMMABLE CALCULATOR CLUB.

9213 Lanham Severn Road, Lanham, Maryland, 20801, USA

This extended issue v5n9-10 will be the last one in 1980. As a consequence may I ask you to renew your subscription to the TI PPC NOTES. Or, in other words, we don't have enough money in the kitty to continue into next year, unless you reach into your pocket, purse or what have you, and send us your 1981 contribution.

For members living within the US,(including Puerto Rico, Virgin Islands, Guam, etc.) we have not contemplated any change. $ 20.00 will still do it. If, however, during the course of the year things get more expensive, we reserve the right to either decrease the number of issues or the number of pages per issue. We are a non-profit organization and will try to divide the available money evenly over the year.

The $ 20.00 is sufficient if we continue bulk rate mailing. If you have trouble with this kind of mail service, and a few people have complained, send us $ 5.00 extra and we will carry you on an extra first class list. But please, don't ask me to change you over from bulk rate to first class in the middle of the year. As you'll remember, I HATE ADMINISTRATION and try to reduce it to the absolute minimum. Thank you.

Members living in Canada and Mexico have received their newsletter so far at first class rates, as the post office does not permit bulk rate to these countries. As the numbers involved were not not very large, the club didn't mind subsidizing this mailing. But that is no longer possible since the list, especially of Canadian members, grows every day. So, Canada and Mexico, please add $ 5.00 extra to your subscription.

Members living outside North America: The special A.O. Mail Printed Matter rate we have been allowed to use is much less expensive than outright Air Mail, of course. It differs from continent to continent, in three categories, between 60 and 86 cents per issue. But, during the summer an International Post Office conference was held in Rio de Janeiro, in which it was proposed to increase all Air Mail tariffs. As an example, an ordinary ½ ounce letter would be increased from 31 to 40 cents. We will not try to anticipate the exact increase it will provoke in each of the three A.O. Mail categories. That would be an impossible task. But increase they will, and soon. The post office has already received large amounts of larger-denominations air-mail stamps. To make it simple, may I propose to those members living outside North America to send $ 10.00 extra. If, however, you don't require air mail delivery, $ 5.00 extra, or the equivalent of domestic first class, will do for surface mail.

Now about the <u>way</u> of sending your contributions: inside the US the great majority sends a bank check. No trouble at all. My local bank takes care of it, without any extra cost to the club.

But members outside the US cause some trouble. If your check is in US dollars <u>and</u> drawn on a US bank, no trouble whatsoever. Most of the members in Australia and New-Zealand work that way. If your check is in foreign currency, forget it. I still have checks bouncing back and forth between my

TABLE OF CONTENTS.

bank and the Riggs bank, the only one in the area which exchanges those kinds of checks. They work not for peanuts, however. Each time there is a substantial collection fee involved, plus long waiting.

The only way besides checks in US currency on US banks is to buy at your local post office a Postal Money Order. Or, as most people in Europe have a Post Giro account, simply send a Post Giro order. Both forms works flawlessly. As a last resort, just put the equivalent of the contribution in your own currency in paper money in an air mail letter. It is much, much easier to exchange money than checks. Thank you.

To put your mind at ease, we have reached the break even point somewhere around October of this year. That means, money is now coming in faster than it is leaving. At one point I feared for my personal savings, but things are looking rosier now.

That brings us to a new subject: we might start thinking about projects that require seed money from the club. We would like to revive the old idea that has been kicking around in clubs such as Heinrich Schnepf's Display, Phillip Rowley's British TI club and others: OUR OWN UTILITY MODULE.

The Swedish club has started the ball rolling by means of an announcement in in the latest Programbiten newsletter. That, after I received a telephone call from Björn Gustavsson, who, by the way, must be quite a polyglot. I received two letters from him in flawless Dutch. (sometimes also called Flemish or, more modern, Netherlandish)

We plan to put all kinds of utility routines on our module: Robert Snow's print code converter, or a derivation thereof, Karl Gailer's abbreviated print code routines, Richard Snow's Alpha sorter, other sorting routines, a good random number generator, in short, anything not covered by the M/U module.

TI is willing to produce the modules according to the following price schedule:

Quantity	Total price in dollars US
250	12,000
500	17,000
1,000	25,000
2,000	42,000
5,000	92,000

I suspect that we will end up with 250 or 500 quantity. TI wants $ 2,000 in advance to test the programs on an emulator. Once that is done and the programs savely on cassette tapes, TI wants half the total price, less the $ 2,000. Then TI will start its production. The remaining half will have to be paid upon delivery of the modules. Sample modules will arrive after six weeks, production modules after three months.

If all this attracts you, please tell me ON A SEPARATE SHEET OF PAPER if you also talk of other subjects in your letter:
What you would like to see on the module, how much money you would like to spend for it (taking in account also that we are going to print extensive documentation in the form of a booklet) in short, anything that comes to your mind and that you think might help the effort.

Permit me a last exhortation about the subscriptions: Don't wait till the end of Januari to send it. Be nice and make my life a little easier by sending it now. I will be very grateful for it.

Have a nice or merry Christmas, or Holidays, as the case may be, and a happy New Year. May in the year to come appear on the market such a fantastic TI programmable calculator that we will win all the calendar printing contests and that I may write articles about it for many years hence. May the machine have Basic, Cobol and Pascal, a floppy disk accessory, a line printer and a telephone modem so that the Snow brothers can send me their latest creations by phone. But, as they say in Chile: "Sonar no cuesta nada", which one of our lab technicians, with a flair for language, translates very aptly as "dreamin' don't cost nothin',boy."

Maurice E.T. Swinnen

CALENDAR PRINTING CONTEST.- The saga of the fast calendar continues. On October 8 I received from Palmer Hanson his revision A: gain 1 sec per year. This program eliminates the long series of X=T tests in the original Panos Galidas program. Palmer thought we were running into the "Existence theorem" again: Why work my tail off, if I am that far ahead in the race. A similar malaise had set in half-way through the contest in 52-Notes, some two years ago. But then, Palmer comes up with revision B, a gain of 6 sec per year. So much for the Existence Theorem. That was on Nov 8. Palmer employed successfully the idea proposed by Richard Snow in v5n8p1 of separating OP 05 ADV sequences as a means of speeding up fast mode.
Palmer also notes that speed could be gained, about 3 %, by running the calculator in scientific mode. It worked well on a Speedy Factor Finder program, but unexplicably, it failed in the Fast Calendar program. Palmer still investigates further.

Then on Nov 28 Roger Hill struck again. I recieved the latest PPC JOURNAL from the HP club and in it Roger Hill explains that his original program of 2 min 19 sec per year looked like an oxcart on the freeway in comparison to Palmer's 1 min 32 sec program. However, in this round of the " ~~friendly~~ competition" as Roger puts it, he managed to write a program that zips along at 1 sec above the theoretical maximum speed, 1 min 14 sec, or 12 seconds faster than Palmer's 1 min 26 sec. Our sincere congratulations to Roger. Once we have recovered sufficiently from the shock we will send you our answer, in the form of a lightning-fast program.

SOLUTION OF THREE EQUATIONS IN THREE UNKNOWNS.- In v5n8p5 I presented a program by COMPLEX. that name written by Bill Beebe. If you observed closely, everything was there EXCEPT BILL'S PROGRAM. I wanted to complement Bill's "for calculator only" program by one that also had some print out. So, I reworked the program and then proceeded to publish only the enhanced one. My sincere apologies to Bill. Here is his original one. By the way, just to spite me, he sent me an improved version, two steps shorter than his first one.
Instructions: Initialize, press E. This clears all memories.
Coefficients entered in POLAR form, in order as indicated in v5n8p5. Once entry 12 has been entered, the program automatically computes the value of the determinant, delta, in complex form. It will place the real part in the display, while the imaginary part may be obtained by pressing x:t. Avoid division by zero.
Please re-read the article on v5n8p5 with respect to error recovery.

```
000  76 LBL
001  18 C'
002  35  +
003  01  1
004  08  8
005  00  0
006  76 LBL
007  19 D'
008  95  =
009  37 P/R
010  44 SUM
011  02  02
012  32 X:T
013  44 SUM
014  01  01
015  92 RTN
016  76 LBL
017  11  A
018  72 ST+
019  00  00
020  69 OP
021  20  20
022  32 X:T
023  72 ST+
024  00  00
025  69 OP
026  20  20
027  97 DSZ
028  01  01
029  00  00
030  15  15
031  17 B'
032  43 RCL
033  02  02
034  42 STO
035  04  04
036  32 X:T
037  43 RCL
038  01  01
039  42 STO
040  03  03
041  92 RTN
042  76 LBL
043  17 B'
044  00  0
045  42 STO
046  01  01
047  42 STO
048  02  02
049  43 RCL
050  06  06
051  65  X
052  43 RCL
053  14  14
054  65  X
055  43 RCL
056  22  22
057  95  =
058  32 X:T
059  43 RCL
060  05  05
061  85  +
062  43 RCL
063  13  13
064  85  +
065  43 RCL
066  21  21
067  19 D'
068  43 RCL
069  12  12
070  65  X
071  43 RCL
072  20  20
073  65  X
074  43 RCL
075  10  10
076  95  =
077  32 X:T
078  43 RCL
079  11  11
080  85  +
081  43 RCL
082  19  19
083  85  +
084  43 RCL
085  09  09
086  19 D'
087  43 RCL
088  18  18
089  65  X
090  43 RCL
091  08  08
092  65  X
093  43 RCL
094  16  16
095  95  =
096  32 X:T
097  43 RCL
098  17  17
099  85  +
100  43 RCL
101  07  07
102  85  +
103  43 RCL
104  15  15
105  19 D'
106  43 RCL
107  10  10
108  65  X
109  43 RCL
110  14  14
111  65  X
112  43 RCL
113  18  18
114  95  =
115  32 X:T
116  43 RCL
117  09  09
118  85  +
119  43 RCL
120  13  13
121  85  +
122  43 RCL
123  17  17
124  18 C'
125  43 RCL
126  16  16
127  65  X
128  43 RCL
129  20  20
130  65  X
131  43 RCL
132  06  06
133  95  =
134  32 X:T
135  43 RCL
136  15  15
137  85  +
138  43 RCL
139  19  19
140  85  +
141  43 RCL
142  05  05
143  18 C'
144  43 RCL
145  22  22
146  65  X
147  43 RCL
148  08  08
149  65  X
150  43 RCL
151  12  12
152  95  =
153  32 X:T
154  43 RCL
155  21  21
156  85  +
157  43 RCL
158  07  07
159  85  +
160  43 RCL
161  11  11
162  18 C'
163  92 RTN
164  76 LBL
165  14  D
166  06  6
167  85  +
168  76 LBL
169  13  C
170  06  6
171  85  +
172  76 LBL
173  12  B
174  05  5
175  95  =
176  42 STO
177  29  29
178  71 SBR
179  01  01
180  97  97
181  17 B'
182  36 PGM
183  04  04
184  18 C'
185  48 EXC
186  29  29
187  71 SBR
188  01  01
189  97  97
190  01  1
191  94 +/-
192  49 PRD
193  04  04
194  43 RCL
195  29  29
196  92 RTN
197  42 STO
198  00  00
199  02  2
200  03  3
201  42 STO
202  01  01
203  06  6
204  42 STO
205  02  02
206  73 RC*
207  01  01
208  63 EX*
209  00  00
210  73 RC*
211  01  01
212  69 OP
213  00  00
214  69 OP
215  31  31
216  97 DSZ
217  02  02
218  02  02
219  06  06
220  92 RTN
221  76 LBL
222  15  E
223  47 CMS
224  01  1
225  76 LBL
226  16  A'
227  75  -
228  01  1
229  00  0
230  85  +
231  42 STO
232  01  01
233  85  +
234  02  2
235  09  9
236  95  =
237  42 STO
238  00  00
239  91 R/S
```

RADAR CALCULATIONS WITH THE TI-59 PROGRAMMABLE CALCULATOR.- Bill Skillman sent me a tentative outline of a book by that title he is writing. I have seen Bill's programs before and I know of his reputation as a technical writer and teacher. So, the book promises to be a winner. Bill would like to invite other programmers to get into the act. The authors will be given due credit, of course. So, here is your chance to be remembered for posterity. To make it easy on Bill (I know what I am talking about, believe me) send your submissions to Bill preferrably on a PPX summission form and include recorded mag cards. Bill's address is : 605 Forest View Road, Linthicum Heights, MD, 21090. Bill is an EE and is employed at Westinghouse in nearby Baltimore.

MICROWAVE CIRCUIT DESIGN USING PROGRAMMABLE CALCULATORS.- J.L. Allen and M.W. Medley, Jr. 297 pp; $ 40.00, Artech House, Inc. 610 Washington St, Dedham MA 02026, USA, Contains well-written design and analysis programs for the TI-59, HP-67/97 and HP-41C. Solves problems encountered in the design of linear amplifiers, matching networks, general two-ports and filter analysis. Equations and flow charts are absent, though. Examples of program execution are given.

A pre-programmed HP-41C module containing the complete Microwave Pac (all the programs in the above book) are available from Compact Engineering Inc, 1070 East Meadow Circle, Palo Alto, CA, 94304, USA, tel (415) 858-1200. I have not heard of any attempts yet to supply a TI-59 module, but if enough people ask, the company might be persuaded.

AN INEXPENSIVE PROGRAMMABLE CALCULATOR SYSTEM FOR EVERYDAY USE IN A RADIATION THERAPY DEPARTMENT, Philip H. Heintz, Ph.D., Radiation Oncology Center of Sutter Community Hospitals, Sacramento, CA 95819, USA. *Int. J. Radiation Oncology Biol.Phys. Vol.5, pp.2117-2119, Pergamon Press Ltd., 1979.*
Programs are for the TI-59 and the Tektronix model 31 calculators.

Related to the above, a new PPX submission by Barry Tepperman of Toronto, Canada: TISSUE-AIR COEFFICIENTS FOR RADIATION THERAPY. As is ususal and normal for Barry, an excellently written program that produces rapid derivation of the correct coefficients for a treatment machine. Derives the appropriate regression coefficients between radiation dose, field size and tissue depth for use in planning radiation therapy by means of Sterling's analytical model for depth-dose distribution.

TI-59 INVERTS LAPLACE TRANSFORMS FOR TIME-DOMAIN ANALYSIS, Kin-chu Woo, Texas A & M University, Dept. of Electrical Engineering, College Station, TX, USA. Electronics, October 9, 1980, pp 178-179.
This program works also on the TI-58.

3-D NAVIGATION.- Dave Leising of Grand Rapids, MI, sends me a brochure on this subject. Jet Electronics and Technology, Inc. 5353 52nd Street, Grand Rapids, MI, 49508, USA, produces a three-dimensional navigation computer called the DAC-7000. It is now possible to use any TI-59 to program the DAC-7000. This feature gives unlimited route storage capabilities. The advantage of all this is, that a pilot saves time and especially fuel when given efficient, straight-line R-NAV routing along with profile climbs and descents in V-NAV.

TIME BOMBS.- Although the program on v5n7p12 was only intended as a demonstration of Richard Snow's application of "branching from the keyboard", some members asked me to publish the user instructions for that program. Here they are:
Live bombs will explode if:1) you cut the red wire, 2) you cut the unknown "deadly" wire, 3) you cut the green wire, THEN cut one of the two primary wires, or 4) you run out of time. (the total of seconds left is displayed)
Live bombs can be defused if you cut the unknown "defuse" wire or you cut one of the two primary wires THEN cut the green wire.
Press A, then to cut a wire press R/S C (for cut.) All output is printed.

HÖHENLINIENDIAGRAM.- That is the German word for "Contour graph." Not possible on the TI-59/PC100 ? I have tried it often, but the task seemed so formidable that I soon gave up. Not Harald M. Oto in Bad Rothenfelde, West Germany. In one of the former issues I promised you to bring one of his "impossible" programs. Here is a program that will plot up to 9 levels in a contour graph, of a function you enter in program memory.

It is true, I have seen a similar program in Peter Poloczek's library by Robert Rudolph from Hamburg. But that program is slower than molasses, and I soon gave up considering it for any practical purpose. There is only so much time available during one working day. Harald's routine, although not in line to win a speed contest, is more than 25 % faster than Robert's, requiring roughly four hours to finish an 8½ by 11 inches contour graph.

It does not require the use of any module and has only 290 steps. You enter your function by pressing E. The calculator goes automatically into LRN mode. There, at step 291, start writing your function. Two examples are given below on the right. On the left hand below is shown the entire sequence of entering the parameters for the first example and the resulting print out. As with all plotting programs that require more than 20 characters horizontally, you will have to paste together the various strips of paper, so as to form the entire diagram. On the next page are printed the two graphs which result from each of the examples given below. On them the axis are drawn in by hand, of course.

After you have entered your function in E, to start the ball rolling press A. Now enter in order, prompted by the calculator:
X1, the start value of X1, which runs in the same direction as the paper strip.
X2, the start value of X2, which runs 90° with the paper strip.
NI, number of "levels" in the contour graph. (0 through 8) NI stands for "Niveau."
BA, the number of paper strips, any value possible. BA=Bahnen or strips.
e, a precision factor. (try 1.5 as a start)
ΔX1, the increment of X1.
ΔX2, the encrement of X2.
N, the number of print lines per paper strip.
NI0 through NI8, value of each level. For each level you have to enter a specific value. There is no default value available.

Once you have done all this, the program takes off and Harald advices :"Sich in Geduld fassen, eine Tasse Kaffee trinken, sich der Frau mal wieder widmen..." which means so much as " Practice patience, drink a cup of that dark brew, dedicate some time again to your wife..."

I promise you, you'll have time for all that and more.

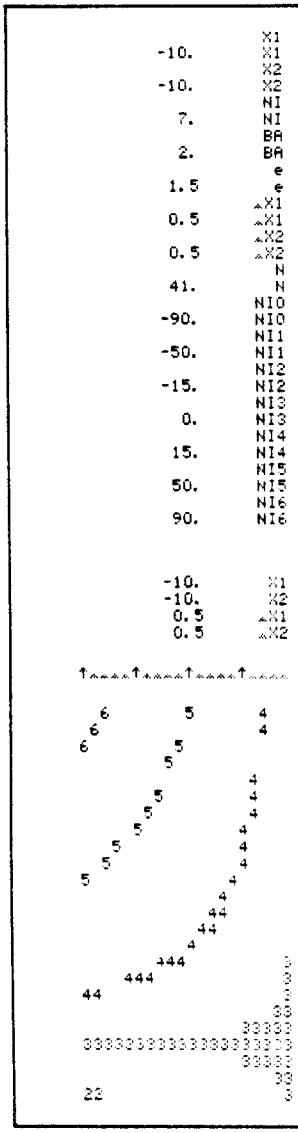
THE PROGRAM ITSELF DOESN'T SEEM TO FIT ANY MORE ON THIS PAGE. THEREFORE, READ "BELOW" ABOVE AS "NEXT PAGE" AND "NEXT PAGE" AS " OVERNEXT PAGE." Got it?

SELECTIVE INV LIST.- Karl-Joseph Meusch wrote this practical routine. It will list a number of data registers to be chosen by the user. It takes about 2.8 sec per register. It mimics exactly INV LIST.

Instructions:
Enter the number of the highest register you want to list and press A.
Enter the number of the lowest register you want to list and press B.
In case you want only one single register listed, enter 0 A N B.

```
LBL E 10 PRD 59 PRD 59 + 1 + LOG INT X 2 RTN LBL B STO 58
( CE DIV E - INT STO 59 ) ( CE X E + RCL 59 ) OP 04
RCL IND 58 OP 06 1 SUM 58 RCL 58 INV GE 020 ADV 0 RTN
LBL A + 1 = X:T RTN
```

HÖHENLINIENDIAGRAM.- Program listing.




```
000 76 LBL      065 42 STD      130 75 -        195 12 12       260 19 D'
001 11 A        066 06 06       131 43 RCL      196 44 SUM      261 85 +
002 69 OP       067 25 CLR      132 03 03       197 07 07       262 07 7
003 00 00       068 14 D        133 95 =        198 97 DSZ      263 05 5
004 07 7        069 01 1        134 50 I×I      199 01 01       264 04 4
005 42 STD      070 14 D        135 22 INV      200 00 00       265 04 4
006 06 06       071 98 ADV      136 77 GE       201 97 97       266 00 0
007 25 CLR      072 06 6        137 01 01       202 43 RCL      267 02 2
008 16 A'       073 00 0        138 48 48       203 14 14       268 95 =
009 17 B'       074 07 7        139 69 OP       204 65 ×        269 92 RTN
010 01 1        075 05 5        140 34 34       205 43 RCL      270 76 LBL
011 16 A'       076 07 7        141 97 DSZ      206 12 12       271 10 E'
012 17 B'       077 05 5        142 00 00       207 95 =        272 85 +
013 03 3        078 07 7        143 01 01       208 22 INV      273 03 3
014 01 1        079 05 5        144 28 28       209 44 SUM      274 01 1
015 02 2        080 07 7        145 61 GTO      210 07 07       275 02 2
016 04 4        081 05 5        146 01 01       211 02 2        276 04 4
017 17 B'       082 69 OP       147 61 61       212 00 0        277 00 0
018 42 STD      083 01 01       148 43 RCL      213 65 ×        278 01 1
019 01 01       084 69 OP       149 26 26       214 43 RCL      279 95 =
020 85 +        085 02 02       150 22 INV      215 13 13       280 61 GTO
021 01 1        086 69 OP       151 28 LOG      216 95 =        281 17 B'
022 04 4        087 03 03       152 65 ×        217 44 SUM      282 76 LBL
023 95 =        088 69 OP       153 43 RCL      218 08 08       283 14 D
024 42 STD      089 04 04       154 00 00       219 98 ADV      284 19 D'
025 25 25       090 69 OP       155 95 =        220 97 DSZ      285 61 GTO
026 01 1        091 05 05       156 52 EE       221 10 10       286 02 02
027 04 4        092 98 ADV      157 22 INV      222 00 00       287 52 52
028 01 1        093 43 RCL      158 52 EE       223 54 54       288 76 LBL
029 03 3        094 14 14       159 44 SUM      224 91 R/S      289 15 E
030 17 B'       095 42 STD      160 05 05       225 76 LBL      290 31 LRN
031 05 5        096 01 01       161 02 2        226 16 A'
032 04 4        097 01 1        162 22 INV      227 85 +
033 17 B'       098 42 STD      163 44 SUM      228 04 4
034 25 CLR      099 06 06       164 26 26       229 04 4
035 19 D'       100 04 4        165 43 RCL      230 00 0
036 17 B'       101 42 STD      166 13 13       231 02 2
037 01 1        102 27 27       167 44 SUM      232 95 =
038 19 D'       103 25 CLR      168 08 08       233 92 RTN
039 17 B'       104 42 STD      169 97 DSZ      234 76 LBL
040 03 3        105 05 05       170 02 02       235 17 B'
041 01 1        106 05 5        171 01 01       236 69 OP
042 17 B'       107 42 STD      172 12 12       237 04 04
043 43 RCL      108 02 02       173 43 RCL      238 69 OP
044 09 09       109 08 8        174 05 05       239 05 05
045 75 -        110 42 STD      175 84 OP+      240 25 CLR
046 43 RCL      111 26 26       176 06 06       241 91 R/S
047 01 01       112 71 SBR      177 69 OP       242 42 STD
048 95 =        113 02 02       178 26 26       243 06 06
049 10 E'       114 91 91       179 97 DSZ      244 69 OP
050 97 DSZ      115 42 STD      180 27 27       245 06 06
051 01 01       116 03 03       181 01 01       246 69 OP
052 00 00       117 43 RCL      182 03 03       247 26 26
053 42 42       118 11 11       183 84 OP+      248 92 RTN
054 98 ADV      119 32 X↑T      184 06 06       249 76 LBL
055 92 RTN      120 43 RCL      185 02 2        250 18 C'
056 07 7        121 06 06       186 00 0        251 16 A'
057 42 STD      122 42 STD      187 65 ×        252 69 OP
058 06 06       123 00 00       188 43 RCL      253 04 04
059 25 CLR      124 43 RCL      189 13 13       254 73 RC+
060 18 C'       125 25 25       190 95 =        255 06 06
061 01 1        126 42 STD      191 22 INV      256 61 GTO
062 18 C'       127 04 04       192 44 SUM      257 02 02
063 01 1        128 73 RC*      193 08 08       258 44 44
064 02 2        129 04 04       194 43 RCL      259 76 LBL
```

$f(x_1 ; x_2) = x_1 \cdot x_2$

```
291 43 RCL
292 07 07
293 65 ×
294 43 RCL
295 08 08
296 95 =
297 92 RTN
```

$f(x_1;x_2) = \sqrt{x_1^2 + x_2^2}$

```
291 43 RCL
292 07 07
293 33 X²
294 85 +
295 43 RCL
296 08 08
297 33 X²
298 95 =
299 34 √X
300 92 RTN
```

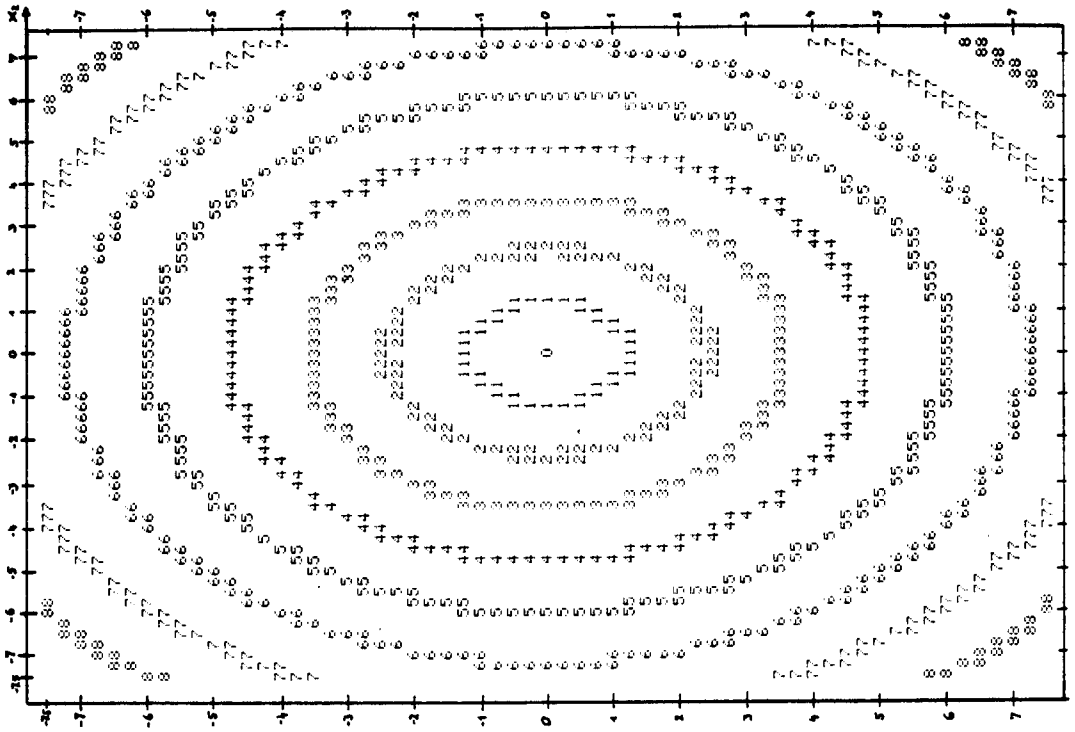
FRACTURED DIGITS.- Dave Leising writes: *I wonder if you have seen this procedure for producing beautiful "kaleidoscopic" displays of fractured digits: From turn-on and in keyboard mode, write:*

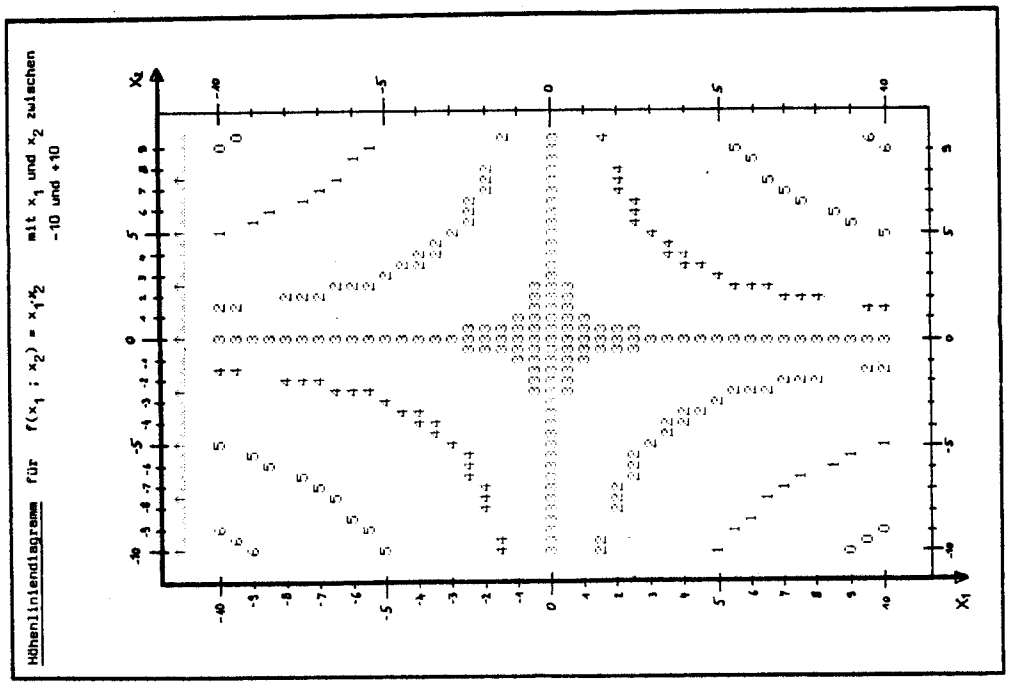
9 OP 17 PGM 12 SBR 444 R/S P-R LRN SST DEL DEL DEL .....

At each delete, a new display of fractured digits appears. I have enclosed a copy of the first 122 patterns. (Too large to publish. Ed.) *They always repeat. This is not a random phenomenon. I suspect that what is being displayed is a coded readout of machine microcode.*

Needless to say that the ML module has to be in place. Maybe somebody gets it to work with another module, maybe with different results?

HOEHENLINIENDIAGRAM.- Sample print-outs.





SPEEDY FACTOR FINDER.- We started the ball rolling on this subject in v5n6p7 with Bill Skillman's program. Suddenly I receive a deluge of similar programs, all claiming to be faster. (some of them are considerably speedier) I once followed with fascination my German friends "slugging it out" in their newsletter Display with the aim of finding out who could write the fastest factor finder program for the HP-67 and the SR-52. Names like K.P. Frank, B. Hoffrichter and V. Lopasic are forever linked in my mind to ingenious approaches and nice programming tricks. In December 1979 I translated all the articles I could find on this subject and distributed them in our local Washington DC club. I still think that it provided the kindle to "light the fire" of the likes of Bill Skillman and Norman Herzberg. Those two more than anybody else have consistently written programs for Speedy Factor Finders (SFF) one faster than the former one. Now, we got some "new" talent working on the problem:
The first program is by Palmer O. Hanson, who adapted Bill's SFF to Fast Mode. It does our two test numbers in really record time:
987654321 (3X3X17X17X379721) in 1 min 40 sec as opposed to Bill's original:2 min 26 s. This one will run also in normal mode at 2 min 53 sec.
103569859 (463X467X479) in 1 min 17 sec , while for bill's original: 1 min 53 sec. This also runs in normal mode at 2 min 25 sec.
To run in fast Mode: Load side 1, RST R/S . Then re-load side 1, CE, enter any integer and press R/S. The calculator/printer will print the integer and all possible factors. It will stop with the highest factor flashing in the display.
Press CE, enter a new integer, etc.

Palmer also sent me a normal mode SFF program, which is not printed here and which is an enhancement of the Louder/Vanderburgh program in v3n11p4 of 52-Notes. It does our first test number in 2 min 14 sec and the second one in 1 min 45 sec. Palmer is now trying to convert that one to fast mode. I will publish it when ready.

The second program here is by Björn Gustavsson from Smedjebacken in Sweden. He wrote a whole new version and converted it also to fast mode. It is even faster than Palmer's. It does the first test number in 1 min 33 sec and the second one in 1 min 11 sec.
To use it, load side 1, RST R/S. Then re-load side 1, enter the integer and press R/S. The integer and its factors will be printed.

By the way, somewhere else in this issue I name all the people who produced correct solutions to George Vogel's programming puzzles. Because Björn's letter was catalogued under the heading "SFF programs" I missed him on the puzzles. My apologies. He also produced the correct answers to all five puzzles. That brings up the subject of "more than one subject in one page." It might be a good idea, if you you talk about more than one subject in your letter, please put them on separate pages. It makes my filing a little easier and avoids omissions like the one above.

The third program comes from Robert Caldwell in Sunnyvale, CA. Bob is an "old" SR-52 programmer. He found that, surprisingly, some of the old SR-52 SFF versions translated to TI-59, run slower than originally on the 52. But, the exceptions confirm the rule. So, when Bob translated PPX # 360016, by Gerald J. Kovacic of Brooklyn, NY, he found that it runs faster than Bill Skillman's original program. It handles the first test number in 2 min 2 sec and the second one in 1 min 37 sec. While it is slower than both "fast mode" programs, it might be the basis for a super-fast fast mode SFF program. I have already one version, but I hope its times will be beaten by newer versions I hope to receive. By the way, as with most SFF programs, this one is slow for small numbers and fast with larger ones.

In recent issues of the HP PPC JOURNAL I found that the HP members were asked whether their "fine-tuned" Fast Factoring Program should be a challenge to TI-59 users. If they accept, why not? I am not sure that the TI-59 is a worthy challenger to the HP-41C in this field, but our fast mode should make up for its deficiencies. I know also that Richard Vanderburgh once talked about the existence of an HP-67/69 SFF program that run our second test number at under one minute. Let's give it a whirl.

The last program is Bill Skillman's version of an SR-56 SFF program. To use it, enter the program, enter the integer, press RST R/S. All factors will be printed. If handheld, replace step 47 with a NOP, step 48 with a +/- and step 80 with R/S. Press R/S for each next factor. The last factor will be displayed as a negative value.

000 00 0	027 43 RCL	054 52 EE	081 42 STO	107 03 03	133 03 03	159 01 01
001 00 0	028 02 02	055 22 INV	082 03 03	108 04 4	134 04 4	160 34 fX
002 00 0	029 95 =	056 52 EE	083 61 GTO	109 61 GTO	135 61 GTO	161 75 -
003 00 0	030 22 INV	057 91 R/S	084 00 00	110 00 00	136 00 00	162 43 RCL
004 00 0	031 59 INT	058 61 GTO	085 21 21	111 22 22	137 22 22	163 02 02
005 36 PGM	032 67 EQ	059 00 00	086 07 7	112 06 6	138 07 7	164 95 =
006 02 02	033 00 00	060 24 24	087 44 SUM	113 44 SUM	139 44 SUM	165 77 GE
007 71 SBR	034 37 37	061 52 EE	088 03 03	114 03 03	140 03 03	166 01 01
008 02 02	035 83 GO*	062 22 INV	089 01 1	115 61 GTO	141 06 6	167 77 77
009 39 39	036 03 03	063 52 EE	090 61 GTO	116 00 00	142 61 GTO	168 43 RCL
010 09 9	037 43 RCL	064 91 R/S	091 00 00	117 21 21	143 00 00	169 01 01
011 00 0	038 01 01	065 76 LBL	092 22 22	118 07 7	144 22 22	170 99 PRT
012 00 0	039 32 X:T	066 15 E	093 06 6	119 44 SUM	145 06 6	171 69 OP
013 00 0	040 43 RCL	067 29 CP	094 44 SUM	120 03 03	146 44 SUM	172 69 69
014 00 0	041 02 02	068 47 CMS	095 03 03	121 04 4	147 03 03	173 98 ADV
015 25 CLR	042 22 INV	069 42 STO	096 61 GTO	122 61 GTO	148 61 GTO	174 61 GTO
016 22 INV	043 49 PRD	070 01 01	097 00 00	123 00 00	149 00 00	175 00 00
017 58 FIX	044 01 01	071 99 PRT	098 21 21	124 22 22	150 21 21	176 61 61
018 61 GTO	045 67 EQ	072 98 ADV	099 06 6	125 06 6	151 07 7	177 01 1
019 00 00	046 01 01	073 94 +/-	100 44 SUM	126 44 SUM	152 44 SUM	178 01 1
020 61 61	047 70 70	074 69 OP	101 03 03	127 03 03	153 03 03	179 02 2
021 02 2	048 99 PRT	075 07 07	102 61 GTO	128 61 GTO	154 06 6	180 42 STO
022 44 SUM	049 29 CP	076 69 OP	103 00 00	129 00 00	155 61 GTO	181 03 03
023 02 02	050 87 IFF	077 19 19	104 21 21	130 21 21	156 00 00	182 61 GTO
024 43 RCL	051 07 07	078 25 CLR	105 07 7	131 07 7	157 22 22	183 01 01
025 01 01	052 00 00	079 08 8	106 44 SUM	132 44 SUM	158 43 RCL	184 08 08
026 55 ÷	053 24 24	080 06 6				

000 00 0	019 11 A	037 02 2	055 04 4	073 04 4	091 61 GTO	109 40 IND
001 00 0	020 29 CP	038 83 GO*	056 83 GO*	074 09 9	092 00 00	110 04 04
002 00 0	021 42 STO	039 03 03	057 03 03	075 42 STO	093 20 20	111 43 RCL
003 00 0	022 01 01	040 01 1	058 02 2	076 04 04	094 44 SUM	112 01 01
004 00 0	023 98 ADV	041 83 GO*	059 83 GO*	077 43 RCL	095 02 02	113 32 X:T
005 36 PGM	024 99 PRT	042 03 03	060 03 03	078 01 01	096 03 3	114 43 RCL
006 02 02	025 98 ADV	043 02 2	061 04 4	079 34 fX	097 44 SUM	115 02 02
007 71 SBR	026 25 CLR	044 83 GO*	062 83 GO*	080 75 -	098 04 04	116 67 EQ
008 02 02	027 42 STO	045 03 03	063 03 03	081 43 RCL	099 43 RCL	117 00 00
009 39 39	028 02 02	046 02 2	064 06 6	082 02 02	100 01 01	118 89 89
010 09 9	029 09 9	047 83 GO*	065 83 GO*	083 95 =	101 55 ÷	119 22 INV
011 00 0	030 04 4	048 03 03	066 03 03	084 77 GE	102 43 RCL	120 49 PRD
012 22 INV	031 42 STO	049 04 4	067 02 2	085 00 00	103 02 02	121 01 01
013 58 FIX	032 03 03	050 83 GO*	068 83 GO*	086 49 49	104 95 =	122 99 PRT
014 22 INV	033 03 3	051 03 03	069 03 03	087 43 RCL	105 22 INV	123 29 CP
015 57 ENG	034 07 7	052 02 2	070 06 6	088 01 01	106 59 INT	124 61 GTO
016 25 CLR	035 42 STO	053 83 GO*	071 83 GO*	089 99 PRT	107 22 INV	125 00 00
017 91 R/S	036 04 04	054 03 03	072 03 03	090 91 R/S	108 67 EQ	126 99 99
018 76 LBL						

000 61 GTO	028 09 9	056 42 STO	084 14 D	112 14 D	140 14 D	169 97 DSZ
001 76 LBL	029 07 7	057 01 01	085 04 4	113 02 2	141 04 4	170 04 04
002 76 LBL	030 54)	058 14 D	086 14 D	114 14 D	142 14 D	171 00 00
003 14 D	031 55 ÷	059 01 1	087 02 2	115 04 4	143 02 2	172 71 71
004 44 SUM	032 02 2	060 14 D	088 14 D	116 14 D	144 14 D	173 43 RCL
005 01 01	033 01 1	061 02 2	089 04 4	117 02 2	145 04 4	174 02 02
006 43 RCL	034 00 0	062 14 D	090 14 D	118 14 D	146 14 D	175 99 PRT
007 02 02	035 54)	063 02 2	091 06 6	119 04 4	147 06 6	176 98 ADV
008 55 ÷	036 59 INT	064 14 D	092 14 D	120 14 D	148 14 D	177 91 R/S
009 43 RCL	037 95 =	065 04 4	093 06 6	121 08 8	149 02 2	178 43 RCL
010 01 01	038 77 GE	066 14 D	094 14 D	122 14 D	150 14 D	179 01 01
011 95 =	039 01 01	067 25 CLR	095 02 2	123 06 6	151 06 6	180 22 INV
012 22 INV	040 73 73	068 13 C	096 14 D	124 14 D	152 14 D	181 49 PRD
013 59 INT	041 42 STO	069 86 STF	097 06 6	125 04 4	153 04 4	182 02 02
014 67 EQ	042 04 04	070 00 00	098 14 D	126 14 D	154 14 D	183 99 PRT
015 01 01	043 42 STO	071 02 2	099 04 4	127 06 6	155 02 2	184 22 INV
016 78 78	044 03 03	072 14 D	100 14 D	128 14 D	156 14 D	185 87 IFF
017 92 RTN	045 92 RTN	073 04 4	101 02 2	129 02 2	157 04 4	186 00 00
018 76 LBL	046 76 LBL	074 14 D	102 14 D	130 14 D	158 14 D	187 00 00
019 13 C	047 11 A	075 02 2	103 06 6	131 04 4	159 02 2	188 06 06
020 75 -	048 29 CP	076 14 D	104 14 D	132 14 D	160 14 D	189 43 RCL
021 53 (	049 99 PRT	077 04 4	105 04 4	133 06 6	161 00 0	190 04 04
022 53 (	050 42 STO	078 14 D	106 14 D	134 14 D	162 14 D	191 75 -
023 43 RCL	051 02 02	079 06 6	107 06 6	135 02 2	163 02 2	192 43 RCL
024 02 02	052 81 RST	080 14 D	108 14 D	136 14 D	164 14 D	193 03 03
025 34 fX	053 76 LBL	081 02 2	109 08 8	137 06 6	165 01 1	194 13 C
026 85 +	054 76 LBL	082 14 D	110 14 D	138 14 D	166 00 0	195 61 GTO
027 01 1	055 01 1	083 06 6	111 04 4	139 06 6	167 14 D	196 00 00
					168 14 D	197 06 06

00 56 CP	13 01 1	25 05 5	37 48 VX	49 41 R/S	61 02 2	73 02 2
01 38 CMS	14 57 SUBR	26 00 0	38 74 -	50 04 4	62 94 =	74 12 INV
02 33 STO	15 05 5	27 04 4	39 34 RCL	51 57 SUBR	63 12 INV	75 30 PROD
03 01 1	16 04 4	28 57 SUBR	40 02 2	52 05 5	64 29 INT	76 01 1
04 97 PRT	17 06 6	29 05 5	41 94 =	53 05 5	65 37 X=T	77 37 X=T
05 98 PAP	18 57 SUBR	30 05 5	42 47 X>T	54 02 2	66 06 6	78 04 4
06 15 CLR	19 05 5	31 06 6	43 01 1	55 35 SUM	67 09 9	79 07 7
07 57 SUBR	20 05 5	32 57 SUBR	44 04 4	56 02 2	68 58 RTN	80 97 PRT
08 05 5	21 57 SUBR	33 05 5	45 34 RCL	57 34 RCL	69 34 RCL	81 56 CP
09 04 4	22 05 5	34 05 5	46 01 1	58 01 1	70 01 1	82 22 GTO
10 01 01	23 00 00	35 34 RCL	47 97 PRT	59 54 DIV	71 32 X:T	83 05 5
11 57 SUBR	24 57 SUBR	36 01 1	48 98 PAP	60 34 RCL	72 34 RCL	84 07 7
12 05 5						

TRANSFORM REGRESSION - by John Worthington and Emil Regelman

Introduction: This program will perform a regression analysis on entered data, following the selection of one of eight available transforms. Additional regression analyses of different transform functions may be conveniently performed, without reentering the data. The program can also list the original or transformed data, as desired.

I. Initializing Program: Press [E'] to initialize the program. The first time [E'] is used, the program will tabulate the available transforms and then print "RST" to indicate that the program is ready to accept data. Subsequently, [E'] will just initialize memories, and print [RST].

II. Specifying Transforms: Enter the desired transforms according to the tabulated listing (see I.), and press [B]. Alternate transforms may be entered, as desired. For example, an entry of 4.8 will result in the transforms X^2 and Y^3. If no transforms are specified, the program will perform a standard linear regression analysis of the data.

III. Entering Data: The X, Y data pairs are stored and printed by entering the X value, pressing [$X \gtrless T$], entering the Y value and pressing [A]. Continue entering pairs of data as desired. The program will accommodate data in groups of 15 pairs.

a. If 15 or fewer data pairs are stored, the data do not have to be recorded.

b. When the 16th data pair is entered, the program will ask for a data card by printing "WRT". Insert a blank card and continue entering data. Repeat as many times as necessary. The cards containing the recorded data groups should be sequentially numbered, so that they may be reinserted in the correct order. When all data have been stored, press [E] and insert another blank card to record the final data points.

IV. Deleting Data:

a. If a data pair just entered was in error, press [A']. The calculator will perform the deletion and print "$\Sigma -$", along with the deleted values. (The X and Y values will be recalled into "T" and the display, respectively.) To defeat the deletion (if desired) press [A] which will reenter the data. Continue to press [A'] to delete as many data pairs as desired. In situations where the deletions extend into another group (previously recorded), the calculator will request the insertion of a card by printing "WRT". When the card is inserted the calculator will record the modified data and request the insertion of the card on which the previous set was recorded by printing "READ". The calculator will then delete the last pair of this group of 15 pairs. Also note that this group has now been modified and must be recorded before computations are done.

b. If the data to be deleted are not the pair just entered;

1. Enter the number of the pair to be deleted, and press [D].
2. Then, press [A']. The deleted data will be printed, as noted earlier.
3. Additional data sets may be deleted by repeating 1. and 2., or as described in IV. a., above.
4. When all desired deletions have been done, the following additional operations must be performed to reestablish the appropriate counting memories.
   i. Enter the new number of data pairs, and press [D].
   ii. To this number, add the total number of deletions, and press [SBR][SBR].
5. It would be useful at this point to list the stored data (see VI) to check for errors. If this is done, step 4 (ii) may be omitted.

V. Performing the Regression Analysis:

a. If less than 16 data pairs have been entered, press [C] to initiate the transform regression (note that the designated transforms are printed at this time). To compute another transform regression, specify the new transform (see II) and press [C].

b. If more than 15 data sets have been entered (and recorded), be sure to record the last-entered group of data (by pressing [E] and inserting a blank card) before proceeding. Insert the first data card and press [C]. Then insert the card on which the next group of data pairs was recorded. Continue to insert the cards as they are read by the calculator. The cards must be inserted in the order in which they were recorded.

c. As data pairs are processed, each X value is briefly displayed (PAU), so that the user can verify that the correct data are being processed; if an incorrect data group was inserted, press [R/S] and RST] and then start again.

d. When the summations are completed, the following will be printed:

N = the number of summations actually performed. (Any transforms which generate an error condition will be ignored by the program).
SLP. = the slope of the regression line.
YINT = the Y intercept of the regression line.
XINT = the X intercept of the regression line.
R^2 = the correlation coefficient, squared.

e. Steps II and V may be repeated as many times as needed to find the combination of transforms that yields the highest R^2 value, or the most suitable regression line.

VI. Listing Stored Data: Press [C'] to list all stored data. If more than 15 pairs have been stored, it will be necessary to insert the data cards as described in V. b., above.

VII. Listing Transformed Data: Press [B'] to list the transformed data. If more than 15 pairs have been stored, it will be necessary to insert the data cards as described in V. b., above.

/SEE PROGRAM LISTING ON NEXT PAGE./

GUARD DIGITS PRINTER.- I am not sure if you'll believe me, but this is the last, the ultimate, the final 13-digits printer I am going to shove down your collective throats.Enter the number in the display and press A. The 10th, 11th, 12th and 13th digits are printed in OP 04 space. My apologies for the sarcasm to Karl-Joseph Meusch, the author of this well-written routine.

3.141592654 5359

```
000  76 LBL    014  58 FIX    028  22 INV    042  69 OP'    056  85  +
001  11  A     015  00  00    029  52 EE     043  06  06    057  01  1
002  53  (     016  52 EE     030  22 INV     044  92 RTN    058  85  +
003  42 STO    017  94 +/-    031  59 INT     045  76 LBL    059  28 LOG
004  00  00    018  65  ×     032  10 E'     046  10 E'     060  59 INT
005  50 I×I    019  00  0     033  10 E'     047  53  (     061  65  ×
006  65  ×     020  42 STO    034  10 E'     048  24 CE     062  02  2
007  53  (     021  01  01    035  10 E'     049  65  ×     063  75  -
008  52 EE     022  01  1     036  43 RCL    050  01  1     064  59 INT
009  55  ÷     023  52 EE     037  01  01    051  00  0     065  44 SUM
010  52 EE     024  07  7     038  69 OP     052  49 PRD    066  01  01
011  00  0     025  54  )     039  04  04    053  01  01    067  54  )
012  00  0     026  22 INV    040  43 RCL    054  49 PRD    068  92 RTN
013  54  )     027  58 FIX    041  00  00    055  01  01
```

TRANSFORM REGRESSION.-

John Worthington &
Emil Regelman.

```
000 92 RTN
001 68 NOP
002 68 NOP
003 23 LNX
004 92 RTN
005 68 NOP
006 35 1/X
007 92 RTN
008 68 NOP
009 34 √X
010 92 RTN
011 68 NOP
012 33 X²
013 92 RTN
014 68 NOP
015 33 X²
016 35 1/X
017 92 RTN
018 22 INV
019 23 LNX
020 92 RTN
021 55 ÷
022 32 X:T
023 92 RTN
024 45 YX
025 03 3
026 92 RTN
027 76 LBL
028 19 D'
029 85 +
030 01 1
031 52 EE
032 06 6
033 54 )
034 82 HIR
035 08 08
036 25 CLR
037 32 X:T
038 69 OP
039 06 06
040 92 RTN
041 76 LBL
042 12 B
043 82 HIR
044 02 02
045 75 -
046 59 INT
047 42 STO
048 08 08
049 95 =
050 42 STO
051 10 10
052 03 3
053 49 PRD
054 08 08
055 03 3
056 00 0
057 49 PRD
058 10 10
059 82 HIR
060 12 12
061 92 RTN
062 76 LBL
063 16 A'
064 03 3
065 01 1
066 32 X:T
067 43 RCL
068 09 09
069 77 GE
070 00 00
071 82 82
072 15 E
073 71 SBR
074 04 04
075 36 36
076 22 INV
077 96 WRT
078 05 5
079 09 9
080 42 STO
081 09 09
082 69 OP
083 00 00
084 07 7
085 07 7
086 02 2
087 00 0
088 69 OP
089 02 02
090 69 OP
091 05 05
092 69 OP
093 39 39
094 00 0
095 23 LNX
096 63 EX*
097 09 09
098 24 CE
099 99 PRT
100 32 X:T
101 00 0
102 23 LNX
103 69 OP
104 29 29
105 63 EX*
106 09 09
107 24 CE
108 99 PRT
109 98 ADV
110 69 OP
111 30 30
112 69 OP
113 39 39
114 69 OP
115 39 39
116 81 RST
117 76 LBL
118 11 A
119 32 X:T
120 69 OP
121 29 29
122 72 ST*
123 09 09
124 69 OP
125 19 19
126 87 IFF
127 07 07
128 01 01
129 46 46
130 72 ST*
131 09 09
132 99 PRT
133 32 X:T
134 99 PRT
135 98 ADV
136 69 OP
137 29 29
138 72 ST*
139 09 09
140 69 OP
141 20 20
142 29 CP
143 43 RCL
144 00 00
145 81 RST
146 85 +
147 15 E
148 03 3
149 00 0
150 42 STO
151 09 09
152 00 0
153 95 =
154 61 GTO
155 01 01
156 30 30
157 76 LBL
158 10 E'
159 22 INV
160 58 FIX
161 04 4
162 69 OP
163 17 17
164 71 SBR
165 40 IND
166 08 08
167 25 CLR
168 06 6
169 69 OP
170 17 17
171 36 PGM
172 01 01
173 71 SBR
174 25 CLR
175 42 STO
176 08 08
177 42 STO
178 10 10
179 14 D
180 86 STF
181 09 09
182 81 RST
183 03 3
184 22 INV
185 96 WRT
186 03 3
187 01 1
188 42 STO
189 09 09
190 61 GTO
191 02 02
192 95 95
193 76 LBL
194 18 C'
195 98 ADV
196 86 STF
197 00 00
198 86 STF
199 03 03
200 61 GTO
201 02 02
202 82 82
203 76 LBL
204 15 E
205 22 INV
206 58 FIX
207 04 4
208 03 3
209 03 3
210 05 5
211 03 3
212 07 7
213 71 SBR
214 04 04
215 44 44
216 96 WRT
217 92 RTN
218 76 LBL
219 17 B'
220 86 STF
221 03 03
222 86 STF
223 02 02
224 76 LBL
225 13 C
226 01 1
227 02 2
228 22 INV
229 28 LOG
230 85 +
231 01 1
232 03 3
233 07 7
234 03 3
235 85 +
236 82 HIR
237 05 05
238 01 1
239 00 0
240 00 0
241 95 =
242 82 HIR
243 07 07
244 43 RCL
245 08 08
246 55 ÷
247 03 3
248 85 +
249 01 1
250 01 1
251 95 =
252 42 STO
253 07 07
254 73 RC*
255 07 07
256 69 OP
257 02 02
258 43 RCL
259 10 10
260 55 ÷
261 03 3
262 85 +
263 02 2
264 00 0
265 95 =
266 42 STO
267 07 07
268 73 RC*
269 07 07
270 69 OP
271 04 04
272 69 OP
273 05 05
274 87 IFF
275 03 03
276 02 02
277 82 82
278 36 PGM
279 01 01
280 71 SBR
281 25 CLR
282 02 2
283 09 9
284 42 STO
285 09 09
286 43 RCL
287 00 00
288 42 STO
289 07 07
290 25 CLR
291 02 2
292 44 SUM
293 09 09
294 29 CP
295 06 6
296 00 0
297 32 X:T
298 43 RCL
299 09 09
300 77 GE
301 01 01
302 83 83
303 73 RC*
304 09 09
305 82 HIR
306 04 04
307 22 INV
308 86 STF
309 07 07
310 69 OP
311 19 19
312 87 IFF
313 07 07
314 02 02
315 90 90
316 32 X:T
317 69 OP
318 39 39
319 73 RC*
320 09 09
321 87 IFF
322 00 00
323 03 03
324 53 53
325 87 IFF
326 02 02
327 03 03
328 30 30
329 66 PAU
330 71 SBR
331 40 IND
332 08 08
333 95 =
334 82 HIR
335 03 03
336 73 RC*
337 09 09
338 32 X:T
339 82 HIR
340 14 14
341 71 SBR
342 40 IND
343 10 10
344 95 =
345 32 X:T
346 82 HIR
347 13 13
348 22 INV
349 87 IFF
350 02 02
351 03 03
352 58 58
353 99 PRT
354 32 X:T
355 99 PRT
356 32 X:T
357 98 ADV
358 32 X:T
359 22 INV
360 86 STF
361 07 07
362 69 OP
363 19 19
364 87 IFF
365 07 07
366 03 03
367 73 73
368 87 IFF
369 03 03
370 03 03
371 73 73
372 78 Σ+
373 69 OP
374 29 29
375 97 DSZ
376 07 07
377 02 02
378 90 90
379 25 CLR
380 87 IFF
381 03 03
382 04 04
383 35 35
384 43 RCL
385 03 03
386 32 X:T
387 03 3
388 01 1
389 19 D'
390 69 OP
391 12 12
392 03 3
393 06 6
394 93 .
395 02 2
396 07 7
397 03 3
398 03 3
399 04 4
400 19 D'
401 69 OP
402 12 12
403 32 X:T
404 04 4
405 05 5
406 93 .
407 02 2
408 04 4
409 03 3
410 01 1
411 03 3
412 07 7
413 42 STO
414 07 07
415 19 D'
416 00 0
417 69 OP
418 15 15
419 32 X:T
420 69 OP
421 37 37
422 43 RCL
423 07 07
424 19 D'
425 69 OP
426 13 13
427 33 X²
428 32 X:T
429 03 3
430 05 5
431 93 .
432 07 7
433 19 D'
434 98 ADV
435 81 RST
436 03 3
437 05 5
438 01 1
439 07 7
440 01 1
441 03 3
442 01 1
443 06 6
444 69 OP
445 00 00
446 24 CE
447 69 OP
448 02 02
449 69 OP
450 05 05
451 03 3
452 92 RTN
453 76 LBL
454 14 D
455 42 STO
456 00 00
457 76 LBL
458 71 SBR
459 75 -
460 01 1
461 95 =
462 55 ÷
463 01 1
464 05 5
465 95 =
466 22 INV
467 59 INT
468 65 ×
469 03 3
470 00 0
471 85 +
472 03 3
473 01 1
474 95 =
475 52 EE
476 42 STO
477 09 09
478 25 CLR
479 92 RTN
480 98 ADV
481 03 3
482 07 7
483 03 3
484 05 5
485 01 1
486 03 3
487 03 3
488 01 1
489 03 3
490 06 6
491 69 OP
492 01 01
493 02 2
494 01 1
495 03 3
496 02 2
497 03 3
498 05 5
499 03 3
500 00 0
501 01 1
502 03 3
503 69 OP
504 02 02
505 03 3
506 07 7
507 02 2
508 04 4
509 03 3
510 02 2
511 03 3
512 01 1
513 00 0
514 00 0
515 69 OP
516 03 03
517 01 1
518 05 5
519 03 3
520 02 2
521 01 1
522 06 6
523 01 1
524 07 7
525 03 3
526 06 6
527 69 OP
528 04 04
529 69 OP
530 05 05
531 69 OP
532 00 00
533 01 1
534 09 9
535 42 STO
536 07 07
537 09 9
538 48 EXC
539 00 00
540 82 HIR
541 04 04
542 01 1
543 00 0
544 00 0
545 65 ×
546 33 X²
547 82 HIR
548 05 05
549 95 =
550 82 HIR
551 07 07
552 08 8
553 00 0
554 93 .
555 00 0
556 00 0
557 00 0
558 01 1
559 06 6
560 04 4
561 82 HIR
562 35 35
563 85 +
564 93 .
565 04 4
566 95 =
567 82 HIR
568 37 37
569 08 8
570 22 INV
571 44 SUM
572 07 07
573 73 RC*
574 07 07
575 69 OP
576 02 02
577 09 9
578 44 SUM
579 07 07
580 73 RC*
581 07 07
582 55 ÷
583 01 1
584 00 0
585 00 0
586 95 =
587 69 OP
588 04 04
589 69 OP
590 05 05
591 22 INV
592 97 DSZ
593 00 00
594 06 06
595 14 14
596 01 1
597 00 0
598 00 0
599 00 0
600 00 0
601 93 .
602 00 0
603 00 0
604 00 0
605 01 1
606 82 HIR
607 35 35
608 95 =
609 82 HIR
610 37 37
611 61 GTO
612 05 05
613 69 69
614 82 HIR
615 14 14
616 48 EXC
617 00 00
618 98 ADV
619 92 RTN
```

DATA

480.	08
0.	09
0.	10
4400000000.	11
2731004400.	12
263440000.	13
5244000000.	14
4470000000.	15
263447000.	16
5466000000.	17
4463450000.	18
4466840000.	19
4500000000.	20
2731004500.	21
263450000.	22
5245000000.	23
4570000000.	24
8263457000.	25
5445000000.	26
4563440000.	27
4566840000.	28
45.243137	29

LABELS

028	19	D'
042	12	B
063	16	A'
118	11	A
158	10	E'
194	18	C'
204	15	E
219	17	B'
225	13	C
454	14	D
458	71	SBR

FAST ALPHA SORT.- In response to the RPN programmer's challenge in v5n8, Richard Snow of Vallejo, California, has written this program. It will sort up to 99 five-letter words. Richard has discovered that a progarm in fast mode can be stopped with a "hard" display if PAU or PRT is used just before R/S. Steps 019 and 020 reflect this principle. The decimal point trick is used here to start the Shell sort routine. We all know that the Shell sort is not the only method available. I hope that this first entry will encourage other good programmers to try new and innovative methods to speed up the sorting.

Recording instructions: Press 10 OP 17 and load regs 01 through 97 and key in program. Then press 9 OP 17 and key in program steps 160 through 174. Be careful not to delete or insert here, because it would destroy the contents of the data registers. Now press 6 OP 17 and record four card sides. Two registers will not fit on the cards (98 and 99). They will be entered from the keyboard later.

Program test: Insert card side 1 and press A. Re-insert card side 1. Now you may either enter manually any number of registers (contents) by pressing R/S each time OR you may force sides 2, 3 and 4 by -2, -3 and -4. In the first case if you want to correct a bad entry, enter reg # +/- R/S, then re-enter the code. If you use the cards, enter now 98 +/- R/S, then enter 13 27 27 32 45 (ALLOY) R/S, followed by 13 27 24 22 31 (ALIGN) R/S. To sort press R/S again. Wait about 6½ minutes. To sort again the words, now in correct order, enter 100 +/- R/S. Wait about 3 minutes.

On the next page Richard explains how the program works.

PRT CODE REGISTERS

4632311736	ZONES	01
4617311735	ZENER	02
4532413723	YOUTH	03
4524172716	YIELD	04
4417353244	XEROX	05
4417313231	XENON	06
4335243717	WRITE	07
4332353723	WORTH	08
4313363717	WASTE	09
4224422416	VIVID	10
4224363235	VISOR	11
4213272416	VALID	12
4213224117	VAGUE	13
4136413533	USURP	14
4131243717	UNITE	15
4130141735	UMBER	16
3735413126	TRUNK	17
3735132437	TRAIT	18
3724301716	TIMED	19
3717441336	TEXAS	20
3713142717	TABLE	21
3637133717	STATE	22
3632311335	SONAR	23
3632272416	SOLID	24
3623321526	SHOCK	25
3623172727	SHELL	26
3517362431	RESIN	27
3517271345	RELAY	28
3517131537	REACT	29
3513332416	RAPID	30
3513162424	RADII	31
3513161335	RADAR	32
3441243526	QUIRK	33
3441173637	QUEST	34
3341273617	PULSE	35
3335243137	PRINT	36
3335243017	PRIME	37
3332431735	POWER	38
3332271335	POLAR	39
3327133117	PLANE	40
3324273237	PILOT	41
3323323732	PHOTO	42
3323323117	PHONE	43
3235221331	ORGAN	44
3230172213	OMEGA	45
3215371327	OCTAL	46
3132371736	NOTES	47
3132353723	NORTH	48
3032351327	MORAL	49
3032161727	MODEL	50
3017371735	METER	51
3017352217	MERGE	52
3013153532	MACRO	53
2732222415	LOGIC	54
2724371735	LITER	55
2724222337	LIGHT	56
2717421735	LEVER	57
2717313617	LENSE	58
2631413527	KNURL	59
2631324331	KNOWN	60
2613351337	KARAT	61
2532412717	JOULE	62
2532243137	JOINT	63
2436364117	ISSUE	64
2431364137	INPUT	65
2431173537	INERT	66
2431161744	INDEX	67
2332313235	HONOR	68
2317353746	HERTZ	69
2235133323	GRAPH	70
2235132431	GRAIN	71
2213371716	GATED	72
2213303013	GAMMA	73
2132351517	FORCE	74
2132154136	FOCUS	75
1734411327	EQUAL	76
1733353230	EPROM	77
1731373545	ENTRY	78
1730243736	EMITS	79
1727171537	ELECT	80
1624321617	DIODE	81
1624222437	DIGIT	82
1617313617	DENSE	83
1617273713	DELTA	84
1545152717	CYCLE	85
1541354217	CURVE	86
1541142415	CUBIC	87
1527321526	CLOCK	88
1527171335	CLEAR	89
1435243117	BRINE	90
1432133516	BOARD	91
1413362415	BASIC	92
1344241327	AXIAL	93
1341162432	AUDIO	94
1336152424	ASCII	95
1331222717	ANGLE	96
1327332313	ALPHA	97

000	55	÷
001	53	(
002	34	FX
003	28	LOG
004	59	INT
005	75	-
006	04	4
007	54	)
008	22	INV
009	28	LOG
010	33	X²
011	95	=
012	72	ST*
013	00	00
014	69	OP
015	20	20
016	43	RCL
017	00	00
018	66	PAU
019	91	R/S
020	93	.
021	67	EQ
022	00	00
023	33	33
024	77	GE
025	00	00
026	00	00
027	50	I×I
028	42	STO
029	00	00
030	61	GTO
031	00	00
032	18	18
033	43	RCL
034	00	00
035	82	HIR
036	08	08
037	82	HIR
038	05	05
039	55	÷
040	02	2
041	95	=
042	59	INT
043	82	HIR
044	07	07
045	82	HIR
046	55	55
047	01	1
048	85	+
049	42	STO
050	00	00
051	73	RC*
052	00	00
053	32	X:T
054	82	HIR
055	17	17
056	44	SUM
057	00	00
058	73	RC*
059	00	00
060	77	GE
061	00	00
062	83	83
063	32	X:T
064	72	ST*
065	00	00
066	82	HIR
067	17	17
068	94	+/-
069	44	SUM
070	00	00
071	32	X:T
072	72	ST*
073	00	00
074	01	1
075	32	X:T
076	44	SUM
077	00	00
078	43	RCL
079	00	00
080	77	GE
081	00	00
082	51	51
083	82	HIR
084	15	15
085	32	X:T
086	01	1
087	85	+
088	22	INV
089	77	GE
090	00	00
091	49	49
092	25	CLR
093	82	HIR
094	18	18
095	82	HIR
096	05	05
097	82	HIR
098	17	17
099	32	X:T
100	01	1
101	03	3
102	22	INV
103	77	GE
104	00	00
105	43	43
106	04	4
107	22	INV
108	77	GE
109	00	00
110	43	43
111	01	1
112	22	INV
113	67	EQ
114	00	00
115	43	43
116	02	2
117	82	HIR
118	68	68
119	61	GTO
120	01	01
121	30	30
122	44	SUM
123	00	00
124	73	RC*
125	00	00
126	69	OP
127	03	03
128	69	OP
129	05	05
130	01	1
131	85	+
132	42	STO
133	00	00
134	32	X:T
135	73	RC*
136	00	00
137	69	OP
138	01	01
139	82	HIR
140	18	18
141	67	EQ
142	01	01
143	25	25
144	77	GE
145	01	01
146	22	22
147	98	ADV
148	29	CP
149	01	1
150	61	GTO
151	00	00
152	27	27
153	69	OP
154	00	00
155	22	INV
156	58	FIX
157	61	GTO
158	00	00
159	14	14
160	00	0
161	00	0
162	76	LBL
163	11	A
164	29	CP
165	36	PGM
166	02	02
167	71	SBR
168	02	02
169	40	40
170	89	π
171	01	1
172	00	0
173	69	OP
174	17	17

ALIGN	MORAL
ALLOY	NORTH
ALPHA	NOTES
ANGLE	OCTAL
ASCII	OMEGA
AUDIO	ORGAN
AXIAL	PHONE
BASIC	PHOTO
BOARD	PILOT
BRINE	PLANE
CLEAR	POLAR
CLOCK	POWER
CUBIC	PRIME
CURVE	PRINT
CYCLE	PULSE
DELTA	QUEST
DENSE	QUIRK
DIGIT	RADAR
DIODE	RADII
ELECT	RAPID
EMITS	REACT
ENTRY	RELAY
EPROM	RESIN
EQUAL	SHELL
FOCUS	SHOCK
FORCE	SOLID
GAMMA	SONAR
GATED	STATE
GRAIN	TABLE
GRAPH	TEXAS
HERTZ	TIMED
HONOR	TRAIT
INDEX	TRUNK
INERT	UMBER
INPUT	UNITE
ISSUE	USURP
JOINT	VAGUE
JOULE	VALID
KARAT	VISOR
KNOWN	VIVID
KNURL	WASTE
LENSE	WORTH
LEVER	WRITE
LIGHT	XENON
LITER	XEROX
LOGIC	YIELD
MACRO	YOUTH
MERGE	ZENER
METER	ZONES
MODEL	

FAST ALPHA SORT. - This program is an enhanced version of the Alphabetical Sort program which was previously published in TI PPC NOTES. The program was re-written in response to the RPN programmers' challenge described in V5N8. A list of 99 words in reverse alphabetical order can be sorted in 6 minutes and 34 seconds. It takes only 3 minutes and 1 second for the program to sort a list of words which is already in alphabetical order. Another 40 seconds are needed to print the list in two columns.

The fast mode is initialized in LBL A to speed up the execution of the program. The " π 1 " at steps 170 and 171 were chosen so that the program counter would be moved to step 144 during the fast mode initialization. The partition is changed (10 OP 17) at step 148 instead of 174 due to this re-location. The program then continues to step 152 and on into the main program.

The first part of the program left-justifies the print code before storing it into a register. This allows words of varying lengths to be alphabetized. The routine will also handle print code for numbers. Alpha-numeric data is sorted first by number and then by letter. (0 to 9 then A to Z) Many computer reports sort alpha-numeric data the other way around. (A to Z then 0 to 9) This can be accomplished in this program simply by using print codes 81 to 92 for numeric data.

Since user labels cannot be used in the fast mode, another method was devised to choose the various program options. (1) To enter the alpha-numeric data, key in the print code and press R/S. The next data number will be displayed. The data number is also the register number where the next entry will be stored. (2) To correct a bad entry or change the contents of any register, enter the number of the register to be corrected and press " +/- R/S ". Enter the revised print code and press R/S. (3) When all the data is entered, simply press R/S. The decimal point trick will branch to the Shell sorting routine and an alphabetized list will finally be printed.

The same optimized Shell sort routine is used in this program as in the original Alphabetical Sort program and is described in V5 N4 P5.

The print code is not reconstructed as in the Alphabetical Sort program, but is kept in a left justified format. This saves program execution time and also saves enough program steps so that all one hundred registers can be used.

The contents of the registers are printed in two columns using the same method which is used in the Alphabetical Sort program described in V5 N4 P6.

This program version was not designed to sort pre-stored print code unless it is already left justified. The program can be modified slightly if you need to sort print code stored in registers and recorded on magnetic cards. First, change 150 to R/S so the program can be stopped when it is finished. Initialize the program by pressing A and re-enter the mag strip with the R/S modification. Force the magnetic cards with print code into the appropriate banks using -2, -3, and -4. This protects data in the registers from being destroyed and assures that the cards will be read while in the 159.99 partition. Enter the LRN mode and backstep to step 018. Change steps 018 to 020 to " NOP RCL IND 00 ". Go out of LRN mode, enter the first register number to be left justified. Press " +/- R/S ". The program will sort the registers up to the register which contains a zero.

(over)

INVERSE LIST PRINT ALL - FAST MODE

by John Worthington and Emil Regelman

This program will list all digits, signs and exponents (including guard digits) of numbers stored in specified memories, similar to the inverse list operation. The program can also be used to print the entire number in the display, when operated in the normal mode.

1. Record all memories (to be listed) onto magnetic cards.
2. Read side one of the program into the calculator, and press [A]*.
3. Reinsert side one of the program (a zero will be displayed).
4. Insert side two of the program (a -4 will be displayed).
5. Insert the data card on which the first memory to be listed was recorded.
6. Designate the memories to be listed as follows: initial "." final**.
7. a. Press [SBR][0][2][3] to left-hand justify the listing (if desired).
   b. Press [SBR][0][2][2] to cancel left-hand justification (if desired).
8. If the first memory to be listed is in ...
   bank 4 (memories 0-29), press [R/S] at least once.
   bank 3 (memories 30-59), press [R/S] at least twice.
   bank 2 (memories 60-89), press [R/S] at least three times.
   bank 1 (memories 90-99), press [R/S] at least four times.
9. Insert the card on which the next bank of memories was recorded (if the memories to be listed extend into the next bank). The calculator will automatically read the card when needed, and continue listing.
10. Press [SBR][0][1][8] to list a new set of data (a -4 will be displayed). Continue as in #5-#9, above.
11. Normal Mode Operation: a. Read in sides one and two.
    b. Initialize with [SBR][0][CMS].
    c. Press [E] to print entire number in display.

---

* This initiates the fast mode (see TI PPC notes V5N6P4 on the fast mode limitations).

** For example, to list memories ten through nineteen, enter 10.19.

000	61	GTO	060	51	51	120	00	0	180	48	EXC	240	50	50	300	61	GTO	360	04	4	420	97	DSZ
001	00	00	061	76	LBL	121	61	GTO	181	59	59	241	08	8	301	04	04	361	04	4	421	55	55
002	25	25	062	15	E	122	04	04	182	95	=	242	44	SUM	302	50	50	362	42	STO	422	04	04
003	76	LBL	063	42	STO	123	50	50	183	42	STO	243	40	40	303	43	RCL	363	54	54	423	30	30
004	11	A	064	50	50	124	43	RCL	184	56	56	244	43	RCL	304	59	59	364	43	RCL	424	43	RCL
005	36	PGM	065	52	EE	125	50	50	185	87	IFF	245	50	50	305	50	I×I	365	53	53	425	57	57
006	02	02	066	32	X:T	126	67	EQ	186	04	04	246	61	GTO	306	55	÷	366	55	÷	426	29	CP
007	71	SBR	067	00	0	127	02	02	187	02	C2	247	04	04	307	01	1	367	03	3	427	67	EQ
008	02	02	068	42	STO	128	11	11	188	55	55	248	11	11	308	07	7	368	07	7	428	40	IND
009	39	39	069	56	56	129	59	INT	189	01	1	249	00	0	309	44	SUM	369	08	8	429	40	40
010	09	9	070	22	INV	130	22	INV	190	07	7	250	42	STO	310	40	40	370	42	STO	430	43	RCL
011	00	0	071	86	STF	131	67	EQ	191	32	X:T	251	40	40	311	03	3	371	40	40	431	57	57
012	69	DP	072	01	01	132	01	01	192	43	RCL	252	61	GTO	312	42	STO	372	01	1	432	59	INT
013	00	00	073	22	INV	133	65	65	193	59	59	253	03	03	313	55	55	373	00	0	433	22	INV
014	22	INV	074	86	STF	134	43	RCL	194	50	I×I	254	23	23	314	01	1	374	54	)	434	44	SUM
015	58	FIX	075	04	04	135	50	50	195	22	INV	255	00	0	315	00	0	375	61	GTO	435	57	57
016	25	CLR	076	05	5	136	85	+	196	87	IFF	256	42	STO	316	95	=	376	04	04	436	85	+
017	91	R/S	077	42	STO	137	32	X:T	197	01	01	257	55	55	317	61	GTO	377	18	18	437	32	X:T
018	25	CLR	078	58	58	138	53	(	198	02	02	258	02	2	318	04	04	378	06	6	438	01	1
019	04	4	079	87	IFF	139	05	5	199	02	02	259	07	7	319	18	18	379	44	SUM	439	00	0
020	94	+/-	080	00	00	140	75	-	200	75	-	260	00	0	320	00	0	380	40	40	440	49	PRD
021	91	R/S	081	00	00	141	43	RCL	201	01	1	261	42	STO	321	42	STO	381	61	GTO	441	57	57
022	22	INV	082	98	98	142	41	41	202	85	+	262	40	40	322	40	40	382	03	03	442	06	6
023	86	STF	083	43	RCL	143	69	DP	203	43	RCL	263	43	RCL	323	87	IFF	383	27	27	443	77	GE
024	00	00	084	50	50	144	10	10	204	41	41	264	56	56	324	02	02	384	01	1	444	04	04
025	25	CLR	085	22	INV	145	54	)	205	69	DP	265	59	INT	325	03	03	385	44	SUM	445	48	48
026	91	R/S	086	67	EQ	146	94	+/-	206	10	10	266	50	I×I	326	55	55	386	51	51	446	02	2
027	99	PRT	087	00	00	147	22	INV	207	95	=	267	61	GTO	327	00	0	387	44	SUM	447	85	+
028	86	STF	088	98	98	148	28	LOG	208	77	GE	268	04	04	328	48	EXC	388	53	53	448	01	1
029	02	02	089	01	1	149	52	EE	209	02	02	269	18	18	329	41	41	389	97	DSZ	449	95	=
030	85	+	090	44	SUM	150	42	STO	210	55	55	270	08	8	330	82	HIR	390	48	48	450	97	DSZ
031	59	INT	091	58	58	151	46	46	211	02	2	271	44	SUM	331	05	05	391	00	00	451	52	52
032	42	STO	092	00	0	152	75	-	212	44	SUM	272	40	40	332	00	0	392	50	50	452	51	BST
033	51	51	093	42	STO	153	43	RCL	213	55	55	273	04	4	333	48	EXC	393	22	INV	453	65	×
034	42	STO	094	40	40	154	46	46	214	02	2	274	00	0	334	42	42	394	86	STF	454	43	RCL
035	53	53	095	61	GTO	155	95	=	215	02	2	275	61	GTO	335	82	HIR	395	02	02	455	52	52
036	94	+/-	096	03	03	156	67	EQ	216	08	8	276	04	04	336	06	06	396	98	ADV	456	22	INV
037	42	STO	097	23	23	157	01	01	217	42	STO	277	50	50	337	00	0	397	83	GO*	457	28	LOG
038	48	48	098	04	4	158	61	61	218	40	40	278	08	8	338	48	EXC	398	55	55	458	52	EE
039	95	=	099	01	1	159	86	STF	219	43	RCL	279	44	SUM	339	43	43	399	03	3	459	85	+
040	65	×	100	42	STO	160	04	04	220	56	56	280	40	40	340	82	HIR	400	00	0	460	01	1
041	01	1	101	54	54	161	43	RCL	221	87	IFF	281	43	RCL	341	07	07	401	22	INV	461	52	EE
042	00	0	102	00	0	162	50	50	222	01	01	282	56	56	342	25	CLR	402	44	SUM	462	01	1
043	00	0	103	42	STO	163	86	STF	223	02	02	283	61	GTO	343	48	EXC	403	51	51	463	02	2
044	95	=	104	57	57	164	01	01	224	28	28	284	04	04	344	44	44	404	04	4	464	95	=
045	44	SUM	105	09	9	165	50	I×I	225	61	GTO	285	11	11	345	82	HIR	405	94	+/-	465	74	SM*
046	48	48	106	42	STO	166	55	÷	226	04	04	286	29	CP	346	08	08	406	22	INV	466	54	54
047	01	1	107	52	52	167	53	(	227	18	18	287	01	1	347	43	RCL	407	96	WRT	467	97	DSZ
048	44	SUM	108	29	CP	168	52	EE	228	00	0	288	07	7	348	50	50	408	61	GTO	468	52	52
049	48	48	109	43	RCL	169	55	÷	229	42	STO	289	44	SUM	349	84	DP*	409	00	00	469	04	04
050	43	RCL	110	50	50	170	52	EE	230	55	55	290	40	40	350	58	58	410	50	50	470	20	20
051	51	51	111	77	GE	171	00	0	231	02	2	291	43	RCL	351	43	RCL	411	50	I×I	471	01	1
052	32	X:T	112	01	01	172	00	0	232	04	4	292	50	50	352	50	50	412	22	INV	472	44	SUM
053	02	2	113	24	24	173	54	)	233	01	1	293	50	I×I	353	83	GO*	413	59	INT	473	54	54
054	09	9	114	01	1	174	42	STO	234	42	STO	294	28	LOG	354	40	40	414	65	×	474	09	9
055	22	INV	115	02	2	175	59	59	235	40	40	295	77	GE	355	03	3	415	01	1	475	42	STO
056	77	GE	116	04	4	176	28	LOG	236	04	4	296	02	02	356	42	STO	416	00	0	476	52	52
057	03	03	117	42	STO	177	52	EE	237	00	0	297	99	99	357	55	55	417	95	=	477	61	GTO
058	99	99	118	40	40	178	44	SUM	238	61	GTO	298	02	2	358	42	STO	418	42	STO	478	04	**04**
059	73	RC*	119	02	2	179	55	55	239	04	04	299	00	**0**	359	**52**	52	419	**57**	57	479	80	**20**

13-DIGIT REGISTER LIST.- I have seen several of these types of listings that print the contents and the register number each on a separate line. But the following two programs print both on one and the same line. The price paid for this "neatness" is, of course, a slow-executing program.

The first program is by Clyde Durbin, Dallas TX. It lists registers 00 through 85 at 26 sec/line. The instructions are simple:

Enter the highest register number and press A. Enter the lowest register number and press R/S. The program halts automatically.

The second program is by Richard Snow, Vallejo CA. As opposed to Clyde's program with 209 steps, Richard's program has only 159 steps. But it is slower: 29 sec/line. It will, however, list more registers: 01 through 89. It could list also registers 90 through 99, but those registers, being located on bank 1, together with the program, would have to loaded by hand, after the program has been read in.

Register 01 is used by this program, but it may be listed the first time around. It has to be re-loaded, however, if a second listing would be required.

The instructions for this program are:

Enter lowest register number and press A. Default value is 01. The program has to be halted manually, otherwise it will stop at the current partitioning.

SAMPLE OF 13-DIGIT PRINT-OUT

```
 3335313744.000  07
 33001719.00000  08
 28.40000412313  09
-24207.00235020  10
 9691.000000000  11
-14341667.34167  12
 100000.0000000  13
-37079.00227310  14
 18492.00007938  15
-18775.00105790  16
 9691.000000000  17
 14948333.42083  18
 100000.0000000  19
-27890.00159418  20
 17684.00000000  21
-25831.00012913  22
-280480.1298510  23
-146190.0001516  24
-121426.0247160  25
-110000.0002000  26
-100000.0000000  27
-144120.0001801  28
-45770.00000000  29
```

```
000  25 CLR
001  97 DSZ
002  88  88
003  00  00
004  64  64
005  31 R/S
006  76 LBL
007  16 A'
008  85  +
009  32 X:T
010  01  1
011  00  0
012  00  0
013  82 HIR
014  47  47
015  97 DSZ
016  87  87
017  00  00
018  29  29
019  04  4
020  00  0
021  42 STO
022  87  87
023  82 HIR
024  37  37
025  00  0
026  61 GTO
027  00  00
028  48  48
029  08  8
030  77  GE
031  00  00
032  39  39
033  04  4
034  82 HIR
035  37  37
036  34 FX
037  94 +/-
038  85  +
039  01  1
040  75  -
041  59 INT
042  82 HIR
043  37  37
044  95  =
045  65  ×
046  01  1
047  00  0
048  55  ÷
049  97 DSZ
050  86  86
051  16 A'
052  82 HIR
053  04  04
054  82 HIR
055  17  17
056  32 X:T
057  05  5
058  48 EXC
059  86  86
060  82 HIR
061  07  07
062  32 X:T
063  92 RTN
064  69 DP
065  00  00
066  01  1
067  44 SUM
068  89  89
069  02  2
070  42 STO
071  86  86
072  03  3
073  42 STO
074  87  87
075  43 RCL
076  89  89
077  55  ÷
078  01  1
079  00  0
080  95  =
081  16 A'
082  69 DP
083  04  04
084  82 HIR
085  06  06
086  73 RC*
087  69  69
088  29 CP
089  67 EQ
090  02  02
091  04  04
092  77 GE
093  01  01
094  01  01
095  32 X:T
096  02  2
097  00  0
098  82 HIR
099  07  07
100  32 X:T
101  50 I×I
102  28 LOG
103  59 INT
104  42 STO
105  87  87
106  32 X:T
107  04  4
108  42 STO
109  86  86
110  02  2
111  44 SUM
112  87  87
113  01  1
114  02  2
115  22 INV
116  77 GE
117  52 EE
118  05  5
119  94 +/-
120  77 GE
121  52 EE
122  00  0
123  32 X:T
124  77 GE
125  01  01
126  35  35
127  01  1
128  42 STO
129  87  87
130  01  1
131  00  0
132  61 GTO
133  01  1
134  42  42
135  67  EQ
136  01  01
137  27  27
138  22 INV
139  28 LOG
140  88 DMS
141  35 1/X
142  65  ×
143  00  0
144  82 HIR
145  16  16
146  69 DP
147  04  04
148  73 RC*
149  89  89
150  50 I×I
151  95  =
152  32 X:T
153  01  1
154  00  0
155  32 X:T
156  22 INV
157  77 GE
158  01  01
159  66  66
160  55  ÷
161  01  1
162  44 SUM
163  37  37
164  32 X:T
165  35  =
166  16 A'
167  69 DP
168  01  01
169  82 HIR
170  14  14
171  16 A'
172  69 DP
173  02  02
174  82 HIR
175  14  14
176  16 A'
177  69 DP
178  03  03
179  69 DP
180  05  05
181  81 RST
182  76 LBL
183  11  A
184  75  -
185  09  9
186  69 DP
187  17  17
188  91 R/S
189  42 STO
190  89  89
191  85  +
192  02  2
193  95  =
194  42 STO
195  88  88
196  01  1
197  22 INV
198  44 SUM
199  89  89
200  81 RST
201  76 LBL
202  52 EE
203  52 EE
204  73 RC*
205  89  89
206  69 DP
207  06  06
208  81 RST
```

```
000  24 CE
001  91 R/S
002  76 LBL
003  16 A'
004  05  5
005  42 STO
006  01  01
007  76 LBL
008  18 C'
009  01  1
010  00  0
011  00  0
012  82 HIR
013  48  48
014  87 IFF
015  00  00
016  00  00
017  31  31
018  97 DSZ
019  00  00
020  00  00
021  36  36
022  86 STF
023  00  00
024  04  4
025  00  0
026  82 HIR
027  38  38
028  61 GTO
029  00  00
030  55  55
031  82 HIR
032  11  11
033  67  EQ
034  00  00
035  58  58
036  08  8
037  00  0
038  82 HIR
039  38  38
040  01  1
041  00  0
042  85  +
043  01  1
044  85  +
045  28 LOG
046  59 INT
047  65  ×
048  02  2
049  75  -
050  59 INT
051  82 HIR
052  38  38
053  54  )
054  65  ×
055  97 DSZ
056  01  01
057  18 C'
058  82 HIR
059  18  18
060  82 HIR
061  58  58
062  92 RTN
063  76 LBL
064  11  A
065  93  .
066  01  1
067  82 HIR
068  04  04
069  69 DP
070  00  00
071  29 CP
072  42 STO
073  00  00
074  25 CLR
075  73 RC*
076  00  00
077  69 DP
078  19  19
079  87 IFF
080  07  07
081  01  01
082  58  58
083  67  EQ
084  01  01
085  50  50
086  32 X:T
087  22 INV
088  86 STF
089  00  00
090  22 INV
091  77 GE
092  01  01
093  00  00
094  02  2
095  00  0
096  00  0
097  00  0
098  82 HIR
099  08  08
100  04  4
101  42 STO
102  01  01
103  01  1
104  42 STO
105  00  00
106  32 X:T
107  50 I×I
108  22 INV
109  77 GE
110  01  01
111  22  22
112  55  ÷
113  28 LOG
114  59 INT
115  44 SUM
116  00  00
117  22 INV
118  28 LOG
119  52 EE
120  22 INV
121  52 EE
122  29 CP
123  71 SBR
124  00  00
125  42  42
126  69 DP
127  01  01
128  16 A'
129  58 FIX
130  08  08
131  69 DP
132  02  02
133  16 A'
134  69 DP
135  03  03
136  22 INV
137  58 FIX
138  25 CLR
139  02  2
140  42 STO
141  01  01
142  82 HIR
143  14  14
144  55  ÷
145  18 C'
146  69 DP
147  04  04
148  69 DP
149  05  05
150  01  1
151  82 HIR
152  34  34
153  82 HIR
154  14  14
155  61 GTO
156  00  00
157  69  69
158  81 RST
```

STRUCTURAL PROGRAMS.- For those who are professionally interested in structural programs such as Simple Span Trusses, Steel Beam Columns and Reinforced Concrete Beams, the firm Engineering Calculations, Box 412, North Baldwin Station, Baldwin, NY 11510 has developed several of them and sells them at prices ranging from $ 10.00 to $ 35.00 per program. The programs come with complete listing (Very rare! Usually you get only cards recorded with minus!) sample problem, & recorded mag card(s). The programs are divided into four groups: A. Analysis and Design of Steel Structures, B. Analysis and Design of Reinforced Concrete Structures, C. Two Way Slab Systems 1977 A.C.I. Code, D. Foundations. Prices range from $ 110.00 to $ 155.00 per group of programs. Write to the above address if interested.

ANALYZE COMPLEX LINEAR NETWORKS WITH A BUILDING-BLOCK CALCULATOR PROGRAM.- This program was published in Electronic Design, April 26, 1980, page 191. S.H. Hartman, a TI PPC Club member, thinks it is a super program. I agree, if you find out which of the steps the printer goofed up. SH found them all. Again I agree, it works like a charm!
Here are the needed corrections:

Step	Should be
71	DIV
177	pi
340	square root
346	DIV
352	=
402	DIV
467	DIV

DAY OF THE WEEK.- Program 20 of the ML module computes the day of the week for you, but your are still required to interpret the output, a digit between 0 and 6 as meaning Saturday through Friday. Several people have sent me elaborate programs to have the machine actually print out the correct day, rather than a "dumb" digit. Among the many attempts the one from Evan Boden is remarkable for its briefness. It would be very well suited for 58 users. Just enter the date in the usual MMDD.YYYY format and press R/S. The output is one single abbreviation, such as TUE or FRI, printed in OP 02 sector. I could not resist enhancing it a little, giving it a user-defined key, printing date and day on one line and writing it such that program and data fit on one and the same card side. Of course, the price paid for all that is double the amount of program steps. For the enhanced program, enter the date in MMDD.YYYY format and press A. For both programs, make sure the ML module is plugged in.
By the way, the Europeans find it cumbersome to remember our "illogical" MMDD.YYYY format. "Logic" is defined as "state the date in ascending order of time duration." Thus,their format is DDMM.YYYY, for which I have seen several conversion programs, so that PGM 20 may be used.
The first program is Evan Boden's, with data in R10 through R16. The second one is the enhanced one by yours truly, with data in R90 through 96. Key in the program and data in 10 OP 17, then record one card side in 6 OP 17.

```
000  36 PGM
001  20  20
002  14  D
003  85  +
004  01  1
005  00  0
006  95  =
007  42 STO
008  18  18
009  73 RC+
010  18  18
011  69 OP
012  02  02
013  69 OP
014  05  05
015  91 R/S
016  81 RST
```

```
361337.      10
364131.      11
303231.      12
374117.      13
431716.      14
37234135.    15
213524.      16

TUE
SUN
TUE
FRI
SAT
```

```
000  76 LBL
001  11  A
002  42 STO
003  19  19
004  01  1
005  00  0
006  69 OP
007  17  17
008  43 RCL
009  19  19
010  36 PGM
011  20  20
012  14  D
013  85  +
014  09  9
015  00  0
016  95  =
017  42 STO
```

```
018  18  18
019  73 RC+
020  18  18
021  69 OP
022  04  04
023  43 RCL
024  19  19
025  58 FIX
026  04  04
027  69 OP
028  06  06
029  22 INV
030  58 FIX
031  06  6
032  69 OP
033  17  17
034  25 CLR
035  91 R/S
```

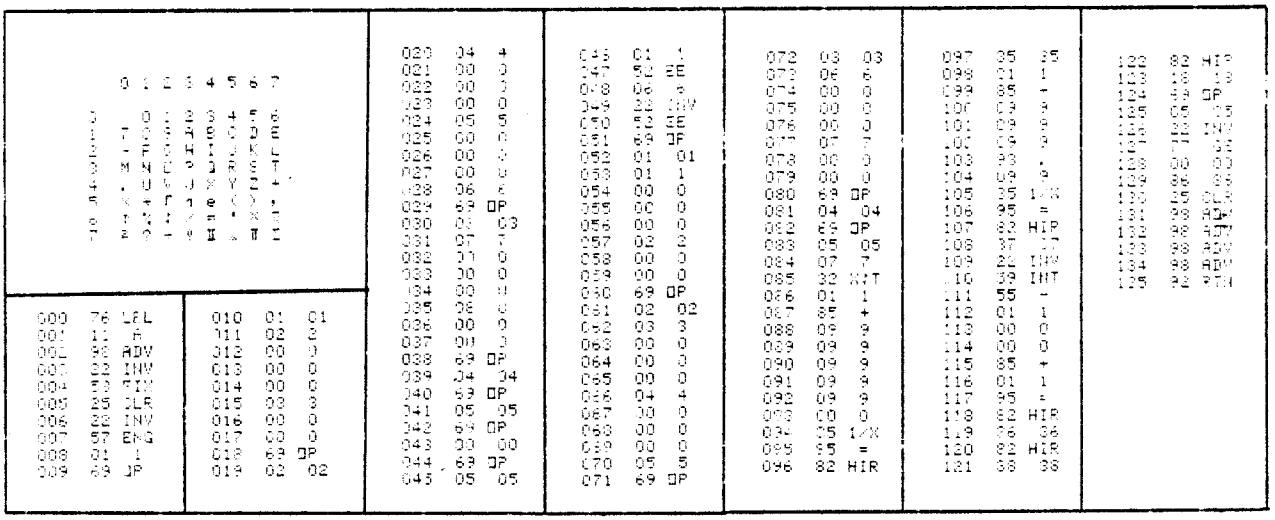
```
361337.      90
364131.      91
303231.      92
374117.      93
431716.      94
37234135.    95
213524.      96

1014.1980    TUE
1019.1980    SUN
1021.1980    TUE
1121.1980    FRI
1122.1980    SAT
```

PROGRAM LISTING WITHOUT PAPER.-(Re-v5n7p9) John Garza III, of Texas City, TX, has an explanation as to how this might work: "This is a classic example of 'It's so simple, why didn't I think of it?' I am sure you are familiar with the use of the GTO key during program execution as stated in Personal Programming. Now think of what OP 08 does. It prints a list of used labels. But first it must scan program memory to find each 76 key code. I believe what you see is this 'scanning' of program memory at a slower than normal rate."

ML-09, SIMPSON'S APPROXIMATION.- John Garza III tells me about an apparent trouble he experienced with repeated use of that program. When re-run according to the instructions the second run gave wrong answers. John found that pressing RST before re-running it cures it. Accordingly, step 12 of the instructions should now read: "For a new interval or new n, key RST, then repeat steps 7 through 11." See page 29 of the ML manual.
Has anybody else had the same troubles? Why does it happen?

PRINT CODE TABLES.- I was unaware of PPX's contest to print the fastest print code table until I got a call from from Bill Beebe in Lilburn, Georgia. As I saw later in the PPX newsletter, Bill received an honorable mention with a super speedy program. Unfortunately, the PPX people do not encourage the use of the fast mode. I don't know why, but I surmize it has not been proven yet beyond a shadow of a doubt that this mode is not detremental to the calculator. I have not had any bad experience with it, nor have I heard any complaints from other TI-59 users about their beloved calculator suddenly going on the blink because of the fast mode. But the caution of TI is proverbial.
Here are a few more programs to print a print code table, all respectably fast. The authors are Bill Beebe, Bill Skillman and Karl-Joseph Meusch.
The first three are rather simple to use: just enter the program, either from the keyboard or from a mag card. Each fits on one card side. Then press A.
The last one is a little more complex. It uses the fast mode and in this mode really goes to town. To use it, put it first on one side of a mag card. Then enter the card, press RST R/S, which initializes the fast mode. Then enter the card again and see the printing start. To show how slow normal mode is in comparison, press RST, which takes the program out of fast mode. Then press A. The same table is printed, but ever so slowly. Note also that this program prints the extended print code table. The author of this jewel is K-J Meusch, Koenigsdorf, West Germany.



(over)

PRINT CODE TABLES. (continued)

```
   0 1 2 3 4 5 6 7

0    0 1 2 3 4 5 6
1  7 8 9 A B C D E
2  - F G H I J K L
3  M N O P Q R S T
4  . U V W X Y Z +
5  × * √ π e ( ) ,
6  ↑ % : / = ' × x̄
7  ² ? ÷ ! Π ∠ π Σ
```

```
2000300.         90
400050006.       91
7000300.         92
1300200.         93
300040005.       94
6000700.         95
1.00000100001    96
1.00100010001    97
1.000010001      98
```

```
000  76 LBL      018  02  02      036  22 INV      054  07  7       072  69 OP
001  11  A       019  43 RCL      037  52 EE       055  42 STO      073  05  05
002  98 ADV      020  91  91      038  69 OP       056  99  99      074  97 DSZ
003  22 INV      021  69 OP       039  01  01      057  52 EE       075  99  99
004  58 FIX      022  03  03      040  43 RCL      058  43 RCL      076  00  00
005  25 CLR      023  43 RCL      041  93  93      059  96  96      077  58  58
006  22 INV      024  92  92      042  69 OP       060  82 HIR      078  25 CLR
007  57 ENG      025  69 OP       043  02  02      061  35  35      079  06  6
008  01  1       026  04  04      044  43 RCL      062  43 RCL      080  69 OP
009  00  0       027  69 OP       045  94  94      063  97  97      081  17  17
010  69 OP       028  05  05      046  69 OP       064  82 HIR      082  25 CLR
011  17  17      029  69 OP       047  03  03      065  37  37      083  98 ADV
012  01  1       030  00  00      048  43 RCL      066  43 RCL      084  98 ADV
013  69 OP       031  69 OP       049  95  95      067  98  99      085  98 ADV
014  01  01      032  05  05      050  69 OP       068  82 HIR      086  98 ADV
015  43 RCL      033  01  1       051  04  04      069  36  36      087  92 RTN
016  90  90      034  52 EE       052  69 OP       070  82 HIR
017  69 OP       035  06  6       053  05  05      071  38  38
```

```
  | 1 2 3 4 5 6 7
--+---------------
0 |   0 1 2 3 4 5 6
1 | 7 8 9 A B C D E
2 | - F G H I J K L
3 | M N O P Q R S T
4 | . U V W X Y Z +
5 | × * √ π e ( ) ,
6 | ↑ % : / = ' × x̄
7 | ² ? ÷ ! Π ∠ π Σ
```

```
1 EE 12                      STO 90
1 EE 12 + 2000300          = STO 91
1 EE 10 + 4000500.06       = STO 92
1 EE 12 + 7000300          = STO 93
1000100.01                   STO 94
1000010                      STO 96
1 EE 12 + 1 EE 6           = STO 97
```

```
000  76 LBL      015  06  06      030  43 RCL      045  69 OP       060  05  05
001  11  A       016  43 RCL      031  94  94      046  05  05      061  97 DSZ
002  25 CLR      017  92  92      032  82 HIR      047  43 RCL      062  00  00
003  98 ADV      018  82 HIR      033  56  56      048  96  96      063  00  00
004  01  1       019  07  07      034  82 HIR      049  82 HIR      064  47  47
005  00  0       020  43 RCL      035  58  58      050  35  35      065  93  .
006  69 OP       021  93  93      036  82 HIR      051  43 RCL      066  01  1
007  17  17      022  82 HIR      037  57  57      052  94  94      067  49 PRD
008  43 RCL      023  08  08      038  01  1       053  82 HIR      068  94  94
009  90  90      024  69 OP       039  00  0       054  36  36      069  06  6
010  82 HIR      025  05  05      040  49 PRD      055  82 HIR      070  69 OP
011  05  05      026  43 RCL      041  94  94      056  38  38      071  17  17*
012  43 RCL      027  97  97      042  07  7       057  82 HIR      072  98 ADV
013  91  91      028  82 HIR      043  42 STO      056  37  37      073  92 RTN
014  82 HIR      029  05  05      044  00  00      059  69 OP
```

```
        1.

   0123456789

0   012345678
1  789ABCDE78
2  -FGHIJKLMN
3  MNOPQRSTMN
4  .UVWXYZ+××
5  ×*√πe(),××
6  ↑%:/='×x̄²?
7  ²?÷!Π∠πΣ²?
8   012345678
9  789ABCDE78
```

```
000  00  0       025  98 ADV      050  03  03      075  05  5       099  06  6       123  01  01
001  00  0       026  69 OP       051  01  1       076  00  0       100  42 STO      124  69 OP
002  00  0       027  00  00      052  00  0       077  06  6       101  04  04      125  05  05
003  00  0       028  93  .       053  02  2       078  00  0       102  32 X:T      126  43 RCL
004  00  0       029  01  1       054  00  0       079  07  7       103  61 GTO      127  00  00
005  36 PGM      030  00  0       055  03  3       080  00  0       104  01  01      128  82 HIR
006  02  02      031  01  1       056  00  0       081  08  8       105  17  17      129  36  36
007  71 SBR      032  00  0       057  04  4       082  00  0       106  97 DSZ      130  82 HIR
008  02  02      033  01  1       058  00  0       083  09  9       107  01  01      131  37  37
009  39  39      034  00  0       059  05  5       084  98 ADV      108  00  00      132  00  0
010  09  9       035  01  1       060  69 OP       085  69 OP       109  90  90      133  83 GO*
011  00  0       036  00  0       061  02  02      086  03  03      110  01  1       134  04  04
012  76 LBL      037  01  1       062  69 OP       087  09  9       111  03  3       135  52 EE
013  11  A       038  42 STO      063  05  05      088  42 STO      112  05  5       136  22 INV
014  69 OP       039  00  00      064  65  ×       089  01  01      113  42 STO      137  52 EE
015  00  00      040  06  6       065  93  .       090  01  1       114  04  04      138  91 R/S
016  22 INV      041  00  0       066  00  0       091  00  0       115  01  1       139  00  0
017  52 EE       042  07  7       067  01  1       092  75  -       116  02  2       140  42 STO
018  22 INV      043  00  0       068  49 PRD      093  43 RCL      117  65  ×       141  04  04
019  57 ENG      044  08  8       069  00  00      094  01  01      118  04  4       142  69 OP
020  22 INV      045  00  0       070  95  =       095  95  =       119  22 INV      143  00  00
021  58 FIX      046  09  9       071  69 OP       096  32 X:T      120  28 LOG      144  61 GTO
022  01  1       047  01  1       072  02  02      097  01  1       121  95  =       145  00  00
023  99 PRT      048  02  2       073  69 OP       098  00  0       122  69 OP       146  25  25
024  25 CLR      049  69 OP       074  20  20
```

PROGRAMMING PUZZLES.- (re-v5n7p5) I received several solutions to selected puzzles. ---------- As you will remember, these puzzles were submitted by George Vogel, Newton Highlands, Mass. The first one went as follows:

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?

George had the obvious solution, which very few members saw:

Making a subroutine from the keyboard erases all subroutine addresses that may be stored in the calculator.(this includes pressing a user-defined key) There is a way out, however: instead of the subroutine command (e.g. A) make a GTO command and press R/S. This does not erase return addresses.

Richard Snow saw through the problem and came up with the correct solution. He offers this comment:

Pressing GTO A or GTO B followed by R/S solves the problem but puts an extra burden on the program user. The following method uses the decimal point trick:

.....R/S . OP 10 X 2 + 1 = RTN

The user can merely press R/S for a type "B" decision or he can enter any number greater than zero and press R/S for a type "A" decision.

J. Huntington Lewis offers as a solution this 42-step routine:

LBL E 2 + D + 100 = INV STF 1 R/S LBL D 2) IFF 1 024 = R/S GTO 005
LBL A + 1) LBL B + 1 = STF 1 GTO 017

in which LBL E is the main program, LBL D up to GTO 005 is the option SBR and LBLs A and B are the options. The RTNs do not equate to the depth of the subroutine in use.

Morton P. Matthew has this 34-step routine:

LBL C X^2 X:T 6 E PRT R/S LBL D 1/X 15 E PRT R/S LBL E STO 59 X:T X
R/S LBL A 2 + LBL B 1 = GTO IND 59

The best I can do is to replace the regular SBR with one that isn't fussy about labels. In the example both LBLs C and D call for the "subroutine" and A or B makes the decision. Register 59 is for subroutine return.

Jeff Rosedale's laconic comment:

Just call A and B by GTO A or B R/S.

---

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?

George says:

Enter each print code into the program as a ten-digit number, starting it 10... (or any two digits of which the first one is not a zero) and supply two more zeros for each blank position, if any. E.g.: 1000131415 will print ABC regardless of FIX.

Both Jeff Rosedale and Richard Snow came up with the same solution:

A ten-digit integer used as alpha code is not affected by the FIX mode. Simply add 1,000,000,000 to the alpha code used for OP 06 printing.

---

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:
$x \leq 60$, $60 \leq x < 65$, $65 \leq x < 70$, etc. Program should have 15 steps or less.

George offers this solution:

DIV 5 = INT - 11 = STO 59 1 SUM IND 59 R/S *This takes advantage of the fact that a negative address is interpreted by the calculator as zero.*

At this point I (the editor) have to beat my chest and say MEA CULPA, MEA MAXIMA CULPA. The first inequality was copied wrong. It should have been given as : $x<60$ and NOT as $x \leq 60$. In spite of that handicap both Richard Snow and Charles Williamson devised the correct solution. Richard even gave a solution in case it was not a typo:

DIV 5 - 11 = STO 09 X:T 1 SUM IND 09 X=T 002 RTN *which is a 16-step solution. When 60 is input, 1 must be added to two different registers since 60 satisfies the first two arguments given in the problem.*

(continues next page)

PROGRAMMING PUZZLES.(continued)
Karl-Joseph Meusch also sent a similar solution to puzzle 3:
DIV 5 - 11 = STO 59 1 SUM IND 59 R/S RST

---

4. Write a program of 14 steps or less that will store from cold start (CMS,RST, 0 x:t CLR, R/S) say .7, 1.7, 2.7, 3.7, ...8.7 in registers 0 through 8.

George Vogel's solution is as follows:
8.7 STO 09 RCL 09 STO IND 09 DSZ 9 005 R/S *This takes advantage of the fact that in extracting the direct address from reg 09 (or any other register) the calculator ignores the fractional part, and suggests a generally useful trick of including reg.00 in a DSZ procedure, e.g. clearing selectively the contents of registers from, say 8 down to 0 (replace RCL 09 by CLR in the above). Register 00 will be included if you use as the initial indirect address not 8, but 8 plus some fraction.*
Most participants didn't have too much trouble with this one. J. Huntington Lewis sends this one: 8.7 STO 09 STO IND 09 - 1) DSZ 9 005 R/S
Karl-Joseph Meusch writes: 8.7 STO 00 STO IND 00 - 1 = GE 003 R/S
And Richard Snow devised George Vogel's routine to a T; so did Charles Williamson.
And finally Morton P. Matthew has one with a PAUSE included, so you will know when to stop: .7 SUM 59 RCL 59 STO IND 59 PAU 1 GTO 002

---

5. Try this on the keyboard : 9.999 EE +/- 87 STO 01 STO 02 1 EE +/- 99 SUM 02 .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 ?

George Vogel offers this comment:
You have synthesized and stored in registers 01 and 02 two numbers which are still legal but differ by less than 1E-99. The comparison is done internally by subtraction which thus yields an illegal number. (This cannot happen unless two numbers are numerically smaller than 1E-87.) Why must CE (or CLR) be pressed twice? The first push is interpreted only as the address for the X=T (or EQ if you frefer) command, and only the second push will stop the flashing. (GE is subject to the same rules as EQ) It is interesting to note that, even when the condition described above causes flashing, the comparison and the corresponding branching are done correctly. It may seem an unlikely condition to occur, but it can happen and cause a big headache.
Richard Snow writes:
As derived from several articles on computers, it is evident that whenever a comparison is made, an internal subtraction is performed. The subtraction from the t-register results in an underflow error in this sample problem, The first CE does not clear the error because it completes the instruction " X=T CE" where CE is merely the name of the label. The second CE is treated as the normal error-clearing instruction.
And Jeff Rosedale? He gave up after puzzle 2. He simply writes:
3, 4, 5 ?
My thanks to the many contributors. It has been fun.

BRAIN TEASERS.- See also v5n7p5 and v5n8p4. A few more solutions came in:
Björn Gustavsson in Sweden has these two:
For .1415... : INV Σ+ RAD INV COS INV INT (that's the beginning of Don O'grady's routine too)
For 197: INV LOG INV LOG X^2 + CE - LOG) INT

Myer Boland on the other hand goes like this:
For .1415..: Σ+ +/- INV COS - X:T - Σ+ = (8 steps)
For 197: Σ+ Σ+ X INV LOG - Σ+ = (7 steps)

MASTER LIBRARY SURVIVAL GUIDE.- Fred Fish who wrote this fantastic manual in 1978, has a new address:
1346 W. 10th Place, Tempe. Arizona, USA. Tel: (602) 894-6881.
Fred still has a few copies available.

FAST MODE MORSE CODE.- Dave Leising.

I couldn't believe my eyes when I first received this program. Morse code on the TI-59/PC100 ? I had seen of course many morse code translators: silly little programs that require the user to enter each dit into A, each dah into B and receive the translation through C. I am an old radio amateur and, although I am woefully out of practice, ! am stll able to recognize any morse code character much faster by ear than through that cumbersome process.

No, this program was entirely different. This one is a real morse "sounder." It will translate into morse code,and sound it,any text you have stored in the still available data registers. The example that Dave gives, stored in registers 00 through 43, is part of a poem by Omar Khayam:

The moving finger writes,
And having writ, moves on.
Nor all your piety nor wit
Can lure it back to cancel half a line.
Nor all your tears wash out a word of it.

INSTRUCTIONS FOR FAST MODE MORSE CODE PROGRAM

1. LOAD PRINT CODE INTO ANY DATA REGISTERS DESIRED FROM 00 THRU 47 INCLUSIVE. SEQUENCE OF DATA OUTPUT WILL BE FROM LEFT TO RIGHT, COMMENCING AT REGISTER 00 AND TERMINATING AT REGISTER 47. ANY CHARACTERS OR EXTENDED PRINT CODE CAN BE LOADED; HOWEVER, THE PROGRAM WILL SOUND ONLY THE CODES FOR LETTERS A THRU Z, OTHER CHARACTERS WILL BE IGNORED. RECORD THIS DATA (BANKS 3 AND 4) UNDER NORMAL MACHINE POWER-UP STATUS ON A MAGNETIC CARD.
2. LOAD SIDE ONE OF THE PROGRAM CARD, THEN PRESS RST R/S. YOU ARE NOW IN THE FAST MODE.
3. PASS, IN SEQUENCE, SIDES 1 AND 2 OF THE PROGRAM CARD, THEN PASS SIDES 3 AND 4 OF THE DATA CARD.
4. THE PRINTER WILL NOW SEQUENTIALLY SOUND IN MORSE CODE ALL THE ALPHABETIC CHARACTERS IN REGISTERS 00 THRU 47 INCLUSIVE. THE PRINTING OF A ROW OF TWENTY DASHES PRODUCES A STACCATO "POP" SIGNIFYING A "DIT", AND THE PRINTING OF A LINE OF TEN EIGHTS PRODUCTS A LOW "DRUM ROLL" SIGNIFYING A "DAH."
5. AT THE COMPLETION OF THE PROCESS:
   A. PRESS R/S TWICE IN SEQUENCE TO HEAR THE SAME MESSAGE AGAIN.
   B. PASS TWO SIDES OF ANOTHER DATA CARD TO HEAR A DIFFERENT MESSAGE.
   C. ENTER A TWO DIGIT ALPHA (LETTER) CHARACTER AND PRESS SBR 175 TO HEAR THE INDIVIDUAL CHARACTER SOUNDED.
6. THERE IS A VARIATION IN THE EFFECTIVENESS OF THE SOUND WITH DIFFERENT SERIES OF PRINTERS AND DIFFERENT RUNS OF PAPER. THE EARLIER PC-100A PRINTERS USING THE ORIGINAL THIN PAPER DO THE BEST JOB.

Who said engineers have no class? You may replace the text by any other, more to your liking, of course. In any case, the morse code will sound at slightly over 6 words per minute. Radio amateurs will tell you that it does not yet constitute "lightning speed", but it is a good beginning. Dave wrote this program, some time ago, for PPX-52, but in normal mode , at about 3 WPM it is too slow. Now, thanks to fast mode we are getting there.

The program is fairly straightforward, except for the interleaving of control program steps from step 175 through 208. Dave ran out of uncontended program memory at step 130, where the vector table starts. So he slipped the code for sounding the single entered characters in between the dummy returns from extended print code 18 and 19 and the unaccepted hyphen 20.Another non-obvious sequence occurs in the table for the code for H (dit dit dit dit) and I (dit dit) at steps 230 through 246. This merging was necessary as four OP 05's and the GTO 082 would have pushed the table code for H out of bounds into the entry locus for I at 240. Dave says that "it was pure serendipity that the character codes and the morse codes for these letters allowed this merging." I agree, you sometimes say to yourselves "I must live right to deserve this kind of luck."

You would of course like to know what the usefulness of all this is. "As is" not too much, but it might form the basis for a new kind of contest: because we miss the very useful beeper our competitors have on the HP-41C, could anybody device an external (hardware) beeper for the TI-59 ? It should not involve any modification of the calculator. Output signals may be sensed at the connector between the TI-59 and the printer. Maybe a particular sequence might be sensed, such as a NOP, which is not used very often. The number of NOPs will then determine the duration of the beep. Each NOP might, for example, give a 100 milliseconds long beep. It would be a very practical addition to our machine and it should not be all that expensive to build it, nor too complicated to be beyond the capabilities of the average Heathkit builder.

FAST MODE MORSE CODE. (continued) Program listing.

```
3723170030.      00
3242243122.      01
  21243122.      02
1735000000.      03

      4035.      06
2437173657.      07
1331160023.      08
1342243122.      09
  43352437.      10
5700000000.      11

    303242.      14
1736003231.      15
3132350013.      16
2727004532.      17
4135003324.      18
1737450000.      19

      3132.      22
3500432437.      23
1513310027.      24
4135170024.      25
3700141315.      26
2600373200.      27
    151331.      28
1517270023.      29
1327210013.      30
  27243117.      31
3132350013.      32
2727004532.      33
4135003717.      34
1335360000.      35
  43133623.      36
  32413700.      37
1300433235.      38
1600000000.      39

3221002437.      43
```

000	00	0	038	08	8	076	00	0	157	00	00	238	69	DP	319	00	0				
001	00	0	039	42	STO	077	22	INV	158	82	82	239	05	05	320	99	PRT	400	61	GTO
002	00	0	040	54	54	078	33	LOG	159	00	0	240	69	DP	321	99	PRT	401	00	00
003	00	0	041	02	2	079	95	=	160	99	PRT	241	05	05	322	99	PRT	402	82	82
004	00	0	042	00	0	080	83	HIR	161	69	DP	242	69	DP	323	61	GTO	403	00	0
005	36	PGM	043	02	2	081	04	04	162	05	05	243	05	05	324	00	00	404	00	0
006	02	02	044	00	0	082	01	1	163	69	DP	244	61	GTO	325	82	82	405	00	0
007	71	SBR	045	02	2	083	00	0	164	05	05	245	00	00	326	00	0	406	00	0
008	02	02	046	00	0	084	00	0	165	61	GTO	246	82	82	327	00	0	407	00	0
009	39	39	047	02	2	085	82	HIR	166	00	00	247	00	0	328	00	0	408	00	0
010	09	9	048	00	0	086	44	44	167	82	82	248	00	0	329	00	0	409	00	0
011	00	0	049	02	2	087	82	HIR	168	00	0	249	00	0	330	69	DP	410	69	DP
012	76	LBL	050	00	0	088	14	14	169	00	0	250	69	DP	331	05	05	411	05	05
013	11	A	051	69	DP	089	26	CP	170	69	DP	251	05	05	332	99	PRT	412	69	DP
014	22	INV	052	01	01	090	67	EQ	171	05	05	252	99	PRT	333	99	PRT	413	05	05
015	58	FIX	053	69	DP	091	01	01	172	61	GTO	253	99	PRT	334	69	DP	414	99	PPT
016	22	INV	054	02	02	092	22	22	173	00	00	254	99	PRT	335	05	05	415	61	GTO
017	57	ENG	055	69	DP	093	54	INT	174	82	82	255	61	GTO	336	61	GTO	416	00	00
018	01	1	056	03	03	094	82	HIR	175	59	INT	256	00	00	337	00	00	417	82	82
019	99	PRT	057	69	DP	095	54	54	176	55	-	257	82	82	338	82	82	418	00	0
020	25	CLR	058	04	04	096	32	X:T	177	61	GTO	258	00	0	339	00	0	419	00	0
021	91	R/S	059	04	4	097	67	EQ	178	01	01	259	00	0	340	99	PRT	420	69	DP
022	99	PRT	060	08	8	098	00	00	179	83	83	260	99	PRT	341	99	PRT	421	05	05
023	25	CLR	061	42	STO	099	82	82	180	61	GTO	261	69	DP	342	69	DP	422	69	DP
024	91	R/S	062	50	50	100	01	1	181	00	00	262	05	05	343	05	05	423	05	05
025	99	PRT	063	43	RCL	101	02	2	182	82	82	263	99	PRT	344	99	PRT	424	69	DP
026	25	CLR	064	50	50	102	77	GE	183	01	1	264	61	GTO	345	61	GTO	425	05	05
027	91	R/S	065	75	-	103	00	00	184	00	0	265	00	00	346	00	00	426	99	PRT
028	99	PRT	066	04	4	104	82	82	185	00	0	266	82	82	347	82	82	427	61	GTO
029	08	8	067	08	8	105	04	4	186	95	=	267	00	0	348	00	0	428	00	00
030	08	8	068	95	=	106	07	7	187	61	GTO	268	00	0	349	00	0	429	82	82
031	08	8	069	94	+/-	107	22	INV	188	01	01	269	00	0	350	69	DP	430	69	DP
032	08	8	070	42	STO	108	77	GE	189	83	83	270	69	DP	351	05	05	431	05	05
033	08	8	071	51	51	109	00	00	190	61	GTO	271	05	05	352	99	PRT	432	99	PRT
034	08	8	072	73	RC*	110	82	82	191	00	00	272	99	PRT	353	69	DP	433	99	PRT
035	08	8	073	51	51	111	32	X:T	192	82	82	273	69	DP	354	05	05	434	61	GTO
036	08	8	074	55	÷	112	42	STO	193	22	INV	274	05	05	355	61	GTO	435	00	00
037	08	8	075	01	1	113	52	52	194	59	INT	275	69	DP	356	00	00	436	82	82
						114	01	1	195	82	HIR	276	05	05	357	82	82	437	00	0
						115	00	0	196	04	04	277	61	GTO	358	00	0	438	00	0
						116	49	PRD	197	61	GTO	278	00	00	359	00	0	439	00	0
						117	52	52	198	02	02	279	82	82	360	69	DP	440	99	PRT
						118	43	RCL	199	03	03	280	61	GTO	361	05	05	441	69	DP
						119	54	54	200	61	GTO	281	00	00	362	69	DP	442	05	05
						120	83	GD*	201	00	00	282	82	82	363	05	05	443	69	DP
						121	52	52	202	82	82	283	00	0	364	69	DP	444	05	05
						122	97	DSZ	203	25	CLR	284	00	0	365	05	05	445	99	PRT
						123	50	50	204	42	STO	285	00	0	366	61	GTO	446	61	GTO
						124	00	00	205	50	50	286	00	0	367	00	00	447	00	00
						125	63	63	206	61	GTO	287	00	0	368	82	82	448	82	82
						126	61	GTO	207	00	00	288	00	0	369	00	0	449	00	0
						127	00	00	208	82	82	289	00	0	370	99	PRT	450	99	PRT
						128	23	23	209	00	0	290	61	GTO	371	61	GTO	451	69	DP
						129	00	0	210	69	DP	291	00	00	372	00	00	452	05	05
						130	69	DP	211	05	05	292	82	82	373	82	82	453	99	PRT
						131	05	05	212	69	DP	293	00	0	374	00	0	454	99	PRT
						132	99	PRT	213	05	05	294	00	0	375	00	0	455	61	GTO
						133	61	GTO	214	99	PRT	295	00	0	376	00	0	456	00	00
						134	00	00	215	69	DP	296	00	0	377	00	0	457	82	82
						135	82	82	216	05	05	297	00	0	378	00	0	458	00	0
						136	00	0	217	61	GTO	298	00	0	379	00	0	459	00	0
						137	00	0	218	00	00	299	00	0	380	61	GTO	460	99	PRT
						138	00	0	219	82	82	300	99	PRT	381	00	00	461	99	PRT
						139	00	0	220	99	PRT	301	99	PRT	382	82	82	462	69	DP
						140	99	PRT	221	99	PRT	302	61	GTO	383	00	0	463	05	05
						141	69	DP	222	69	DP	303	00	00	384	00	0	464	69	DP
						142	05	05	223	05	05	304	82	82	385	00	0	465	05	05
						143	69	DP	224	61	GTO	305	00	0	386	00	0	466	61	GTO
						144	05	05	225	00	00	306	00	0	387	00	0	467	00	00
						145	69	DP	226	82	82	307	00	0	388	00	0	468	82	82
						146	05	05	227	00	0	308	00	0	389	00	0	469	00	0
						147	61	GTO	228	00	0	309	00	0	390	61	GTO	470	61	GTO
						148	00	00	229	00	0	310	99	PRT	391	00	00	471	00	00
						149	82	82	230	00	0	311	69	DP	392	82	82	472	82	82
						150	99	PRT	231	00	0	312	05	05	393	00	0	473	00	0
						151	69	DP	232	00	0	313	61	GTO	394	00	0	474	00	0
						152	05	05	233	00	0	314	00	00	395	00	0	475	00	0
						153	99	PRT	234	00	0	315	82	82	396	00	0	476	00	0
						154	69	DP	235	00	0	316	00	0	397	00	0	477	00	0
						155	05	05	236	69	DP	317	00	0	398	00	0	478	00	0
						156	61	GTO	237	05	05	318	00	0	399	00	0	479	00	0

ROUNDING ROUTINE.- Milton Cragg wrote this short and practical routine. It will round any positive number up to nine decimal places in length to any number of decimal places up to nine.

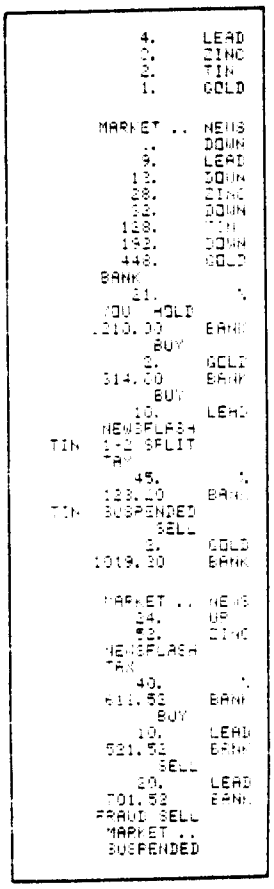
Enter the number and press A. Enter the number of decimal places desired and press R/S.

LBL A PRT X:T 10 Y^X R/S = X X:T + .5 = INT DIV X:T = PRT R/S

.....

MARKET , *a game program by Robert Snow.*

The game of MARKET is an enhancement of STOCK MARKET, which was first published in Great Britain and arrived at our offices via Germany.(K-J.Meusch) The original author is unknown to us. But his version had several objectionable parts in it, so that Robert Snow decided to enhance it. Tax and prices are now more realistic, many unnecessary print-outs have been eliminated, the verbage has been altered to be more meaningful, label A is used for Bank Account, rather than the cumbersome SBR SBR. The market runs now for nine years, instead of a random time as in the original version. And above all, the code has been greatly optimized.



At the onset you are provided with about $ 1000.00 which you are supposed to use for trading four commodities: Gold, tin, zinc and lead. Fluctuations are shown in the nine yearly Market News reports. In addition to price changes, bank interests on your account are paid annually. The following events can occur and are reported in the News Flashes:

1.TIN SUSPENDED: No buying or selling of tin allowed.
2.MARKET SUSPENDED: No buying or selling of any commodity permitted.
3.BANKRUPT: A stock goes broke and is removed from your portfolio.
4.TAX: A percentage of your bank account is taxed.
5.BANK CRASH: Your bank account is zero'd.
6.FRAUD: You attempted an illegal buy or sell and the market is temporarily closed.
7.MARKET CRASH: All stocks held by you are sold at par value and credited to your account. The game has ended.
8.DIVIDEND: A percentage of your bank account is credited to your account.
9.ZINC SUSPENDED: No buying or selling of zinc allowed.
10.TAX REBATE: A tax refund has been credited to your account.
11.1-2 SPLIT: One share of stock is earned for each two full shares held by you.
12.DIVIDEND SUSPENDED: At the last minute, a declared tax or dividend is cancelled.

(Some of these "events" seem rather fiendish)

PROCEDURE:

1. Initialize, press F.
2. To buy or sell, enter in the form X.Y, where X= number of stocks
Y= stock code.

The code for gold is 1
tin is 2
zinc is 3
lead is 4.

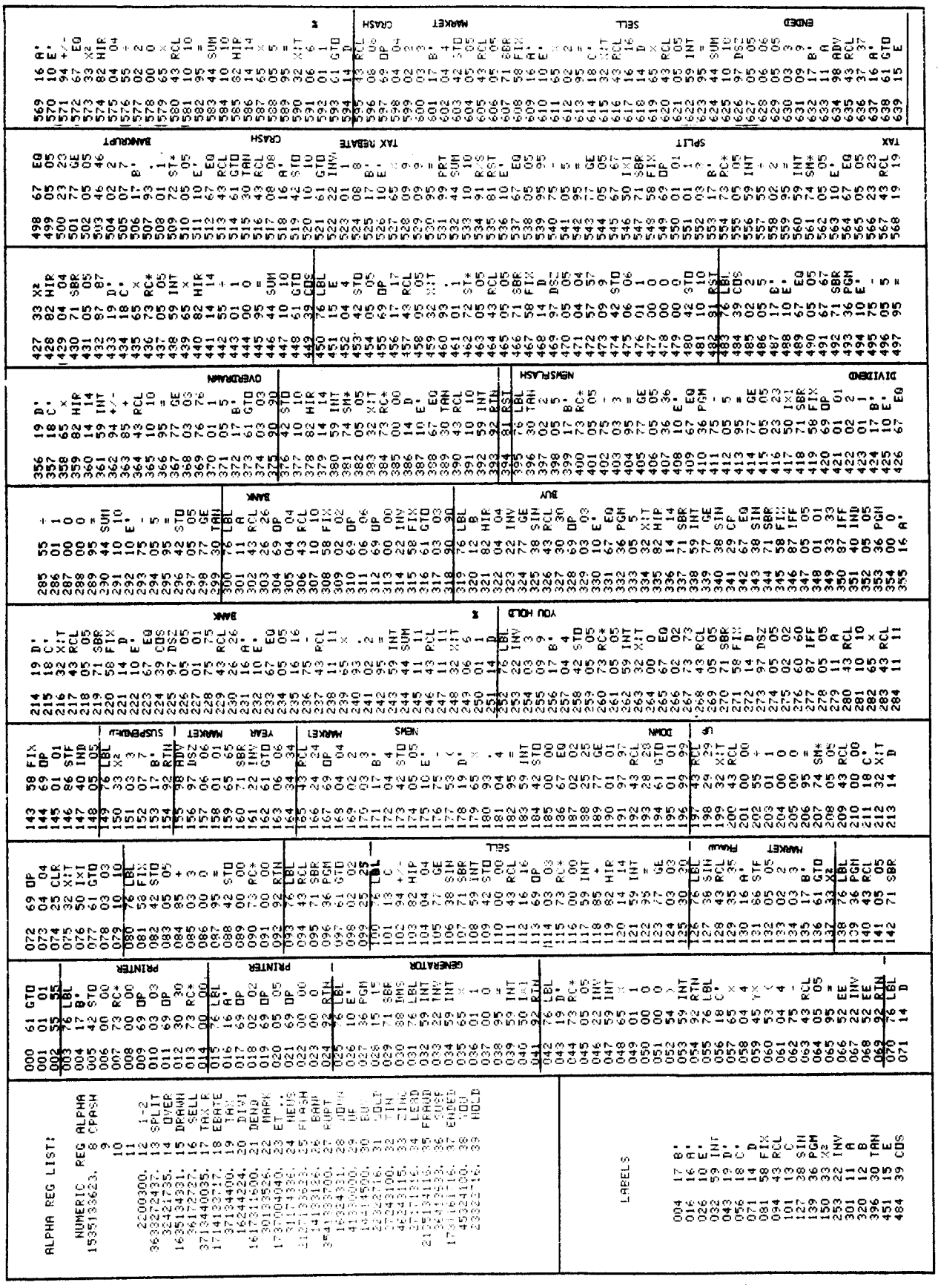
Thus, if you want to sell, say, four shares of gold, you enter as 4.1
To buy, press B. To sell, press C.
3. To obtain the current MARKET NEWS, at all times press R/S.
4. Your account is displayed after every buy or sell. To get a printed statement, press A.
5. Go to steps 2 or 3.

Note: OVERDRAWN on a buy transaction means that you do not have adequate funds to buy stock.
If you try to buy or sell a commodity that is suspended, the entire market is suspended, and the suspension will extend through the following year.

The above print-out is an example of what to expect, especially if you cheat, as I tried to do.
Program keyed-in in 4 OP 17, recorded in 6 OP 17.

SEE PROGRAM ON NEXT PAGE.

MARKET program by Robert Snow.



THE TWELVE DAYS OF CHRISTMAS.- Two years ago Jill Zimmerman wrote a program by that name in Kilobaud. If one has 16 K memory and Basic available, printing that famous Christmas carol is a cinch. But it is of course the idea that counts. Now, when the Snow brothers saw that program and said to themselves:" How about making the TI-59 do it?", that was a real challenge. So, just before Christmas 1979 I received a version by Richard Snow. Once it was suitably distributed among the members of our local club, lots of new versions cropped up. I even wrote one myself and sent it to PPX, not as a submission, but to amuse the my friends the program analysts.

The version presented here is the shortest and most program-optimized I have seen so far. It has only 119 steps. Who else but Bill Skillman could have written such a concise routine?

Entering the program steps is easy and fast. Just be careful with steps 097..... Enter as DSZ STO 89 GTO 049, then go back and delete the STO and the GTO. But entering all the print code data in registers 01 through 90 will take you a while. Key in the program and load the registers in 10 OP 17 partition. When done, go back to 6 OP 17 and record four card sides.

To run the program, load the four card sides and press A. About a yard of paper will be the result of all the printing."But it is cute," you'll say.

```
000  76 LBL
001  17 B'
002  73 RC*
003  00  00
004  69 OP
005  30  30
006  69 OP
007  04  04
008  73 RC*
009  00  00
010  69 OP
011  30  30
012  69 OP
013  03  03
014  73 RC*
015  00  00
016  69 OP
017  30  30
018  69 OP
019  02  02
020  73 RC*
021  00  00
022  69 OP
023  01  01
024  69 OP
025  05  05
026  92 RTN
027  76 LBL
028  11  A
029  25 CLR
030  01  1
031  00  0
032  69 OP
033  17  17
034  06  6
035  02  2
036  42 STO
037  00  00
038  42 STO
039  88  88
040  17 B'
041  69 OP
042  00  00
043  69 OP
044  05  05
045  01  1
046  02  2
047  42 STO
048  89  89
049  01  1
050  44 SUM
051  88  86
052  73 RC*
053  88  88
054  69 OP
055  00  00
056  69 OP
057  02  02
058  01  1
059  44 SUM
060  88  88
061  73 RC*
062  88  88
063  69 OP
064  03  03
065  43 RCL
066  87  87
067  69 OP
068  01  01
069  69 OP
070  05  05
071  05  5
072  06  6
073  42 STO
074  00  00
075  17 B'
076  69 OP
077  30  30
078  17 B'
079  04  4
080  95  +
081  42 STO
082  00  00
083  17 B'
084  97 DSZ
085  00  00
086  00  00
087  83  83
088  69 OP
089  00  00
090  43 RCL
091  57  57
092  69 OP
093  01  01
094  43 RCL
095  58  58
096  13  C
097  97 DSZ
098  89  89
099  00  00
100  49  49
101  25 CLR
102  43 RCL
103  90  90
104  69 OP
105  03  03
106  43 RCL
107  87  87
108  76 LBL
109  13  C
110  69 OP
111  02  02
112  69 OP
113  05  05
114  69 OP
115  00  00
116  69 OP
117  05  05
118  92 RTN
```

```
          0.  00
1300331335.  01
3735241622.  02
1700243100.  03
1300000000.  04
3743010037.  05
4135372717.  06
1601421736.  07
5700133116.  08
3723351717.  09
  21351731.  10
1523002317.  11
3136570000.  12
2101413500.  13
1513272724.  14
3122001424.  15
3516365700.  16
2124421700.  17
3201271600.  18
0524312236.  19
5700000000.  20
3624440022.  21
1717361700.  22
1320271345.  23
2431225700.  24
 800364313.  25
3136001320.  26
3643243030.  27
2431225700.  28
 900301324.  29
1636001320.  30
3024272624.  31
3122570000.  32
1200163541.  33
3030173536.  34
  16354130.  35
3024312257.  36
3717310033.  37
2433173536.  38
  33243324.  39
3122570000.  40
 202002713.  41
1624173600.  42
1613311524.  43
3122570000.  44
 203002701.  45
3516360013.  46
2027171333.  47
2431225700.  48
3735411700.  49
2701421700.  50
2213421700.  51
3701003017.  52
1613450001.  53
2100152335.  54
2436373013.  55
3600304500.  56
3317133500.  57
3735171740.  58
 203001613.  59
4536000121.  60
  15233524.  61
3637301336.  62
2124353637.  63
          0.  64
3617150131.  65
1600000000.  66
3723243516.  67
          0.  68
2101413537.  69
2300000000.  70
2124213723.  71
          0.  72
3624443723.  73
          0.  74
3617421731.  75
3723000000.  76
1724222337.  77
2300000000.  78
3124313723.  79
          0.  80
3717313723.  81
          0.  82
1727174217.  83
3137230000.  84
3743172721.  85
3723000000.  86
  37231700.  87
          0.  88
          0.  89
1731160000.  90
```

12 DAYS OF CHRISTMAS

THE FIRST
DAY OF CHRISTMAS MY
TRUE LOVE GAVE TO ME
A PARTRIDGE IN A
PEAR TREE.

THE SECOND
DAY OF CHRISTMAS MY
TRUE LOVE GAVE TO ME
TWO TURTLEDOVES, AND
A PARTRIDGE IN A
PEAR TREE.

THE THIRD
DAY OF CHRISTMAS MY
TRUE LOVE GAVE TO ME
THREE FRENCH HENS,
TWO TURTLEDOVES, AND
A PARTRIDGE IN A
PEAR TREE.

FIVE GOLD RINGS,
FOUR CALLING BIRDS,
THREE FRENCH HENS,
TWO TURTLEDOVES, AND
A PARTRIDGE IN A
PEAR TREE.

THE TENTH
DAY OF CHRISTMAS MY
TRUE LOVE GAVE TO ME
TEN PIPERS PIPING,
9 DRUMMERS DRUMMING,
8 MAIDS A-MILKING,
7 SWANS A-SWIMMING,
SIX GEESE A-LAYING,
FIVE GOLD RINGS,
FOUR CALLING BIRDS,
THREE FRENCH HENS,
TWO TURTLEDOVES, AND
A PARTRIDGE IN A
PEAR TREE.

THE ELEVENTH
DAY OF CHRISTMAS MY
TRUE LOVE GAVE TO ME
11 LADIES DANCING,
TEN PIPERS PIPING,
9 DRUMMERS DRUMMING,
8 MAIDS A-MILKING,
7 SWANS A-SWIMMING,
SIX GEESE A-LAYING,
FIVE GOLD RINGS,
FOUR CALLING BIRDS,
THREE FRENCH HENS,
TWO TURTLEDOVES, AND
A PARTRIDGE IN A
PEAR TREE.

THE TWELFTH
DAY OF CHRISTMAS MY
TRUE LOVE GAVE TO ME
12 LORDS A-LEAPING,
11 LADIES DANCING,
TEN PIPERS PIPING,
9 DRUMMERS DRUMMING,
8 MAIDS A-MILKING,
7 SWANS A-SWIMMING,
SIX GEESE A-LAYING,
FIVE GOLD RINGS,
FOUR CALLING BIRDS,
THREE FRENCH HENS,
TWO TURTLEDOVES, AND
A PARTRIDGE IN A
PEAR TREE.

THE END

See you next year, Maurice

9213 Lanham Severn Road, Lanham MD 20801

TI PPC NOTES, 1981 SUBSCRIPTION FORM. *Please copy this form and mail back.*

Distribute a few copies among your friends.

I WANT A 1981 SUBSCRIPTION ONLY []

Please check appropriate box(es)

US bulk rate mail	*@ 20.00*	[]
US First class	*@ 25.00*	[]
Overseas air mail	*@ 30.00*	[]
Overseas surface mail	*@ 25.00*	[]
Canada & Mexico	*@ 25.00*	[]

Please send only:

Checks drawn in US currency
Postal money orders.
Curreny (US or foreign)

NO CHECKS IN FOREIGN CURRENCY.

I WANT ALL THE BACK ISSUES FOR 1980 []

US, Canada & Mexico	*@ 20.00*	[]
Overseas air mail	*@ 24.00*	[]

FIRST & LAST NAME .....

ADDRESS (# & STREET) .....

CITY .....

STATE & ZIP CODE # .....

COUNTRY .....

I WOULD LIKE TO TAKE PART IN THE DESIGN AND PRODUCTION OF OUR OWN CLUB UTILITY MODULE. []

I HAVE N..... DOLLARS US AVAILABLE TO BUY M .....NUMBER OF MODULES.

I WOULD LIKE TO SEE THE FOLLOWING ROUTINES INCLUDED IN THE MODULE:

If needed, include routines (listings, documentation) on separate sheet of paper.

NOTE: Send only your subscription dues now. Do not send any money for the module yet. We will publish the results of this survey. Thank you.