Please note the change in zip code: 20706 from the old 20801. You may also use P.O. Box 710, Lanham MD 20706.

The lady who stood by my side for the last thirty three years suddenly passed away on June 18th. Annie had been very ill for the last fourteen years. But in spite of her illness she actively worked on the newsletter by stuffing envelopes, licking stamps and providing general encouragement to her hubby when things got difficult and the deadline was near. We will all miss her here in Lanham.

We finally located Fred Fish. He is alive and well and working. Fred is the author of the famous *SURVIVAL GUIDE FOR THE MASTER LIBRARY*, which has been a best seller for years. And rightly so. It contains a wealth of information on how to use the ML module more effectively, tricks to bypass a lot of key punching and make good use of the subroutines within the programs. It has a listing of each program and a flow chart with explanation how it works.

But, first for the sad news, Fred doesn't have any more copies for sale. Now, for the good news: we made a deal with Fred to have the manual printed by the TI PPC Club. I still had a copy which was in mint condition, which the printer used as the master for his photo-offset process. The 119 pages have been printed on 8½ by 11 inches paper, both sides of the paper, three-hole punch, one staple and with front and back in heavy yellow semi cardboard. Exactly 100 copies is all what I have to distribute, which will be done on a strictly first-come-first-serve basis. The original price was $ 12.50 but we were able to cut that down to $ 8.00. Add to this $ 2.00 for first class delivery in the US and Canada.($ 3.50 for A.O. Mail delivery overseas.)

Please allow six weeks for delivery, as at the time you receive this issue I will be on a one-month vacation.

Have a happy summer. I will see you back in September.

Maurice E.T. Swinnen

I'm glad to see all the new ideas. I am amazed at
all the new discoveries that have been uncovered
since you started writing *TI-PPC Notes!* Keep up the
good work.
>   ---Richard

I'm giving in to an overpowering urge to tell you
how really excellent your *TI* newsletter is. Even
though my use of my -59 keeps falling as I become
more dependent upon my computer, I always read the
Notes cover to cover and I always learn something new.
And usually I end up trying out some new programming
trick or new program that you've written up. I do
hope it keeps going well, and that you take good
care of yourself, too.
>   ---Burt
>   Potomac, MD

A real bargain. You have done a wonderful job with
the newsletter. Each issue is better than the last.
If you ever get to Dallas, be sure to call. I would
like to meet you face to face, even though I feel
that I know you from our correspondence and your
revelations in print. Hope you can keep up the
good work.
>   ---John
>   Dallas, TX

Notes have helped me a lot and I find uses for the
calculator I never knew existed. Thank you again,
and keep up the good work. I would like to hear more
about the new module when it is available.
>   ---Raymond
>   Carlsbad, CA

Please pardon this belated thanks for the copy of
Display that you so graciously sent me. The latest
issue of *TI-PPC* is superb. Your quality is better
than the   *(*)*   Keep up the excellent work.
The programs are good. I found a number of useful
tricks to improve my programs. In your previous
letter you said that you really like RPN. So do I,
but I believe that AOS is nearly as powerful. Based
upon my extensive experience, I often find AOS easier
to program. Next time I'll send you the names of a
few people who may wish to subscribe to the *TI-PPC.*
Thank you again for your prompt attention. My best
wishes and regards.
>   ---F
>   Edmonton, Alberta, Canada

I have received my final issue of *TI-PPC Notes*,
and like all of the other issues, it contained a
wealth of very creative programming ideas. During
the year that I have been receiving the *Notes*, my
programming skills have advanced more than I can
measure. You are to be congratulated for editing
a publication of this quality. Now for the negative
news: I shall not be renewing my subscription. The
reason is that I have become a "Deserter" and have
purchased an HP 41-C. I am now in the process of
learning RPN and I am attempting to translate my
large library of TI programs into the new language.
The frustration of hundreds of upon hundreds of mis-
reads of my cards, plus the time it took to load all
Alpha in machine code, etc. became too much. If some
day in the future, you too look to new horizons on a
41-C and start a new newsletter, please contact me to
become a charter subscriber. Your clear analytical
thinking would make great reading on ANY topic or new
hardware
>   ---Josef
>   Santa Monica, CA

*Are you sure you didn't leave the calculator
in FIX 2, each time you tried to read a card?
Ed.*

Thank you for the rapid response to my letter.
I received the 1st class (certainly in quality if not
in postage) copy yesterday. The bulk rate one arriv-
ed last Wednesday or Thursday. Perhaps it is time
for a new addition to Murphy's Law. "The proba-
bility of receiving the item in question increases
greatly after you send the letter of complaint."
I think that the *TI-PPC Notes* is so much better
than the   *(*)*   that it is <u>not</u> funny. The
Notes have given me a lot of help in programming the
TI-59 as well as providing some interesting programs.
It's publication is much more of a   *(*)*
designed to persuade one to buy more   *(*)*
programs. As for the $2. per issue complaint; even
if you are making a little money on this, I think you
would deserve it for all the work that you do! Since
I now have two copies of the *Notes*, I shall give the
second one to a friend who is thinking of buying a
TI-59.
>   ---Evan
>   Seattle, WA

(*) Expletive deleted.Ed.

CONTOUR GRAPH, FAST MODE.- As with the original HOEHENLINIENDIAGRAM program in
----------------------- v5n9/10p5, Harald M. Otto is also the author of this one.
It permits you to plot in Fast Mode, at an increase of a factor of two in speed and
in Normal Mode, at a snail's pace.

I received several attempts at putting this program in Fast Mode, but I judge
the one from Harald the most friendly to the user. On top of that, it is also the
fastest one, as Harald did not only convert it to Fast Mode, but did some clever
optimizations which gained considerable execution time.

The instructions are:
1. Key in all the program steps up to step 380 and record them on sides 1 and 2 of
   a mag card.
2. In use, read in sides 1 and 2 of the mag card. Make sure the ML module is plugged
   in if you plan to use the Fast Mode. Otherwise, ML module is not needed.
3. Press E. Calculator goes automatically into LRN mode. Start writing your function.
   See examples at the end of the program listing. The variable X_1 is in R07 and X_2
   is in R08. The value of the function must be in the display at the end of the
   routine. End the routine with GTO 118.Press LRN.
4. Enter parameters: press A. Printer will prompt you. See also original program in
   v5n9/10p5. Press R/S after each parameter has been put in the display.
   The prompting will be in this order:
   X1=start value for X1, X2=start value for X2, NL=number of paper strips,
   e=sharpness parameter, dX1=increment in X1, dX2=increment in X2,
   N=number of steps in the X1 direction, L0 through Ln= values of each level,
5. Select between Normal Mode and Fast Mode. Press either B or C.
   If you select B, Normal Mode, the display goes blank and the plotting will start
   right away.
   If you select C, Fast Mode:
   a. Display goes blank. Slide side 2 in the slot, to re-record that side, this time
      with your function included.
   b. A "4" is printed. Take a new mag card and slide it into the slot. Bank 4 will
      be recorded, containing the parameter data.
   c. The calculator nows goes automatically into Fast Mode. This wipes out program
      and data.
   d. Slide side 1 into the slot, to reload that side, a 1 will be printed.
      Slide side 2 into the slot, a 2 will be printed.
      Slide side 4 into the slot, a 4 will be printed.
   e. The display stays blank and the plotting starts.

While I am writing this, the postman brings the latest Programbiten from Sweden. One
of their members, Bo Nordlin, has put Harald's program into Fast Mode in a slightly
different way, reducing the program steps to a mere 314. It is, however, slower than
the one done by the guru himself.

Bo has a really impressive curve which he describes as a hyperbolic curve of
the form:

$$f(X1:X2)= \frac{(X1)^2}{2ca^2} - \frac{(X2)^2}{2cb^2}$$

and in which his parameters are : X1=X2=-10, dX1=dX2=.25, the level values are:
-12, -8, -4, 0, 4, 8, and 12. The e-value = .4 . He used about 90 steps in printing
this curve and employed four paper strips.

SEE HARALD'S PROGRAM ON NEXT PAGE.

CONTOUR GRAPH, Fast Mode,-Harald M. Otto.

```
000  00   0      059  12  12      118  42  STO     177  65   x
001  00   0      060  69  OP      119  03  03      178  43  RCL
002  00   0      061  06  06      120  43  RCL     179  13  13
003  00   0      062  07   7      121  11  11      180  95   =
004  00   0      063  05   5      122  32  X:T     181  22  INV
005  36  PGM     064  04   4      123  43  RCL     182  44  SUM
006  02  02      065  04   4      124  09  09      183  08  08
007  71  SBR     066  00   0      125  42  STO     184  43  RCL
008  02  02      067  03   3      126  00  00      185  12  12
009  39  39      068  69  OP      127  43  RCL     186  44  SUM
010  09   9      069  04  04      128  25  25      187  07  07
011  00   0      070  43  RCL     129  42  STO     188  97  DSZ
012  69  OP      071  13  13      130  04  04      189  01  01
013  00  00      072  69  OP      131  43  RCL     190  01  01
014  22  INV     073  06  06      132  03  03      191  00  00
015  58  FIX     074  98  ADV     133  75   -      192  43  RCL
016  22  INV     075  06   6      134  73  RC*     193  14  14
017  57  ENG     076  00   0      135  04  04      194  65   x
018  98  ADV     077  07   7      136  95   =      195  43  RCL
019  01   1      078  05   5      137  77  GE      196  12  12
020  99  PRT     079  07   7      138  01  01      197  95   =
021  25  CLR     080  05   5      139  51  51      198  22  INV
022  91  R/S     081  07   7      140  50  IXI     199  44  SUM
023  99  PRT     082  05   5      141  22  INV     200  07  07
024  25  CLR     083  07   7      142  77  GE      201  02   2
025  91  R/S     084  05   5      143  02  02      202  00   0
026  99  PRT     085  69  OP      144  20  20      203  55   ÷
027  76  LBL     086  01  01      145  69  OP      204  43  RCL
028  12   B      087  69  OP      146  34  34      205  13  13
029  98  ADV     088  02  02      147  97  DSZ     206  95   =
030  04   4      089  69  OP      148  00  00      207  44  SUM
031  04   4      090  03  03      149  01  01      208  08  08
032  00   0      091  69  OP      150  31  31      209  98  ADV
033  02   2      092  04  04      151  02   2      210  97  DSZ
034  69  OP      093  69  OP      152  22  INV     211  10  10
035  04  04      094  05  05      153  44  SUM     212  00  00
036  43  RCL     095  98  ADV     154  26  26      213  29  29
037  07  07      096  43  RCL     155  43  RCL     214  98  ADV
038  69  OP      097  14  14      156  13  13      215  98  ADV
039  06  06      098  42  STO     157  44  SUM     216  52  EE
040  04   4      099  01  01      158  08  08      217  22  INV
041  04   4      100  01   1      159  97  DSZ     218  52  EE
042  00   0      101  42  STO     160  02  02      219  91  R/S
043  03   3      102  06  06      161  01  01      220  43  RCL
044  69  OP      103  04   4      162  15  15      221  26  26
045  04  04      104  42  STO     163  43  RCL     222  22  INV
046  43  RCL     105  27  27      164  05  05      223  28  LOG
047  08  08      106  25  CLR     165  84  OP*     224  65   x
048  69  OP      107  42  STO     166  06  06      225  43  RCL
049  06  06      108  05  05      167  69  OP      226  00  00
050  07   7      109  05   5      168  26  26      227  95   =
051  05   5      110  42  STO     169  97  DSZ     228  52  EE
052  04   4      111  02  02      170  27  27      229  22  INV
053  04   4      112  08   8      171  01  01      230  52  EE
054  00   0      113  42  STO     172  06  06      231  44  SUM
055  02   2      114  26  26      173  84  OP*     232  05  05
056  69  OP      115  61  GTO     174  06  06      233  61  GTO
057  04  04      116  03  03      175  02   2      234  01  01
058  43  RCL     117  31  31      176  00   0      235  51  51


236  76  LBL     295  03   3      354  25  CLR
237  11   A      296  06   6      355  92  RTN
238  69  OP      297  03   3      356  76  LBL
239  00  00      298  07   7      357  19  D'
240  07   7      299  18  C'      358  85   +
241  42  STO     300  01   1      359  07   7
242  06  06      301  04   4      360  05   5
243  25  CLR     302  69  OP      361  04   4
244  16  A'      303  02  02      362  04   4
245  01   1      304  03   3      363  00   0
246  16  A'      305  01   1      364  02   2
247  03   3      306  69  OP      365  61  GTO
248  01   1      307  03  03      366  03  03
249  02   2      308  01   1      367  36  36
250  07   7      309  03   3      368  76  LBL
251  17  B'      310  05   5      369  10  E'
252  42  STO     311  03   3      370  85   +
253  01  01      312  00   0      371  02   2
254  35  1/X     313  01   1      372  07   7
255  01   1      314  03   3      373  00   0
256  04   4      315  02   2      374  01   1
257  95   =      316  07   7      375  61  GTO
258  42  STO     317  18  C'      376  03  03
259  25  25      318  98  ADV     377  36  36
260  03   3      319  91  R/S     378  76  LBL
261  01   1      320  76  LBL     379  15   E
262  03   3      321  13   C      380  92  RTN
263  06   6      322  02   2
264  17  B'      323  99  PRT
265  05   5      324  96  WRT
266  04   4      325  04   4      f(x1,x2)=x1·x2
267  17  B'      326  99  PRT
268  25  CLR     327  96  WRT     381  43  RCL
269  19  D'      328  31  RST     382  07  07
270  01   1      329  76  LBL     383  65   x
271  19  D'      330  16  A'      384  43  RCL
272  03   3      331  85   +      385  08  08
273  01   1      332  04   4      386  95   =
274  17  B'      333  04   4      387  61  GTO
275  43  RCL     334  00   0      388  01  01
276  09  09      335  02   2      389  18  18
277  75   -      336  95   =
278  43  RCL     337  76  LBL
279  01  01      338  17  B'
280  95   =      339  18  C'      f(x1,x2)=√(x1²+x2²)
281  10  E'      340  91  R/S
282  97  DSZ     341  72  ST*     381  43  RCL
283  01  01      342  06  06      382  07  07
284  02  02      343  69  OP      383  33  X²
285  75   -      344  06  06      384  95   =
286  98  ADV     345  69  OP      385  43  RCL
287  01   1      346  26  26      386  08  08
288  05   5      347  92  RTN     387  33  X²
289  69  OP      348  76  LBL     388  95   =
290  02  02      349  18  C'      389  34  √X
291  02   2      350  69  OP      390  61  GTO
292  01   1      351  04  04      391  01  01
293  01   1      352  69  OP      392  18  18
294  03   3      353  05  05
```

PATENTS.- Patent # 4,151,596 describes the principle behind DC-59, the $ 250.00 interface unit produced by the Science Accessories Corp. of Southport, Conn., which we mentioned in v5n6. Not much of a schematic, just block diagrams. But at least it explains a little more than what they themselves mention in their promotional literature. For this patent, write the Patent Bureau in Washington DC, as mentioned in v6n2p4.

TI-59 PRICES.- At some stores in this part of the country, the TI-59 has been offered at extremely low prices. I thought I had found an even lower price when I saw an ad from Olympic Sales, which offered the TI-59 at $ 199.95. However, W.J. McGonigle, in Wilmington, Delaware, sent me this ad from the the Philadelphia Inquirer. I couldn't believe my eyes. So I called up the firm. The ad is for real. At my question why they did offer the calculator at such a rediculously low price, they hinted at the imminent possiblity of TI common on the market with "something new." What it was exactly I could not ascertain. Everybody seems to be rather tightlipped about it.

<u>Software Programs for Structural Beam Design</u>   by George F. Poland

Available from: Olana Press, Box 212, 77 Ives Street, Providence,
Rhode Island 02906. Price: $29.95

     This manual contains a series of 42 programs, one for each of
the loading conditions shown in the American Institute of Steel
Construction's <u>Manual of Steel Construction.</u>

     Each program gives a sketch of the example showing the loading
of the beam and the shear and moment diagrams. Included are the
procedures one must use to operate the program. These procedures are
consistently incorrect. Example: "Press E, R/S. Enter load, press
A, R/S" when the program is similar to "LBL E CMs R/S LBL A STO 00
R/S ..." Both pressings of R/S will place the program in a location
it is not yet intended to be at.

     Many program listings contain errors -- too many open parentheses,
program steps not in correct order, and wrong memory being recalled
(RCL 08 instead of RCL 18). The programs that require more than 480
steps do not repartition themselves and none of the programs have any
provisions for a printer. When the author was confronted with the
errors, he replied that "an extremely thorough analysis of the programs"
had been made by this reviewer and that the author's programming skills
had improved since 1978 when these programs were written, but made no
mention of correcting (or even acknowledging) the errors.

     The equations and diagrams given in the AISC <u>Manual</u> are basic
enough for anyone to write programs from. If this manual is obtained,
a copy of mistakes found during this review may obtained by sending
a SASE to Frank Blachly, 5024 53rd Place, Hyattsville, Maryland 20781.

DAY OF THE WEEK.- Although the ML module has provisions for computing the day of the
----------------   *week, some members would like to know how they may compute it when*
using some other module. So I asked around and found out that a fellow named Zeller
has his name attached to the algorithm, called, not surprisingly, Zeller's Congruence:

$$f = ([2.6 \; m - 0.2] + k + D + [D/4] + [C/4] - 2 \; C) \; mod \; 7$$

in which   m is the month (March is 1, April is 2, etc. January and February are 11 and
          12 respectively, of the <u>previous</u> year)
          k is the day of the month
          D is the year of the century
          C is the century
          f is a single digit, 0 for Sunday, 1 for Monday, etc.
          mod 7 means "find the remainder after dividing by 7."
          The square brackets '[]' mean "greatest integer in."

For example, on what day will January 1, 1982 fall ?

m=11 , k=1 , C=19, D=81 (remember, "previous year")

$$f = ([2.6 \times 11 - 0.2] + 1 + 81 + [81/4] + [19/4] - (2 \times 19)) \; mod \; 7$$
$$f = ([\qquad 28.4 \quad] + 1 + 81 + [20.25] + [4.75] - (38 \quad)) \; mod \; 7$$
$$f = (\qquad 28 \qquad] + 1 + 81 + [20] + [4] - 38 \quad) \; mod \; 7$$
$$f = (\qquad 96 \qquad) \; mod \; 7$$
$$f = 96 \; / \; 7 \; = \; 13..... \qquad 96 - (7 \times 13) = 5$$
f = 5 for Friday.

Note that ML-20 returns a "6" for the same date, meaning "Friday."

MILEAGE AND EXPENSES SYSTEM by John Worthington and Emil Regelman

John Worthington and Emil Regelman have again combined efforts in their typically masterful way to provide for our pleasure and information a complete programming system, designed to help us all save money. The system, entitled "Mileage and Expenses" consists of three different programs to store expense information in a variety of categories (such as total trip expenses, non-fuel costs, etc.), manipulate the data in several ways, and print out information in an extremely useful format. The system must be seen to be believed! As a bonus to their monumental efforts, the three programs contain a number of unique and interesting routines.

A data packing routine (program #1 - DATA ENTRY, steps 083-153) keeps track of the day, month, and amount of a purchase, # of gallons, mileage for the purchase, and whether the expense was local or on a trip. ALL OF IT IN ONE MEMORY. Here's how it works, folks !!

The information is stored as +/- M.MMMMCCCCGGG DD , where...

"+" = local purchase, "-" = trip purchase, "M." = month of purchase, "MMMM" = # of miles, "CCCCC" = cost, "GGG" = gallons, and "DD" = day of month

The value in the memory is made negative for trip purchases.

The number of the month (through August) are stored directly as "M.". September through December are stored by adding 50 to the day of the month, then subtracting 8 from the number of the month (for example, May 17th would be stored as 5.xxxxxxxxxxxx 17, while September 17th would be stored as 1.xxxxxxxxxxxx 67).

Mileage (to the nearest 0.1) is stored directly as the next 4 digits, Costs (to the nearest cent) are stored as the next 5 digits. Gallons (to the nearest 0.1) are stored as the final 3 digits.

An interesting flag setting/testing routine was used in program #2 (steps 038-057/164-193). The flags are set in various combinations by the program to selectively determine the types of purchases to be printed or summed.
Flag 1 is set to skip the printing or summing of local fuel purchases;
Flag 2 is set to skip the printing or summing of non-fuel purchases;
Flag 3 is set to skip the printing or summing of trip fuel purchases;

DMS is used in an interesting way (program #1, step 126), to multiply the fractional portion of a number by 100, and store it in HIR 8.

P/R is used (program #1, step 021, and program #2, step 440) to store the contents of both the display and "t" register in HIR 7 and HIR 8 in a single step.

LBL invalidates the CLR instruction (LBL CLR, program #3, step 050) unless the CLR step is called by a direct address.

% is used (program #1, steps 175 and 180, program #2, steps 344 and 354) to compute total MPG and total cost/mile, by storing the appropriate values in R01, R03 and R04.

The Mileage and Expenses System consists of three programs, each designed to perform a different function.

Program #1 is used at the time of purchase to enter and record the data and provide some initial calculations.
Program #2 may be used to tabulate or manipulate raw purchase data.
Program #3 may be used to tabulate accumulated data.

The system will provide the following information:
• Vehicle descriptor.
• Initial and last odometer readings.
• Total number of purchases.
• Purchase #, date, miles, gallons, and cost of each purchase.
• Accumulated miles (local, trip and total).
• Accumulated gallons (local, trip and total).
• Accumulated costs (local, trip, fuel, non-fuel and total).
• Miles per gallon (local, trip and total).
• Cents per mile (local, trip, fuel, non-fuel and total).
• Summations of selected purchases.
• Listings or summations of selected purchases by type (local, trip, fuel, non-fuel and total).

Initial Operations
Initialize the system (only the 1st time used) as follows:
1) Read in side one of card #1
2) Store the initial odometer reading in memory 09
3) Store a descriptor in memory 00, if desired.
4) Press [E] and insert side 2 of card #1 to record data

Program #1 - DATA ENTRY
Program #1 will simultaneously sum entered the mileage, fuel usage, and cost data, store the raw purchase data, and calculate MPG and cents per mile.

1) Read in sides 1 and 2 of program card #1.
2a) For Fuel Purchases:
   i) Enter the current odometer reading...[X≷T]
   ii) Enter the date as MM.dd (e.g. Nov. 1 = 11.01)...[A]
   iii) Enter # of gallons (to the nearest 1/10, up to 99.9) ...[X≷T]
   iv) Enter the cost (up to $999.99)...[A]. The display will show the MPG for the current purchase.
2b) For Non-fuel Purchases:
   i) Enter date as MM.dd (e.g. Nov. 1 = 11.01)...[A]
   ii) Enter the cost (up to $999.99)...[A]. The display will show total cents per mile for all purchases.

3) Repeat 2a or 2b for additional purchases, as desired.
4) To record data, press [E] and insert side 2 of card #1.
   (also, see Note B for permanently preserving the data).
5) For error corrections see Note C.
6) Optional Calculations:
   [C] = MPG for local fuel purchases
   [C'] = "   " trip   "   "
   [D] = "   " all   "   "
   [B] = cents/mile for current purchase
   [B'] = "   "   " fuel   purchases
   [D'] = "   "   " all   "

   As each new purchase (in excess of 20) is entered, the
   earliest purchase is deleted. To retain the raw data
   for more than 20 purchases (for subsequent listing or
   manipulation), record every 20 purchases on one side of
   a separate mag card, as they are completed (if desired).
   Each storage mag card can thus hold the data for up to
   40 purchases.

C. Correcting data entry errors (program #1)...

Error condition	Corrective Action Required
Incorrect data entered and [A] pushed once	Press [RST] [R/S] and enter the correct data
Incorrect data entered and [A] pushed twice	Read in the data card again and enter the correct data
Incorrect data entered, [A] pushed twice, and data recorded	The correction is complex. Call John Worthington for required adjustments, at (301) 262-3851

D. The program is designed to process data from the lowest
   selected purchase to the highest. If the necessary data
   are not currently in memory, "card" will be printed to
   indicate that the program is waiting for a card contain-
   ing the correct data to be inserted.

E. The system (program #2) permits the separate manipulation
   of the raw data and the accumulated data. When new raw
   data are required (see D above), the program automatical-
   ly stores the accumulated data, reads the new data
   card, and then returns the accumulated data to their
   respective memories.

   These operations may also be conviently performed from
   the keyboard to permit the summing of non-consecutive
   purchases from different data cards. To store the accum-
   ulated data totals, enter 0 and press [D]. Read the
   appropriate data card read into the calculator and
   press [E]. To restore the accumulated data to their
   respective memories, enter 50 and press [D] again.

   Data manipulated by program #2 may be used by programs
   #2 and #3 (and recorded on data storage cards), but
   will not be operated on by program #1.

F. Program #1 may be used to provide the "optional calcula-
   tions" on data accumulated by program #2 (see 5 of
   program #1), if no printer is available,

G. Program #2 will operate with the M/U library. Change
   step 218 from "12" to "04."

H. The values listed for Initial and Last purchases are the
   corresponding odometer readings. These two listings
   are automatically deleted if the data is assembled by
   Program #2. The number of purchases value is the number
   of purchases summed to result in the tabulated values.

Program #2 - DATA MANIPULATION
   Program #2 will either list or accumulate the raw data
which has previously been stored by Program #1 (either in
memories or on magnetic cards).

1) Read in the raw data stored on side 2 of program card #1
   or from the appropriate data storage card (if the appro-
   priate raw data are not currently in the calculator).
2) Enter the number of the first purchase to be listed or
   summed and press [X≥T].
3) Enter the number of the last purchase to be listed or
   summed. (If no value is entered into the t register,
   only the displayed purchase will be designated.)
4) [A] will list the designated purchases. (see Note D)
   [B] will clear the accumulating memories and sum the
         designated purchases. (see Note D)
   [C] will sum the designated purchases to previously
         accumulated data. (see Note D and E)
5) The program may also be used to list or accumulate
   specific types of purchases, by pressing [A'] - [D']
   before [A], [B] or [C].
   [A'] = fuel purchases only
   [B'] = non-fuel purchases only
   [C'] = local fuel purchases only
   [D'] = trip fuel purchases only
   After pressing [A], [B] or [C] the types of
   purchases must be designated again for a subsequent
   operation.
6) If an error is made in pressing [A'] through [D'],
   press [RST].

Program #3 - PRINT ROUTINE
   Program #3 will provide a complete tabulation of the
accumulated data whether accumulated by the Data Entry
program or the Data Manipulation program.

1) Read in the data from side 2 of card #1, if the data
   are not currently in the calculator from either Program
   #1 or Program #2 operations.
2) [A] prints...
   • Purchases (Initial, Last and Number) See Note C.
   • Miles (total, local and trip)
   • Gallons (total, local and trip)
   • Costs (total, fuel, non-fuel, local fuel, and trip fuel)
   • Miles per Gallon (total, local and trip)
   • Cents/Mile (total, fuel, non-fuel, local fuel and trip)
3) [A'] prints only the tabulated miles
   [B']   "   "   "   "   gallons
   [C']   "   "   "   "   costs
   [D']   "   "   "   "   MPG
   [E']   "   "   "   "   cents/mi.

NOTES
A. MPG data are always displayed in FIX 1 mode, and cost
   data are always displayed in FIX 2 mode.

B. The vechicle desciptor (R_{00}), accumulated data (R_{01-08}),
   odometer reading (R_{09}), and the raw data (R_{10-29}) may be
   recorded by pressing [E] in program #1. R_{10-29}
   contain the raw data for the most recent 20 purchases.

To designate a purchase as a trip purchase:
Enter the date as a negative number and
press A. This is relevant to program #1.

PROGRAM 2

PROGRAM 1

PROGRAM 3

PROGRAM 2 CONTINUED

KEYED LABEL PRINTER.- Don Lambert, of Lambert Programming Service, Los Angeles, CA,
------------------- wrote this version. It is shorter and easier to use than the
one published in v6n3p8.
          To use it, press RST R/S. When the display shows a flashing zero,
the program is ready. Press any key and in a few seconds you will see

478   CC   LLL in which CC is the code corresponding to the key you pressed and
             in which LLL is the mnemonic of the key you pressed.

You cannot use the keys LRN, SST, BST, DEL, INS, the 2nd key all by itself.
But you can use the 2nd key in combination with all the other keys, to obtain the
second function of each key. For example, you can even press 2nd CP and your program
will NOT be wiped out.

Keys			Step			Step			Step			Step			Step			Step		
478	12	B	000	25	CLR	048	91	R/S	096	76	LBL	144	76	LBL	192	76	LBL	240	76	LBL
			001	61	GTO	049	09	9	097	80	GRD	145	36	PGM	193	34	ГX	241	10	E'
			002	02	02	050	76	LBL	098	08	8	146	03	3	194	03	3	242	01	1
478	42	STO	003	74	74	051	19	D'	099	00	0	147	76	LBL	195	76	LBL	243	52	EE
			004	76	LBL	052	01	1	100	76	LBL	148	61	GTO	196	42	STO	244	95	=
478	43	RCL	005	99	PRT	053	76	LBL	101	77	GE	149	06	6	197	04	4	245	52	EE
			006	09	9	054	90	LST	102	07	7	150	76	LBL	198	76	LBL	246	00	0
478	32	X:T	007	76	LBL	055	09	9	103	76	LBL	151	16	A'	199	24	CE	247	01	1
			008	98	ADV	056	00	0	104	76	LBL	152	01	1	200	02	2	248	59	INT
			009	09	9	057	76	LBL	105	07	7	153	76	LBL	201	04	4	249	85	+
478	34	ГX	010	76	LBL	058	88	DMS	106	76	LBL	154	60	DEG	202	76	LBL	250	07	7
			011	89	π	059	08	8.	107	67	EQ	155	06	6	203	14	D	251	69	OP
478	24	CE	012	08	8	060	76	LBL	108	06	6	156	00	0	204	01	1	252	17	17
			013	76	LBL	061	87	IFF	109	76	LBL	157	76	LBL	205	76	LBL	253	09	9
			014	97	DSZ	062	08	8	110	75	-	158	55	÷	206	40	IND	254	02	2
478	25	CLR	015	09	9	063	76	LBL	111	07	7	159	05	5	207	04	4	255	00	0
			016	76	LBL	064	78	Σ+	112	76	LBL	160	76	LBL	208	00	0	256	00	0
478	13	C	017	79	Σ	065	07	7	113	57	ENG	161	54	)	209	76	LBL	257	93	.
			018	07	7	066	76	LBL	114	05	5	162	05	5	210	33	X²	258	07	7
478	65	×	019	76	LBL	067	86	STF	115	07	7	163	76	LBL	211	03	3	259	06	6
			020	96	WRT	068	08	8	116	76	LBL	164	45	YX	212	76	LBL	260	00	0
			021	09	9	069	76	LBL	117	47	CMS	165	04	4	213	32	X:T	261	08	8
478	43	RCL	022	76	LBL	070	68	NOP	118	04	4	166	76	LBL	214	03	3	262	06	6
			023	69	OP	071	06	6	119	07	7	167	53	(	215	76	LBL	263	09	9
478	55	-	024	06	6	072	76	LBL	120	76	LBL	168	05	5	216	23	LNX	264	95	=
			025	76	LBL	073	85	+	121	37	P/R	169	76	LBL	217	02	2	265	42	STO
			026	95	=	074	08	8	122	03	3	170	35	1/X	218	03	3	266	60	60
478	81	RST	027	09	9	075	76	LBL	123	07	7	171	03	3	219	76	LBL	267	06	6
			028	76	LBL	076	58	FIX	124	76	LBL	172	76	LBL	220	13	C	268	69	OP
478	91	R/S	029	59	INT	077	05	5	125	27	INV	173	52	EE	221	01	1	269	17	17
			030	05	5	078	08	8	126	02	2	174	05	5	222	76	LBL	270	71	SBR
478	47	CMS	031	76	LBL	079	76	LBL	127	76	LBL	175	76	LBL	223	30	TAN	271	04	04
			032	94	+/-	080	48	EXC	128	71	SBR	176	25	CLR	224	03	3	272	75	75
478	50	I×I	033	09	9	081	04	4	129	07	7	177	02	2	225	00	0	273	98	ADV
			034	76	LBL	082	08	8	130	76	LBL	178	05	5	226	76	LBL	274	07	7
			035	49	PPD	083	76	LBL	131	17	B'	179	76	LBL	227	22	INV	275	69	OP
478	39	COS	036	04	4	084	38	SIN	132	01	1	180	15	E	228	02	2	276	17	17
			037	76	LBL	085	03	3	133	76	LBL	181	01	1	229	02	2	277	07	7
478	29	CP	038	93	.	086	08	8	134	70	RAD	182	76	LBL	230	76	LBL	278	01	1
			039	09	9	087	76	LBL	135	07	7	183	50	I×I	231	12	B	279	42	STO
478	63	NOP	040	76	LBL	088	28	LOG	136	00	0	184	05	5	232	01	1	280	60	60
			041	39	COS	089	02	2	137	76	LBL	185	00	0	233	76	LBL	281	06	6
478	89	π	042	03	3	090	76	LBL	138	66	PAU	186	76	LBL	234	20	CLR	282	69	OP
			043	09	9	091	81	RST	139	06	6	187	44	SUM	235	02	2	283	17	17
			044	76	LBL	092	08	8	140	76	LBL	188	04	4	236	00	0	284	25	CLR
478	25	CLR	045	29	CP	093	76	LBL	141	65	×	189	76	LBL	237	76	LBL	285	61	GTO
			046	02	2	094	18	C'	142	06	6	190	43	RCL	238	11	A	286	04	04
			047	76	LBL	095	01	1	143	05	5	191	04	4	239	01	1	287	79	79

PROGRAMS TO COME.- In spite of summer and "everybody being on vacation", the revie-
----------------- wers have been busy. They have returned to me with recommendations
to publish a lot of good programs. Those will appear in future issues, space permit-
ting:
          TRUTH IN LENDING, by Glenn Ellis, a well documented financial program.
          KENDALL'S COEFFICIENT OF CONCORDANCE, by Glenn Ellis. A fine statistics
program, with all the bells and whistles, for the TI-59.
          XOR, a Logic game program, by Dejan Ristanović.
          REGISTER OPERATIONS, by Philip Brassine.
          13-DIGIT LIST, by Lem Matteson. The ultimate 13-digit lister. This one
is superb, though. Does everything by brush your shoes.
          COMPLEX KEYBOARD, by Bill Beebe, Jr. Converts your calculator into a
complex one. It even sports an RPN-type single-level push stack.
          ANNUITIES II, by Bill Beebe, Jr. Similar to ML-19, with a new twist.
          PRINTED WIRING BOARD, by Wally Agy. It isn't a Computervision Cadds IV
CAD-CAM yet, but we are getting closer.
          SHOPPING LIST, by Jules Bell. This is THE program that will put you in
good stead with the sweetheart in your life.

HANDBOOK OF OCEANIC PIPELINE COMPUTATIONS by A. Marks

Publishers: PennWell Books, P.O. Box 1260, Tulsa, Oklahoma, 74101. Price: $50.00

This book is one of the first books published specifically for oceanic pipeline
computations. The book covers the technical and economical feasibility, design
and construction of oceanic pipelines. The author divides the handbook into three
principal sections; 1) Preliminary Pipeline Sizing, 2) Preliminary Corridor Selection,
and 3) Final Design. Each section is well written with numerous technical references
and programmable calculator programs. The author states in his forward that the
programs have been included for convenience and portability. The programs were
written for the HP-67.

For each program the author includes the theoretical formulae and its source, the
required input and what is being computed, the user instructions, an example problem
and the program listing. The write-up for the programs are clear, concise and easy
to follow. Even though the programs are in RPN, the write-up provides all the
necessary information to translate the program into AOS or even BASIC if one so
desires.

The general asset of this handbook is the acculumation of a wide variety of technical
and economical data on oceanic pipeline design. The calculator programs assist in
the technical and economical feasibilty studies by freeing the design engineer from
time consuming and error-prone computations. Overall the handbook is an invaluable
aid to those who are involved in oceanic pipeleine computations.

Review by Barry Franz.

GRAPHICS MODE.- Dave Leising has found that besides at step 024 it is also possible to
---------------   insert the 1F hexadecimal code into steps 008 and 016. The basic proce-
dure to put hexadecimal codes into TI-59 user RAM is as follows:
1. Insert code MN in user RAM at the first step in an octet. Leave RAM program counter
   positioned at this step.
2. Get machine into the "Display Firmware" mode.
3. Position the firmware program counter at the same step number as RAM counter, display-
   ing code PQ.
4. Execute : 2nd INS.
5. Exit "Display Firmware" mode, reset fix and partitioning as required.
6. The result of the hexadecimal subtraction MN-PQ will be found in place of the original
   MN code in RAM.
         EXAMPLE: From turn-on: GTO 016 LRN) BST LRN 9 OP 17 PGM 12 SBR 444
                                R/S P/R LRN INS LRN RST INV ENG 6 OP 17 CLR
                                Go into TRACE mode, GTO 016   LRN LRN LRN
Step 016 in RAM is FF and displays as 016 05 or 017 o5 and trace-prints as 169 5.
This is the result of the hexadecimal subtraction of 55 from 54.

GRAPHICS MODE.- *The results of the discovery of the Graphics Mode are coming in. Dave*
------------   *Leising, who has an emulator at his disposition (lucky stiff) has disco-*
vered that it is possible to put the "magic hex code" that produces Graphics Mode at
several other program locations: 008, 016 and 184. Several other members, including
Dave, discovered also that the number of NOPs in front of the location where you place
the hex code will determine how many dots of the original character you are going to
print. I will publish a summary in next issue I hope.

Here are a few selected samples of new programs in Graphics Mode. Some of them
are, admittedly, rather silly, but serve only as demonstrators.

The first one is by Dave Leising and he calls it "Mosaic." The finished product
looks like hieroglyphics or inscriptions found on the tikis of Easter Island.The instruc-
tions are deliberately omitted, as a challenge to you to come up with the correct ones.

The next one, also by Dave Leising, is a modification of Richard Snow's Print
Code Table of v6n4/5p4. It is self-contained and also prints reference code lines above
the partial lines. Because Dave put the hex code at 008, the partial lines consists of
only two elements (dots) printed each time.

To use this program:
1. Load card side 1.
2. From the keyboard : GTO 008 9 OP 17 PGM 12 SBR 444 R/S P/R LRN INS   LRN
*   RST GTO 009 LRN INV SBR LRN A*
3. After printout is completed, just press R/S or A for another copy.

The last one is by Frèdèric De Mees. He took up my "Mickey Mouse" challenge and
did the little critter in Graphics Mode.

To key in the program: Key in the commands, steps 000 through 147 and registers
90 through 99 in 10 OP 17 partition. Then go to 6 OP 17 partitioning and record bank 1
on side 1 of a mag card.

Next key in the print code in registers 00 through 89. Use, of course, 10 OP 17
partitioning. Record banks 2, 3 and 4 respectively on sides 2, 3 and 4 of mag cards,
still in 10 OP 17 partitioning.

And lastly, key in new print code data in registers 30 through 59 and record
bank 3 on one side of the third mag card. Mark it as"side 5." Still in 10 OP 17.

You now should have three mag cards: sides 1 and 2 on the first one, sides 3
and 4 on the second one and side 5 on the third one.

As you will undoubtedly have noticed, most of the registers have been listed
by means of a "13-digit list" program. And you might have noticed also that most of the
registers contain a 12-digit number. Just to remind you how it is done:
Take, for example, register 30, side 5:
Enter 83 19 82 70 37 STO 30 100 PROD 30 19 SUM 30

To use the program: Press CLR and load in side 1. Then initialize the Graphics
Mode by : E CLR PGM 19 GTO 024 SBR 045 P/R LRN ·INS LRN RST A

At this point the paper will advance. You now read, in any order, sides 2, 3
and 4. DO NOT PRESS CLR OR ANY OTHER KEY. Just slide in the mag cards, one after another.
The program contains automatic read instructions.

Be patient and watch the print out. It will take about a quarter of an hour.
When everything is silent for a while, slide in side 5, after which some more printing
will be produced. Take a clean towel from the linen closet, put all the printing in it
and throw it in the garbage. Sorry, take scissors and a bottle of your best paste or
glue and from this unsightly mess try to compose a nice drawing representing Mickey
himself. That's all !

Dave Leising says, and several other members have echoed this sentiment in one
way or another, "I strongly feel that time will show that this discovery may be the
most significant yet made on the TI-59. There are 156 two-digit hex codes to be inves-
tigated as initial operators, modifiers, indirect modifiers, addresses, etc.etc. Who
knows what wonderful new capabilities will be discovered over the next year or so."

Somewhere else in this issue you will find Dave's recipe for emplacing hexa-
decimal code into user RAM. It is a sound basis to investigate all possible hex codes
and their possible effects. The only thing you need is time, which sometimes translates
into patience, perseverance and dilligence.

On the next page you will find the program listings for the three programs
described above.

MOSAIC-   Dave Leising

000	92	RTN	012	12	12	023	71	SBR	034	24	CE
001	76	LBL	013	71	SBR	024	00	00	035	65	×
002	11	A	014	00	00	025	06	06	036	05	5
003	61	GTO	015	29	29	026	61	GTO	037	22	INV
004	00	00	016	84	OP*	027	00	00	038	28	LOG
005	10	10	017	12	12	028	10	10	039	95	=
006	69	OP	018	97	DSZ	029	36	PGM	040	65	×
007	05	05	019	12	12	030	15	15	041	05	5
		0	020	00	00	031	71	SBR	042	22	INV
009	92	RTN	021	13	13	032	88	DMS	043	28	LOG
010	04	4	022	25	CLR	033	85	+	044	95	=
011	42	STO							045	92	RTN

PRINT CODE TABLE-   Dave Leising

	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	×	*	[	•	(	)	,	
6	↑	%	!	/	=	!	×	R
7	±	?	+	:	I	^	II	Σ

000	92	RTN
001	76	LBL
002	11	A
003	61	GTO
004	00	00

005	17	17
006	69	OP
007	05	05
008	23	LNX
009	00	0

010	00	0	040	00	0	070	00	0	100	25	CLR	130	55	+
011	00	0	041	00	0	071	02	2	101	71	SBR	131	01	1
012	00	0	042	06	6	072	00	0	102	00	00	132	00	0
013	00	0	043	69	OP	073	00	0	103	06	06	133	00	0
014	00	0	044	03	03	074	69	OP	104	98	ADV	134	85	+
015	00	0	045	07	7	075	02	02	105	01	1	135	01	1
016	98	ADV	046	00	0	076	03	3	106	85	+	136	95	=
017	22	INV	047	00	0	077	00	0	107	09	9	137	82	HIR
018	58	FIX	048	00	0	078	00	0	108	09	9	138	36	36
019	25	CLR	049	08	8	079	00	0	109	09	9	139	82	HIR
020	22	INV	050	00	0	080	04	4	110	09	9	140	38	38
021	57	ENG	051	00	0	081	00	0	111	09	9	141	69	OP
022	01	1	052	69	OP	082	00	0	112	00	0	142	05	05
023	69	OP	053	04	04	083	00	0	113	35	1/X	143	25	CLR
024	01	01	054	69	OP	084	05	5	114	95	=	144	71	SBR
025	02	2	055	05	05	085	69	OP	115	82	HIR	145	00	00
026	00	0	056	69	OP	086	03	03	116	35	35	146	06	06
027	00	0	057	00	00	087	06	6	117	01	1	147	98	ADV
028	00	0	058	69	OP	088	00	0	118	85	+	148	82	HIR
029	03	3	059	05	05	089	00	0	119	09	9	149	18	18
030	00	0	060	01	1	090	00	0	120	09	9	150	59	INT
031	00	0	061	52	EE	091	07	7	121	09	9	151	22	INV
032	69	OP	062	06	6	092	00	0	122	93	.	152	77	GE
033	02	02	063	22	INV	093	00	0	123	09	9	153	01	01
034	04	4	064	52	EE	094	69	OP	124	35	1/X	154	06	06
035	00	0	065	69	OP	095	04	04	125	95	=	155	25	CLR
036	00	0	066	01	01	096	07	7	126	82	HIR	156	92	RTN
037	00	0	067	01	1	097	32	X:T	127	37	37	157	61	GTO
038	05	5	068	00	0	098	69	OP	128	22	INV	158	11	A
039	00	0	069	00	0	099	05	05	129	59	INT			

There is not enough space to fit the MICKEY MOUSE program here. See next page.

PATENTS.- In our ongoing, and never satisfied, quest for knowledge (translate this one
------- as "we are nosey") we have discovered a few more relevant patents:
        Patent # 3,934,233. READ-ONLY MEMORY FOR PROGRAMMABLE CALCULATORS. Jan.20,
1976. Roger J. Fisher and Gerald D. Rogers. This patent describes the ROM, implemented
as a large-scale-integrated circuit chip.
        Patent # 4,020,465. THERMAL LINE PRINTER. April 26, 1977. Michael J. Cochran
and Stephen B. Hamilton. From the date, I would say "this is not the PC-100, because it
existed already by that time. It might be an improved version we haven't seen as yet.
        Both patent descriptions contain a wealth of flow charts, detailed mechanical
drawings and tables.
        By the way, the US Patent Office tells me that the reason it usually takes
about six weeks to fill an order is not so much bureaucratic sloth as it takes time for
your check to be cashed, logged at several places, etc. A much quicker way if you order
patents very often, they say, is to buy 50 coupons at $ 25.00. That is still $ 0.50 if
my TI-59 is correct. You just send them a coupon each time you need a copy of a patent
and they will fill your order within 20 days, they tell me. Booklets with 50 coupons
may be purchased from U.S. Patent and Trade Mark Office
                        Office of Finance
                        Washington DC 20231, USA.

MICKEY MOUSE- Frederic De Mees.

MICKEY MOUSE

1100111100.	90	
1100111111.	91	
1111000000.	92	
1111000011.	93	
1111001100.	94	
1111001111.	95	
1111110000.	96	
1111110011.	97	
1111111100.	98	
1111111111.	99	

000	92	RTN	032	69	OP	064	42	STO	106	01	1
001	76	LBL	033	00	00	065	65	65	107	44	SUM
002	11	A	034	43	RCL	066	83	GO*	108	59	59
003	98	ADV	035	58	58	067	65	65	109	04	4
004	22	INV	036	22	INV	068	55	÷	110	42	STO
005	58	FIX	037	28	LOG	069	01	1	111	58	58
006	25	CLR	038	52	EE	070	00	0	112	61	GTO
007	22	INV	039	22	INV	071	95	=	113	00	00
008	96	WRT	040	57	ENG	072	42	STO	114	35	35
009	25	CLR	041	65	×	073	65	65	115	76	LBL
010	22	INV	042	33	X²	074	22	INV	116	15	E
011	96	WRT	043	95	=	075	59	INT	117	01	1
012	25	CLR	044	35	1/X	076	65	×	118	00	0
013	22	INV	045	65	×	077	02	2	119	69	OP
014	96	WRT	046	73	RC*	078	75	-	120	17	17
015	61	GTO	047	59	59	079	59	INT	121	92	RTN
016	00	00	048	95	=	080	82	HIR	122	43	RCL
017	33	33	049	22	INV	081	04	04	123	66	66
018	76	LBL	050	59	INT	082	95	=	124	69	OP
019	16	A'	051	65	×	083	65	×	125	01	01
020	25	CLR	052	03	2	084	05	5	126	43	RCL
021	69	OP	053	00	0	085	95	=	127	67	67
022	05	05	054	00	0	086	42	STO	128	69	OP
023	41	SST	055	32	X:T	087	64	64	129	02	02
024	74	SM*	056	03	3	088	73	RC*	130	43	RCL
025	80	80	057	22	INV	089	65	65	131	63	63
026	00	0	058	28	LOG	090	84	OP*	132	69	OP
027	00	0	059	95	=	091	64	64	133	03	03
028	00	0	060	59	INT	092	29	CP	134	98	ADV
029	00	0	061	77	GE	093	82	HIR	135	69	OP
030	00	0	062	00	00	094	14	14	136	05	05
031	00	0	063	69	69	095	67	EQ	137	02	2
						096	01	01	138	00	0
						097	03	03	139	69	OP
						098	16	A'	140	00	00
						099	69	OP	141	69	OP
						100	00	00	142	04	04
						101	16	A'	143	69	OP
						102	97	DSZ	144	05	05
						103	58	58	145	61	GTO
						104	00	00	146	01	01
						105	35	35	147	03	03

469992853709.	00	972713849742.	30	0.	60			831982703719.	30
701713822809.	01	753919772949.	31	0.	61			751843703759.	31
711982843849.	02	753949976399.	32	0.	62			981713999921.	32
701693929691.	03	423449041692.	33	3001413617.	63			693929686636.	33
462929732929.	04	493949801752.	34	0.	64			689689123639.	34
702713879692.	05	853499731942.	35	0.	65			689689689689.	35
452879753959.	06	703449701982.	36	3024.	66			689689689689.	36
949989729729.	07	723759701692.	37	1526174500.	67			689689989926.	37
729729709699.	08	468751872993.	38	0.	68			966986986986.	38
639689689689.	09	349801993853.	39	11.	69			996966926686.	39
689689689689.	10	929991992843.	40	1100.	70			689689689689.	40
693719719709.	11	969491992943.	41	1111.	71			689846846766.	41
709719639689.	12	469991992733.	42	110000.	72			326696986846.	42
639689689689.	13	879831982738.	43	110011.	73			841833999841.	43
639639689689.	14	711923748748.	44	111100.	74			712848967716.	44
993699692923.	15	718718733809.	45	111111.	75			966846846846.	45
359712843959.	16	841733789841.	46	11000000.	76			846846846846.	46
712963809712.	17	753789941743.	47	11000011.	77			846846846991.	47
709759712393.	18	859761962452.	48	11001100.	78			967986926686.	48
699692943849.	19	631912843992.	49	11001111.	79			636686696686.	49
493929833999.	20	961942899961.	50	11110000.	80			135000000000.	50
712399639689.	21	939399921633.	51	11110011.	81				
689689125689.	22	429921753919.	52	11111100.	82				
689833849993.	23	421939841939.	53	11111111.	83				
929699393969.	24	841693859341.	54	1100000000.	84				
692993969993.	25	695939841713.	55	1100000011.	85				
969832993969.	26	719921703719.	56	1100001100.	86				
842759369851.	27	120000000000.	57	1100001111.	87			4.	58
762713929961.	28	4.	58	1100110000.	88			30.	59
932853849691.	29			1100110011.	89				

ASTRONOMY.- Turner F. Morgan, of Omaha, Nebraska, is an HP PPC member who received the
--------- Astro programs I translated from the German. Turner is a senior field
engineer with Westinghouse and works on the Defense Meterological Satellite Program. He
is one of the few people I know who are able to "legally" play with a hobby at work.
        Turner also has a few colleagues at work who are receiving the TI PPC NOTES
and he noted that David Bartholomew had some trouble using the Astro programs. (see
v6n1p11) So he set out to check into the matter and has converted most of the Astro
programs for HP-41C use. Neither the Solar nor the Planets programs had some real pro-
blems. But the Phase programs showed some discrepansies. On top of that, the Lunar pro-
gram is still giving him some problem he has not been able to solve as yet.
        I have a total of 17 pages of HP-41C listings and sample runs which I will
sent to anyone needing it. Please send me $ 3.00 to cover copying and mailing costs.
Just ask for the updated HP-41C Astro programs.

TI-59 RESERVOIR ENGINEERING MANUAL by R. Hollo and H. Fifadara of
Keplinger and Assocs. Publishers: PennWell Books, P.O.Box 1260,
Tulsa, Oklahoma, 74101. Price: $50.00

This book is one of the first books published specifically for the TI-59
for use in reservoir engineering (petroleum engineering). The authors
assume that the user has a strong understanding of reservoir engineering
and a basic understanding of the TI-59. The book is divided into five
sections: (1) Basic reservoir engineering, (2) Economic analysis, (3)
Geology, (4) Log interpretation, and (5) Advanced reservoir engineering.
Each section contains either four or eight calculator programs for use.
The documentation of these programs is among the finest one could come
across. The documentation begins with a write-up of what the program is
calculating, formulas used, and excellent reference sources where additional
discussions may be found. Next, a well devised and well thought out flow
charting of the paths the program will take. An example problem follows
written on the TI program record sheets with a printout of the results.
Finally the actual program listing is given. The programs have excellent
alphanumeric descriptors utilizing the OP 04- OP 06 sequence.
The book is well written with much thought given to the sequences and
sectioning. Overall the book is complete, concise, and easy to comprehend.
Reservoir engineers or those involved in some aspect of reservoir
engineering will have an invaluable aid through the use of this book
and its programs.

Review by Barry Franz.

LOCAL CLUB.- James Goodwin has organized a TI-59 club in the Thousand Oaks, CA
-----------   area. Meetings are held each 3th Tuesday of the month from 7 to 9
PM at the TI-store in Thousand Oaks.
        Newcomers please call James at home (805) 492-3488 or write him at
2932 Hyacinth Court, Thousand Oaks CA 91360.

CARRYING CASE.- Charles G. Stipp, one of the early pioneers dating back to SR-52
-------------   days, got slightly unhooked when he saw the astronomical price of the
super-deluxe TI-59/PC-100 carrying case I was promoting in v6n3. He reminds me of
availability of a more "democratic" type of case at Radio Shack. It is # 21-542
and retails for $ 14.95. (The price has gone up a bit, according to Radio Shack,
but it is still under $ 20.00) Charles has carried his machine around in it on
airplanes all over the country.(I did too, Charles, until somebody thoughtful
sent me the $ 85.00 variety one.) I thank Charles for reminding me to keep both
feet sturdily on the ground.
----

by Bob Arendt

Why it Exists: 'Fast-Mode' is really the natural state of the calculator when executing code in a library ROM. Actually, it might be more appropriate to call the user-mode of operation the 'Slow-Mode'. During Slow-Mode program execution, the calculator performs a series of display formatting calculations after the execution of every op-code. This ensures that the display register contains a displayable value after the execution of each op-code. This feature is necessary for the operation of the TI-59 trace features (PAUSE every step, or TRACE). During execution of the library modules (which supposedly contain fully debugged software) this feature is disabled, resulting in a faster executing program (Notice that a library module will not trace or pause!). By properly bollixing the calculator it is possible to halt the calculation while in this state, enabling the user to run his/her programs under the auspices of Fast-Mode.

A Demonstration and Explanation: The code to the right was entered into the calculator and saved on a magnetic card. The code of steps 000-015 shall be refered to as the "Header Program" and is the section which starts the calculator off in Fast-Mode. I shall now explain the program's execution step-by-step, starting at instruction zero.

1) RST R/S: The calculator reads in instructions in blocks of 8 from the RAM memory (user program) into a command buffer. It executes steps 000-007 without undue difficulty (note the subroutine call lies on a memory boundary). Instructions 008-015 are read into the command buffer, and the subroutine jump is executed. But while executing the code in ROM the calculator comes across a LRN code (the module was not intended to be entered at this point) which stops it dead in its tracks. The calculator is halted with a FIX 0, partion 479.59, and in the infamous Fast-Mode. The program memory is cleared. The command buffer still contains instructions 008-015, and is ready to resume execution (still in Fast-Mode) from step 010. Pressing any non-numeric key except LRN, CLR, or CE results in the execution of that key and resumption of program execution. RST or PGM will result in an exit from Fast-Mode if used inside or outside of a program.

2) Read Card or R/S: The program memory is cleared, so we must enter the program by card (or by hand . . . but I'll let you figure it out). CLR may be pressed, and a card read in. After the card is read in the calculator immediately executes step 010-013, where it is halted by a R/S. A PAU must precede the R/S. In order for the calculator to recognize a R/S it must first have a valid number in the display register; the PAU, PRT, or EE INV EE sequence will perform this operation. If this is not done, the R/S might be treated as a NOP.

	Step	Code	Key
H	000	00	0
E	001	00	0
A	002	00	0
D	003	00	0
E	004	00	0
R	005	36	PGM
	006	02	02
	007	71	SBR
	008	02	02
	009	39	39
	010	09	9
	011	00	0
	012	66	PAU
	013	91	R/S
	014	66	PAU
	015	91	R/S
P	008	02	2
R	009	39	COS
O	010	09	9
G	011	00	0
R	012	66	PAU
A	013	91	R/S
M	014	66	PAU
	015	91	R/S
	016	69	OP
	017	00	00
	018	01	1
	019	06	6
	020	03	3
	021	02	2
	022	03	3
	023	01	1
	024	01	1
	025	07	7
	026	00	0
	027	00	0
	028	69	OP
	029	01	01
	030	69	OP
	031	03	03
	032	69	OP
	033	05	05
	034	66	PAU
	035	91	R/S

3) <u>CLR, Read Card or R/S</u>: The display now shows a 90. Hit the CLR key, and read in the next card (or R/S). Steps 014-015 are executed, again halting the calculator.

4) <u>CLR, Read Card or R/S & Program</u>: Aha! I bet you were wondering why I repeated steps 008-015 in the program listing. During the previous steps 1) - 3) the calculator was executing the contents of the command buffer, which was un-changed when the calculator entered Fast-Mode. The three R/S's provide ample opportunity to reload the calculator memory <u>without affecting the contents of the command buffer</u>. When finished executing the contents of the buffer it grabs its <u>next</u> buffer-full starting at step 008-015. For convenience I assume the header exists on the <u>same</u> card the program resides in. If not, the calculator starts execution at step 008 of the new user program. If it's the <u>same</u> data, then the execution sequence continues as shown. Note that there are a total of 5 halts where cards may be read into memory.

Program execution proceeds 'normally' from step 008, but executing at approx-imately twice the normal rate.

See SUMMARY of Fast Mode on next page.

FAST MODE.- Of the many comments I have received lately about Fast Mode, most of them
--------- *seem to have little or no <u>practical</u> value. However, Frederic De Mees tells me that Fast Mode initialization <u>does not</u> clear the HIR registers. Now, that might have some very practical value. Suppose you have a Normal Mode program that has to be followed by a Fast Mode program. The Fast Mode program has to use some data generated by the Normal Mode program. Normally, the data would be wiped out by the Fast Mode ini-tialization. Now, just keep them, temporarily, in the HIR registers, from which the Fast Mode program will retrieve them.*

WANTED.- This time we need two programs. I will publish the best ones and will mention
------ *all contributors. (honestly, I will not forget this time. I know you love to see your name printed on something other than a traffic ticket.)*

The first challenge is a (fast) program to add, subtract, multiply and divide entries of mixed feet, inches, fractions of inches. Both calculator-only and printer versions combined are preferred, as this program would appeal to civil engineers working both "on the job" and at the office.
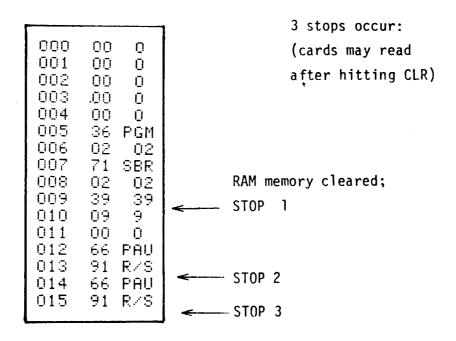
The second program challenge is a multi-strip Cartesian Plot. Unlike Jared Weinberger's celebrated contribution, this one should have at least a 60 by 60 point plot capability. Speed is good, but not a must.

ANGLE CONVERTERS.- Ralph H. Donelly writes: " Dear Maurice, I have been flattered to
---------------- *have my name appear linked to Frank Blachly's in your valuable pu-blication. The routine which I submitted and you printed in v5n4/5p25 was developed specifically to solve the problem you mention in v6n3p10. Since most of my programs are for coordinate geometry, much use is made of the R/P capability which gives the negative angle in what we surveyors call the fourth quadrant. After every INV P/R, add 360 to the angle and then follow it with the program which you have impeached and you will get a positive answer less than 360. Don't worry about changing D.dd because that is the way it will have been computed. I assume that the word <u>than</u> was omitted in your comments."*
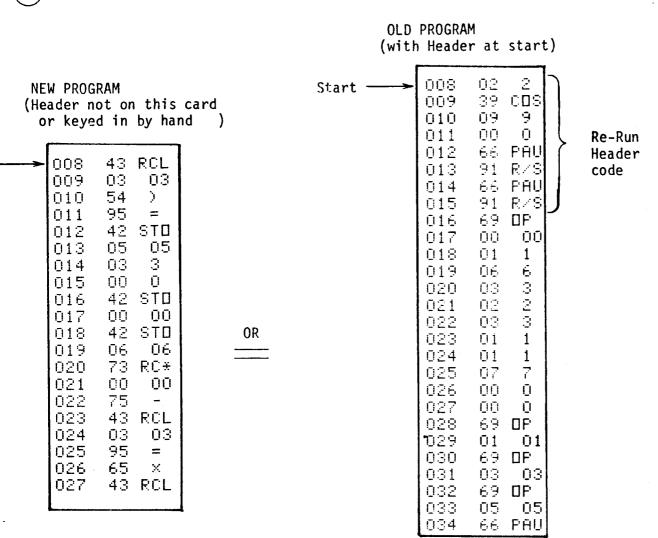
Yes, it was, unfortunately, Ralph. My apologies. ED.

LBL A PRT X:T 360 X:T P/R INV P/R CP GE STO + 360) LBL STO PRT RTN

LBL B PRT + 360 = DIV 360 = INV INT X 360 = PRT RTN

Summary: (Fast Mode)

(A) General Header must be entered & executed to secure Fast-Mode privileges:

```
000   00    0
001   00    0
002   00    0
003   00    0
004   00    0
005   36   PGM
006   02    02
007   71   SBR
008   02    02
009   39    39          ←——— RAM memory cleared;
010   09    9                      STOP  1
011   00    0
012   66   PAU
013   91   R/S          ←——— STOP 2
014   66   PAU
015   91   R/S          ←——— STOP 3
```

3 stops occur:
(cards may read
after hitting CLR)

(B) Program continues after stop 3 at step 008 in Fast-Mode

NEW PROGRAM
(Header not on this card
or keyed in by hand   )

```
008   43   RCL
009   03    03
010   54    )
011   95    =
012   42   STO
013   05    05
014   03    3
015   00    0
016   42   STO
017   00    00
018   42   STO
019   06    06
020   73   RC*
021   00    00
022   75    -
023   43   RCL
024   03    03
025   95    =
026   65    ×
027   43   RCL
```

OR

OLD PROGRAM
(with Header at start)

Start ——→

```
008   02    2
009   39   COS
010   09    9
011   00    0
012   66   PAU
013   91   R/S
014   66   PAU
015   91   R/S
016   69   OP
017   00    00
018   01    1
019   06    6
020   03    3
021   02    2
022   03    3
023   01    1
024   01    1
025   07    7
026   00    0
027   00    0
028   69   OP
029   01    01
030   69   OP
031   03    03
032   69   OP
033   05    05
034   66   PAU
```

Re-Run
Header
code

Hydrologic and Hydraulic Computations on Small Programmable
Calculators and Synthetic-Hydrograph Computations on Small
Programmable Calculators   by Dr. Thomas E. Croley II, Ph.D.

Available from: Iowa Institute of Hydraulic Research, The Univer-
sity of Iowa, Iowa City, Iowa   52242.   Price: $17.95 and
$12.00, respectively.

     Wow! What a collection of programs! Hydrologic & Hydraulic
Computations contains over 870 programs (some of the calculators
with a smaller capacity require more than one program per topic)
in 40 areas for 11 programmable calculators (HP25 thru HP97 and
SR52 thru TI59). Synthetic-Hydrograph Computations contains over
130 programs for the same machines.

     Hydrologic & Hydraulic Computations is broken into two sections.
Included in the Hydrologic Problem Section are: unit hydrograph
derivation, construction, convolution and duration transformation;
infiltration; evaporation, four routing methods; well hydraulics
(steady and unsteady radial flow); Log-Pearson Type III distribution
and fit; as well as routines for data sorting (increasing, decreasing,
or into intervals) and Fourier Analysis. Included in the Hydraulic
Problem Section are: uniform flow through various channel shapes
or circular pipe; open channel momentum and specific energy function;
hydraulic jump in rectangular channels; water surface profiles (direct
step; numerical integration and standard step); turbulent pipe flow
and the Hardy-Cross method of pipe networks.

     Synthetic-Hydrograph Computations includes overland flow
hydrographs (Izzard and kinematic wave); synthetic hydrographs
(Horton-Izzard, given runoff volume, or given inflection time);
unit hydrographs (kinematic wave, linear reservoirs cascade, desired
durations); and dimensionless hydrographs. Also included is a section
on numerical convolution.

     Each section in both books includes a good background of the
topic, starting with common equations and quickly develops the theory
and equations required so that the assumptions, limitations and
requirements of each program are understood. User directions and
program listings (unfortunately typed, not from the printers when
available) are given separately for each machine as the directions
differ slightly for the various calculators. Also included are
worked-out examples (several per section), references and comments
on the programs (limitations, quirks, etc.).

     The programs for the TI58 and TI59 are fairly basic since they
are also written for the other calculators. A few programs on
similar topics may be combined (example: uniform flow in 4 channel
types) by the user for his convenience. It is also up to the user
to add any alphanumerics he desires since none are in the listings
provided.

     These books are well written and contain a wealth of information
that should prove useful to anyone dealing with hydraulics or hydro-
logy.

Review by Frank Blachly.

ACCOUNTS.- Bill Carpenter of Bakersfield CA, reworked Dick Blayney's Talley Sheet
--------   from v6n3p5, such that it has become one of the most user-friendly pro-
grams you can imagine. It allows to enter up to 99 accounts, update the same, record
them, read them in again, almost completely automatic.

   Key in the program and record it on one card side, which you mark "1", in
6 OP 17 partitioning. (normal turn-on.)
   User instructions:
1. Read in the program by pressing CLR and sliding card side 1 into the slot.
2. Key in the account number, 1 through 99, and press A. See the account # in the
   display.
3. Enter the amount you want to credit to that account and press B.
   See the account number printed, followed by the amount you just entered, followed
   by the total in that account. You will see two different print-outs once you have
   called the same account two or more times.
4. Repeat 2 and 3 as many times as needed to enter all your accounts and the amounts
   corresponding to them.
   In case of error, enter the account number and press A. Enter the amount and press
   +/- B. That will subtract the erroneous amount. Now enter the account number again,
   and press A. Then enter the correct amount and press B.
   You may call the same account as many times during an entry session as you want.
5. When the session is over, press E. You will get a complete listing as shown in the
   example. Only those accounts that have an amount in them will be listed. The lis-
   ting should be interpreted as follows:
   example: 1.003 means account number 1 with three entries.
         1298.50 is the total amount in that account.
   After the listing, a "1" will be printed, inviting you to slide side 1 of the
   card in the slot. It will record your program again, BUT ALSO REGISTERS 90 THROUGH
   99. Then a "2" will be printed, prompting you to enter side 2. When that is done,
   a "3" is printed. Take a new card and slide in side 3. Lastly, a "4" is printed
   inviting you to slide side 4 into the slot. Program stops with 459.79 in the display.
6. For a total wipe-out at the end of the fiscal year, just press D'.
   It will leave only your program on side 1.

   For a new entry session, read in sides 1 through 4 and go to 2 and 3 above.
   Note also that at the end of the listing, a grand total is printed.

```
000  58 FIX    032  32 X:T    063  16 A'     094  42 STO    125  98 ADV       1.          2.
001  00 00     033  58 FIX    064  18 C'     095  00 00     126  98 ADV    1235.12     2000.00
002  06  6     034  03 03     065  43 RCL    096  18 C'     127  98 ADV    1235.12     7569.12
003  69 OP     035  52 EE     066  00 00     097  69 OP     128  71 SBR
004  17 17     036  22 INV    067  65 +      098  20 20     129  00 00                    99.
005  32 X:T    037  52 EE     068  53 (      099  01  1     130  00 00       2.         2000.00
006  58 FIX    038  29 CP     069  73 RC*    100  00  0     131  58 FIX   4569.12       3254.10
007  02 02     039  99 PRT    070  00 00     101  00  0     132  00 00    4569.12
008  92 RTN    040  77 GE     071  65 +      102  32 X:T    133  01  1
009  76 LBL    041  00 00     072  01  1     103  43 RCL    134  99 PRT      2.           1.001
010  18 C'     042  47 47     073  00  0     104  00 00     135  96 WRT   4569.12       1235.12
011  58 FIX    043  75 -      074  00  0     105  77 GE     136  02  2    4569.12
012  00 00     044  61 GTO    075  54 )      106  01 01     137  99 PRT                   2.003
013  01  1     045  00 00     076  22 INV    107  22 23     138  96 WRT      2.         7569.12
014  00  0     046  48 48     077  59 INT    108  29 CP     139  03  3    1000.00
015  69 OP     047  85 +      078  95 =      109  73 RC*    140  99 PRT   5569.12         3.001
016  17 17     048  01  1     079  58 FIX    110  00 00     141  96 WRT                 4569.12
017  92 RTN    049  52 EE     080  03 03     111  67 EQ     142  04  4      36.
018  76 LBL    050  05  5     081  99 PRT    112  00 00     143  99 PRT   2369.23        36.001
019  11 A      051  94 +/-    082  73 RC*    113  37 37     144  96 WRT   2369.23       2369.23
020  42 STO    052  95 =      083  00 00     114  71 SBR    145  98 ADV
021  00 00     053  74 SM*    084  58 FIX    115  00 00     146  98 ADV      98.          98.001
022  58 FIX    054  00 00     085  02 02     116  65 65     147  98 ADV   2364.45       2364.45
023  00 00     055  25 CLR    086  99 PRT    117  82 HIR    148  98 ADV   2364.45
024  92 RTN    056  73 RC*    087  98 ADV    118  38 38     149  81 RST                  99.002
025  76 LBL    057  00 00     088  92 RTN    119  61 GTO    150  76 LBL      99.         3254.10
026  12 B      058  99 PRT    089  76 LBL    120  00 00     151  19 D'    1254.10
027  32 X:T    059  98 ADV    090  15 E      121  37 37     152  18 C'    1254.10       21361.14
028  18 C'     060  32 X:T    091  25 CLR    122  32 X:T    153  47 CMS
029  43 RCL    061  81 RST    092  69 OP     123  18 18     154  81 RST
030  00 00     062  76 LBL    093  00 00     124  99 PRT
031  99 PRT
```

CHECKSUM-NUMBER ROUTINE ASSURES CORRECT PROGRAM ENTRY.- This program appeared in
--- *Electronic Design, May 28,*
*1981, pp133-135. It was written by Colin Gyles, a senior engineer at Data Precision
Corp.in Danvers, MA, 01923. Although the program seems to be OK, the author doesn't
seem to know how to handle 8's or 9's in the p-digit position of an octet. So, he
recommends checking those locations manually. Does anybody have a better way?*

DATA STORAGE AND FUNCTION PLOTTER.- This super program was written by Lonnie Mount,
-------------------------------- of Technical Calculator Programs in Riverside,
California. This program consists of two sub-programs:

Sequential Data Storage Sub-Program:
    1. Stores input data into sequentially ascending data registers beginning at
       any register desired.
    2. Prompts for data by sequentially ascending numbers beginning with item 1.
    3. Pauses after each data entry to indicate register number in which last data
       input was stored.
    4. Uses register 00, one merged label and no flags nor absolute addresses.

Function Plotting Sub-Program:
    1. Will plot curves generated by a user-supplied routine that will yield a y-
       value for an input x-value. User routines may be entered manually, 284
       steps maximum or by magnetic card, 240 steps maximum.
    2. Program uses merged lables, leaving the non-merged ones free for the user
       to employ in routines. User program uses no flags nor absolute addresses
       and should run with any user-routine not using registers 50 through 59.
       Program may be placed in either bank 1 or bank 2.
    3. Plots curves either upright or inverted on a 19 line scale. Program is able
       to provide a mirror plot of either the upright or the inverted plot.
       If the curve runs off the plot, the program stops and indicates which range
       number has been exceeded.
    4. Program locates absolute zero when it falls within the plot and brackets
       the plotted curve with a y-scale of 20 points at both the beginning and the
       end of the plot. The user may specify an arbitrary reference line within the
       plot itself.
    5. The program prints the beginning address of the plotted curve, the value of
       the y-increment and the x-increment.

As noted above, the programs contain merged labels, which enable them to run with al-
most any user-supplied program or routine, with no conflict of labels whatsoever.
Because of their very nature, merged labels are seldom, if ever, used in a normal pro-
gramming. The merged labels used here are those associated with indirect functions.
The procedure for program duplication of merged labels is shown below:

TO DUPLICATE: PRESS:

LBL ST* .... 2nd LBL SST BST STO 2nd IND
GTO ST* .... GTO SST BST STO 2nd IND
LBL EX* .... 2nd LBL SST BST 2nd EXC IND
SBR PG* .... SBR SST BST 2nd PGM IND

This same technique is used for LBL PG*, LBL RC*, SBR RC*, SBR EX*, LBL PG*, LBL PD*,
GTO PD* and LBL SM*. Two merged lables need special attention:

GE SM* .... 2nd X>T SST BST SUM 2nd IND
GE DEL .... STO 56 BST BST X>T SST

The variations between the four plotting programs are minor. It is suggested that
Plot 1 be keyed into the program memory and recorded on a mag card. Plots 2, 3 and 4
are then created by revising Plot 1. This will require you to enter the merged labels
only once. The program does not contain absolute addresses, so don't be afraid to
delete and insert steps when creating Plots 2, 3 and 4 out of Plot 1.

User Function Routine Requirements:
    1. User routine must not use registers 50 through 59.
    2. User routine must not exceed 284 steps. Routines that exceed 240 steps
       should begin at step 000. The first 240 steps may be entered by means of a
       mag card. Program steps between 240 and 283 must be entered manually by
       overwriting the sequential data storage sub-program, LBL STO, which is
       immediately in front of the plotter program, LBL R/S.
    3. User-supplied routines must not use the common labels ST nor R/S. Neither
       must they use merged labels nor absolute addresses to program locations
       greater than 283.
    4. User routines should end in RTN. (INV SBR)

Data Storage and Function plotter (cont.)

>   5. For best results, the user-supplied routine should not do any printing of
>   its own.

Plotter Program Requirements:

The plotter program itself requires only seven items of input data and a user-supplied routine before it can be run. The seven items of input data are:

Item # 1, R50   A three digit number which locates the beginning step of the user-supplied routine. (function routine)

Item # 2, R51   Upper y-range value, a number, either positive or negative. which is slightly larger than the maximum anticipated value of y, calculated by the function routine.

Item # 3, R52   Lower y-range value, a number, either positive or negative, which is slightly smaller than the anticipated minimum value of y, calculated by the function routine.

Item # 4, R53   Y-value of a user-specified x-reference line. If entered value of y is not within the range of the plot, the reference line will be omitted.

Item # 5, R54   X-value at the start of the plot, minimum valye of x.

Item # 6, R55   X-value at the end of the plot, maximum value of x.

Item # 7, R56   Plotting increment of x, which must be positive.

Sample problem: Beginning at step 036, enter the function $f(x) = 4 \sin (x^2)$
Plot f(x) between 0 and 30 degrees at 1 degree interval.
The x-reference is not to appear in the first four plots. However, repeat plot 4 with a zero-reference line.

Key in as follows:

> 50 SBR STO, which initializes the sequential data storage.
> 36 R/S, tells the program the beginning step of the function routine.
> 5 R/S, upper y-rabge value
> 5 +/- R/S, lower y-range value.
> 6 R/S, puts the reference out of the range of 5 to -5 and thus will not appear. In the repeated plot 4, enter a zero here.
> 0 R/S, the beginning x-value of f(x)
> 30 R/S, the end x-value of f(x)
> 1 R/S, x-plotting increment. In the repeated plot 4, use a 2.
> SBR   R/S

SEE THE FIVE PLOTTED CURVES AND THE INPUT ROUTINE SAMPLE ON NEXT PAGE.

PLOT 1. Listing.

240	00	0	275	00	00	310	06	6	345	91	R/S	379	55	÷	413	43	RCL	447	01	1
241	00	0	276	43	RCL	311	05	5	346	69	OP	380	01	1	414	54	54	448	52	EE
242	00	0	277	00	00	312	06	6	347	00	00	381	09	9	415	42	STO	449	08	8
243	00	0	278	59	INT	313	05	5	348	03	3	382	95	=	416	57	57	450	94	+/-
244	00	0	279	66	PAU	314	69	OP	349	03	3	383	42	STO	417	76	LBL	451	95	=
245	00	0	280	69	OP	315	01	01	350	02	2	384	58	58	418	64	PD*	452	22	INV
246	00	0	281	20	20	316	69	OP	351	07	7	385	50	I<I	419	71	SBR	453	52	EE
247	00	0	282	61	GTO	317	02	02	352	03	3	386	99	PRT	420	40	IND	454	22	INV
248	00	0	283	72	ST*	318	69	OP	353	02	2	387	04	4	421	50	50	455	77	GE
249	76	LBL	284	76	LBL	319	03	03	354	03	3	388	04	4	422	32	X:T	456	74	SM*
250	42	STO	285	63	EX*	320	69	OP	355	07	7	389	00	0	423	43	PCL	457	43	PCL
251	59	INT	286	75	-	321	04	04	356	00	0	390	00	0	424	51	51	458	59	59
252	85	+	287	43	PCL	322	69	OP	357	02	2	391	22	INV	425	22	INV	459	69	OP
253	35	1/X	288	52	52	323	05	05	358	69	OP	392	01	01	426	77	GE	460	07	07
254	55	÷	289	95	=	324	69	OP	359	02	02	393	71	SBR	427	56	DEL	461	24	CE
255	02	2	290	55	-	325	00	00	360	69	OP	394	73	PC*	428	43	RCL	462	32	X:T
256	95	=	291	43	PCL	326	92	RTN	361	05	05	395	43	PCL	429	52	52	463	61	GTO
257	42	STO	292	58	58	327	76	LBL	362	43	RCL	396	56	56	430	77	GE	464	64	PD*
258	00	00	293	95	=	328	73	PC+	363	50	50	397	99	PRT	431	56	DEL	465	76	LBL
259	76	LBL	294	58	FIX	329	03	3	364	99	PRT	398	43	PCL	432	32	X:T	466	74	SM*
260	72	ST+	295	00	00	330	06	6	365	04	4	399	53	53	433	63	EX*	467	71	SBR
261	43	PCL	296	52	EE	331	01	1	366	05	5	400	71	SBR	434	63	EX*	468	62	PG*
262	00	00	297	22	INV	332	05	5	367	00	0	401	63	EX*	435	69	OP	469	25	CLR
263	75	-	298	52	EE	333	01	1	368	00	0	402	42	STO	436	07	07	470	71	SBR
264	22	INV	299	22	INV	334	03	3	369	69	OP	403	59	59	437	43	PCL	471	63	EX*
265	59	INT	300	58	FIX	335	02	2	370	01	01	404	25	CLR	438	56	56	472	69	OP
266	35	1/X	301	92	RTN	336	07	7	371	71	SBR	405	98	ADV	439	44	SUM	473	07	07
267	55	-	302	76	LBL	337	01	1	372	73	PC*	406	71	SBR	440	57	57	474	25	CLR
268	02	2	303	62	PG*	338	07	7	373	43	RCL	407	63	EX*	441	43	PCL	475	98	ADV
269	85	+	304	06	6	339	69	OP	374	51	51	408	69	OP	442	57	57	476	98	ADV
270	01	1	305	05	5	340	02	02	375	75	-	409	07	07	443	32	X:T	477	98	ADV
271	95	=	306	06	6	341	69	OP	376	43	PCL	410	24	CE	444	43	PCL	478	98	ADV
272	59	INT	307	05	5	342	69	OP	377	52	52	411	71	SBR	445	55	55	479	92	RTN
273	31	R/3	308	06	6	343	92	RTN	378	95	=	412	62	PG*	446	95	+			
274	72	ST*	309	05	5	344	76	LBL												

SAMPLE PLOTTINGS.

PLOT1
```
036   33 X²
037   38 SIN
038   65 ×
039   04 4
040   95 =
041   92 RTN

PLOT1
        36.
Y SCALE
  .5263157895
X SCALE
         1.
```

PLOT2
```
036   33 X²
037   38 SIN
038   65 ×
039   04 4
040   95 =
041   92 RTN

PLOT2
        36.
Y SCALE
  .5263157895
X SCALE
         1.
```

PLOT3
```
036   33 X²
037   38 SIN
038   65 ×
039   04 4
040   95 =
041   92 RTN

PLOT3
        36.
Y SCALE
  .5263157895
X SCALE
         1.
```

PLOT4
```
036   33 X²
037   38 SIN
038   65 ×
039   04 4
040   95 =
041   92 RTN

PLOT4
        36.
Y SCALE
  .5263157895
X SCALE
         1.
```

PLOT4
```
036   33 X²
037   38 SIN
038   65 ×
039   04 4
040   95 =
041   92 RTN

PLOT4
        36.
Y SCALE
  .5263157895
X SCALE
         2.
```

PLOT 2, listing.

```
240 00  0      275 00  00     310 06  6      345 91 R/S     379 55  ÷      413 43 RCL     447 01  1
241 00  0      276 43 RCL     311 05  5      346 69 OP      380 01  1      414 54  54     448 52 EE
242 00  0      277 00  00     312 06  6      347 00  00     381 09  9      415 42 STO     449 08  8
243 00  0      278 59 INT     313 05  5      348 03  3      382 95  =      416 57  57     450 94 +/-
244 00  0      279 66 PAU     314 69 OP      349 03  3      383 42 STO     417 76 LBL     451 95  =
245 00  0      280 69 OP      315 01  01     350 02  2      384 58  58     418 64 PD*     452 22 INV
246 00  0      281 20  20     316 69 OP      351 07  7      385 50 I×I     419 71 SBR     453 52 EE
247 00  0      282 61 GTO     317 02  02     352 03  3      386 99 PRT     420 40 IND     454 22 INV
248 00  0      283 72 ST*     318 69 OP      353 02  2      387 04  4      421 50  50     455 77 GE
249 76 LBL     284 76 LBL     319 03  03     354 03  3      388 04  4      422 32 X:T     456 74 SM*
250 42 STO     285 63 EX*     320 69 OP      355 07  7      389 00  0      423 43 RCL     457 43 RCL
251 59 INT     286 75  -      321 04  04     356 00  0      390 00  0      424 51  51     458 59  59
252 85  +      287 43 RCL     322 69 OP      357 03  3      391 69 OP      425 22 INV     459 69 OP
253 35 1/X     288 51  51     323 05  05     358 69 OP      392 01  01     426 77 GE      460 07  07
254 55  ÷      289 95  =      324 69 OP      359 02  02     393 71 SBR     427 56 DEL     461 24 CE
255 02  2      290 55  +      325 00  00     360 69 OP      394 73 RC*     428 43 RCL     462 32 X:T
256 95  =      291 43 RCL     326 92 RTN     361 05  05     395 43 RCL     429 52  52     463 61 GTO
257 42 STO     292 58  58     327 76 LBL     362 43 RCL     396 56  56     430 77 GE      464 64 PD*
258 00  00     293 95  =      328 73 RC*     363 50  50     397 99 PRT     431 56 DEL     465 76 LBL
259 76 LBL     294 58 FIX     329 03  3      364 99 PRT     398 43 RCL     432 32 X:T     466 74 SM*
260 72 ST*     295 00  00     330 06  6      365 04  4      399 53  53     433 71 SBR     467 71 SBR
261 43 RCL     296 52 EE      331 01  1      366 05  5      400 71 SBR     434 63 EX*     468 62 PG*
262 00  00     297 22 INV     332 05  5      367 00  0      401 63 EX*     435 69 OP      469 25 CLR
263 75  -      298 52 EE      333 01  1      368 00  0      402 42 STO     436 07  07     470 71 SBR
264 22 INV     299 22 INV     334 03  3      369 69 OP      403 59  59     437 43 RCL     471 63 EX*
265 59 INT     300 58 FIX     335 02  2      370 01  01     404 25 CLR     438 56  56     472 69 OP
266 35 1/X     301 92 RTN     336 07  7      371 71 SBR     405 98 ADV     439 44 SUM     473 07  07
267 55  +      302 76 LBL     337 01  1      372 73 RC*     406 71 SBR     440 57  57     474 25 CLR
268 02  2      303 62 PG*     338 07  7      373 43 RCL     407 63 EX*     441 43 RCL     475 98 ADV
269 85  +      304 06  6      339 69 OP      374 52  52     408 69 OP      442 57  57     476 98 ADV
270 01  1      305 05  5      340 02  02     375 75  -      409 07  07     443 32 X:T     477 98 ADV
271 95  =      306 06  6      341 69 OP      376 43 RCL     410 24 CE      444 43 RCL     478 98 ADV
272 59 INT     307 05  5      342 05  05     377 51  51     411 71 SBR     445 55  55     479 92 RTN
273 91 R/S     308 06  6      343 92 RTN     378 95  =      412 62 PG*     446 85  +
274 72 ST*     309 05  5      344 76 LBL
```

PLOT 3, listing.

```
240 00  0      275 00  00     310 06  6      345 91 R/S     379 55  ÷      413 43 RCL     447 75  -
241 00  0      276 43 RCL     311 05  5      346 69 OP      380 01  1      414 55  55     448 01  1
242 00  0      277 00  00     312 06  6      347 00  00     381 09  9      415 42 STO     449 52 EE
243 00  0      278 59 INT     313 05  5      348 03  3      382 95  =      416 57  57     450 08. 8
244 00  0      279 66 PAU     314 69 OP      349 03  3      383 42 STO     417 76 LBL     451 94 +/-
245 00  0      280 69 OP      315 01  01     350 02  2      384 58  58     418 64 PD*     452 95  =
246 00  0      281 20  20     316 69 OP      351 07  7      385 50 I×I     419 71 SBR     453 22 INV
247 00  0      282 61 GTO     317 02  02     352 03  3      386 99 PRT     420 40 IND     454 52 EE
248 00  0.     283 72 ST*     318 69 OP      353 02  2      387 04  4      421 50  50     455 77 GE
249 76 LBL     284 76 LBL     319 03  03     354 03  3      388 04  4      422 32 X:T     456 74 SM*
250 42 STO     285 63 EX*     320 69 OP      355 07  7      389 00  0      423 43 RCL     457 43 RCL
251 59 INT     286 75  -      321 04  04     356 00  0      390 00  0      424 51  51     458 59  59
252 85  +      287 43 RCL     322 69 OP      357 04  4      391 69 OP      425 22 INV     459 69 OP
253 35 1/X     288 52  52     323 05  05     358 69 OP      392 01  01     426 77 GE      460 07  07
254 55  ÷      289 95  =      324 69 OP      359 02  02     393 71 SBR     427 56 DEL     461 24 CE
255 02  2      290 55  ÷      325 00  00     360 69 OP      394 73 RC*     428 43 RCL     462 32 X:T
256 95  =      291 43 RCL     326 92 RTN     361 05  05     395 43 RCL     429 52  52     463 61 GTO
257 42 STO     292 58  58     327 76 LBL     362 43 PCL     396 56  56     430 77 GE      464 64 PD*
258 00  00     293 95  =      328 73 RC*     363 50  50     397 99 PRT     431 56 DEL     465 76 LBL
259 76 LBL     294 58 FIX     329 03  3      364 99 PRT     398 43 RCL     432 32 X:T     466 74 SM*
260 72 ST*     295 00  00     330 06  6      365 04  4      399 53  53     433 71 SBR     467 71 SBR
261 43 RCL     296 52 EE      331 01  1      366 05  5      400 71 SBR     434 63 EX*     468 62 PG*
262 00  00     297 22 INV     332 05  5      367 00  0      401 63 EX*     435 69 OP      469 25 CLR
263 75  -      298 52 EE      333 01  1      368 00  0      402 42 STO     436 07  07     470 71 SBR
264 22 INV     299 22 INV     334 03  3      369 69 OP      403 59  59     437 43 RCL     471 63 EX*
265 59 INT     300 58 FIX     335 02  2      370 01  01     404 25 CLR     438 56  56     472 69 OP
266 35 1/X     301 92 RTN     336 07  7      371 71 SBR     405 98 ADV     439 94 +/-     473 07  07
267 55  ÷      302 76 LBL     337 01  1      372 73 RC*     406 71 SBR     440 44 SUM     474 25 CLR
268 02  2      303 62 PG*     338 07  7      373 43 PCL     407 63 EX*     441 57  57     475 98 ADV
269 85  +      304 06  6      339 69 OP      374 51  51     408 69 OP      442 43 RCL     476 98 ADV
270 01  1      305 05  5      340 02  02     375 75  -      409 07  07     443 57  57     477 98 ADV
271 95  =      306 06  6      341 69 OP      376 43 RCL     410 24 CE      444 32 X:T     478 98 ADV
272 59 INT     307 05  5      342 05  05     377 52  52     411 71 SBR     445 43 RCL     479 92 RTN
273 91 R/S     308 06  6      343 92 RTN     378 95  =      412 62 PG*     446 54  54
274 72 ST*     309 05  5      344 76 LBL
```

PLOT 4, listing.

```
240 00  0      275 00  00     310 06  6      345 91 R/S     379 55  ÷      413 43 RCL     447 75  -
241 00  0      276 43 RCL     311 05  5      346 69 OP      380 01  1      414 55  55     448 01  1
242 00  0      277 00  00     312 06  6      347 00  00     381 09  9      415 42 STO     449 52 EE
243 00  0      278 59 INT     313 05  5      348 03  3      382 95  =      416 57  57     450 08  8
244 00  0      279 66 PAU     314 69 OP      349 03  3      383 42 STO     417 76 LBL     451 94 +/-
245 00  0      280 69 OP      315 01  01     350 02  2      384 58  58     418 64 PD*     452 95  =
246 00  0      281 20  20     316 69 OP      351 07  7      385 50 I×I     419 71 SBR     453 22 INV
247 00  0      282 61 GTO     317 02  02     352 03  3      386 99 PRT     420 40 IND     454 52 EE
248 00  0      283 72 ST*     318 69 OP      353 02  2      387 04  4      421 50  50     455 77 GE
249 76 LBL     284 76 LBL     319 03  03     354 03  3      388 04  4      422 32 X:T     456 74 SM*
250 42 STO     285 63 EX*     320 69 OP      355 07  7      389 00  0      423 43 RCL     457 43 RCL
251 59 INT     286 75  -      321 04  04     356 05  5      390 00  0      424 51  51     458 59  59
252 85  +      287 43 RCL     322 69 OP      357 05  5      391 69 OP      425 22 INV     459 69 OP
253 35 1/X     288 51  51     323 05  05     358 69 OP      392 01  01     426 77 GE      460 07  07
254 55  ÷      289 95  =      324 69 OP      359 02  02     393 71 SBR     427 56 DEL     461 24 CE
255 02  2      290 55  ÷      325 00  00     360 69 OP      394 73 RC*     428 43 RCL     462 32 X:T
256 95  =      291 43 RCL     326 92 RTN     361 05  05     395 43 RCL     429 52  52     463 61 GTO
257 42 STO     292 58  58     327 76 LBL     362 43 RCL     396 56  56     430 77 GE      464 64 PD*
258 00  00     293 95  =      328 73 RC*     363 50  50     397 99 PRT     431 56 DEL     465 76 LBL
259 76 LBL     294 58 FIX     329 03  3      364 99 PRT     398 43 RCL     432 32 X:T     466 74 SM*
260 72 ST*     295 00  00     330 06  6      365 04  4      399 53  53     433 71 SBR     467 71 SBR
261 43 RCL     296 52 EE      331 01  1      366 05  5      400 71 SBR     434 63 EX*     468 62 PG*
262 00  00     297 22 INV     332 05  5      367 00  0      401 63 EX*     435 69 OP      469 25 CLR
263 75  -      298 52 EE      333 01  1      368 00  0      402 42 STO     436 07  07     470 71 SBR
264 22 INV     299 22 INV     334 03  3      369 69 OP      403 59  59     437 43 RCL     471 63 EX*
265 59 INT     300 58 FIX     335 02  2      370 01  01     404 25 CLR     438 56  56     472 69 OP
266 35 1/X     301 92 RTN     336 07  7      371 71 SBR     405 98 ADV     439 94 +/-     473 07  07
267 55  ÷      302 76 LBL     337 01  1      372 73 RC*     406 71 SBR     440 44 SUM     474 25 CLR
268 02  2      303 62 PG*     338 07  7      373 43 RCL     407 63 EX*     441 57  57     475 98 ADV
269 85  +      304 06  6      339 69 OP      374 52  52     408 69 OP      442 43 RCL     476 98 ADV
270 01  1      305 05  5      340 02  02     375 75  -      409 07  07     443 57  57     477 98 ADV
271 95  =      306 06  6      341 69 OP      376 43 RCL     410 24 CE      444 32 X:T     478 98 ADV
272 59 INT     307 05  5      342 05  05     377 51  51     411 71 SBR     445 43 RCL     479 92 RTN
273 91 R/S     308 06  6      343 92 RTN     378 95  =      412 62 PG*     446 54  54
274 72 ST*     309 05  5      344 76 LBL
```

----

*COMPUTERS IN MEDICINE AND BIOLOGY.- This journal is looking for good articles using
------------------------------ TI-59 programs related to statistics, optics,
physiology, stress analysis, etc. with a potential use to biologists, physiologists,
dentists and physicians, such as reducing errors in lab procedure.*

    *Please contact TIPPC Club member Dr. Cliff Lieberman, 1020 N.Quincy St.,
Apt. 1006, Arlington, VA, 22201, USA. Cliff is willing to help you in any way he can
to make your article a success.*

    *If you get one published, please tell the TI PPC NOTES about it. I would
like to inform our members about it. Thank you.*

THE POEM MACHINE II.- We recently acquired the youngest member of the club, 13-year
------------------- old Michael Malik. I suspect Michael did not get his exalted
state of calculator fanaticism by pure chance. His father is a fellow scientist of
John Worthington and Emil Regelman at EPA. Whatever the "aggravating circumstances",
Michael is a fine programmer. Among the several letters I received from him a found
a new version of the famous POEM MACHINE. Michael's is not version II in reality, as
about a year ago I had a rather lively exchange of different versions with the Snow
brothers.(which program didn't they write?) We ended up with three basic versions:
one suitable for a party at which you could invite the preacher, rabbi or priest,
(it was clean enough to invite even mother superior) one suitable for a party at
which the first three might come(but I would definitely leave out mother superior)
and a last version good for a party at which I would only invite my bachelor friends.

Well, Michael's program is of the first variety. It does not rely on the ML
module, as it contains its own random number generator. Both reviewers told me to
publish it, even when they didn't know Michael's age.

The instructions are simple:

1. *Put your calculator in 10 OP 17 partitioning and key in both program steps and*
   data registers.
2. *Put your calculator in 6 OP 17 partitioning and record four card sides.*

To use the program:

1. *Read in four card sides in 6 OP 17, turn-on partitioning. Program repartitions*
   automatically to 10 OP 17.
2. *Enter any seed and press A. The printer will print lovely nonesense verses, free*
   non-rhyming lines of California-style poetry.

POEMS

PAT HIS SICK DOG

KISS YOUR LAZY PIG

LOVE YOUR GRIMY MULE

SELL OUR HAIRY NOSE

BUY MY SOFT HAND

BAKE ONE SLIMY APE

MISS THE PUNY TOE

GRAB A SHY EGG

HELP SOME SUPER BEER

TAKE SOME SWEET COW

BEAT THAT SWELL RAG

EAT HER WILD BIKE

RUB HIS DIRTY DOG

FEEL YOUR LAZY CAR

LOVE OUR EVIL BUG

SELL MY UGLY BIRD

BUY MY CRAZY HAND

PITY ONE SLIMY TOAD

HAIL THE LARGE LEG

GET A NEAT EGG

THE POEM MACHINE II

000	76 LBL	041	17 17	082	42 STO
001	17 B'	042	69 OP	083	00 00
002	89 π	043	00 00	084	73 RC*
003	22 INV	044	03 3	085	00 00
004	59 INT	045	03 3	086	69 OP
005	82 HIR	046	03 3	087	02 02
006	34 34	047	02 2	088	17 B'
007	53 (	048	01 1	089	65 ×
008	02 2	049	07 7	090	03 3
009	04 4	050	03 3	091	02 2
010	02 2	051	00 0	092	35 +
011	09 9	052	03 3	093	02 2
012	08 8	053	06 6	094	06 6
013	65 ×	054	69 OP	095	95 =
014	82 HIR	055	01 01	096	59 INT
015	14 14	056	69 OP	097	42 STO
016	85 +	057	05 05	098	00 00
017	09 9	058	98 ADV	099	73 RC*
018	09 9	059	17 B'	100	00 00
019	09 9	060	65 ×	101	69 OP
020	09 9	061	02 2	102	03 03
021	01 1	062	05 5	103	17 B'
022	54)	063	85 +	104	65 ×
023	55 ÷	064	01 1	105	03 3
024	01 1	065	95 =	106	02 2
025	09 9	066	59 INT	107	85 +
026	09 9	067	42 STO	108	05 5
027	00 0	068	00 00	109	08 8
028	01 1	069	73 RC*	110	95 =
029	07 7	070	00 00	111	59 INT
030	95 =	071	69 OP	112	42 STO
031	22 INV	072	01 01	113	00 00
032	59 INT	073	17 B'	114	73 RC*
033	92 RTN	074	65 ×	115	00 00
034	76 LBL	075	01 1	116	69 OP
035	11 A	076	00 0	117	04 04
036	82 HIR	077	85 +	118	69 OP
037	04 04	078	09 9	119	05 05
038	01 1	079	00 0	120	61 GTO
039	00 0	080	95 =	121	00 00
040	69 OP	081	59 INT	122	58 58

23412200.	01
2732421700.	02
2332271600.	03
2117172700.	04
2624363600.	05
35411400.	06
33133700.	07
17133700.	08
36171700.	09
1417133700.	10
4313362300.	11
3713261700.	12
36171700.	13
2617173300.	14
2317273300.	15
22173700.	16
2235131400.	17
3113242700.	18
3024363600.	19
3324374500.	20
1413261700.	21
4313353000.	22
14414500.	23
4313272600.	24
3617272700.	25
3641331735.	26
3335244617.	27
3117133700.	28
36234500.	29
3124151700.	30
2713352217.	31
3341314500.	32
3624272745.	33
3627243045.	34
3631321400.	35
1535134645.	36
3632213700.	37
14242200.	38
4122274500.	39
2313243545.	40
2232321600.	41
1742242700.	42
2235243045.	43
1535134645.	44
2713464500.	45
32271600.	46
1624353745.	47
3624152600.	48
3615413045.	49

4324271600.	50
21133700.	51
4324161700.	52
3643172727.	53
1641301400.	54
3643171737.	55
4317243516.	56
3327413033.	57
13331700.	58
15133700.	59
23133116.	60
43323530.	61
14243516.	62
31323617.	63
36243126.	64
14412200.	65
30412717.	66
21353222.	67
15133500.	68
33242200.	69
22321337.	70
16322200.	71
26243122.	72
14242617.	73
22311337.	74
21271713.	75
35132200.	76
21243623.	77
31323617.	78
15324300.	79
27243231.	80
45132600.	81
14171735.	82
15353243.	83
17222200.	84
23322200.	85
27172200.	86
37321700.	87
43243117.	88
37321316.	89
23173500.	90
3723133700.	91
3632301700.	92
130000.	93
37231700.	94
32311700.	95
30450000.	96
32413500.	97
453241350Û.	98
23243600.	99

PRINT CODE PROCESSOR
by John Worthington and Emil Regelman

This program will permit the convenient storage and manipulation of print code data, up to a total of 14 lines.

Initialization

Press [E'] to clear all previously stored data, and initialize appropriate memories. The 1.01 in the display indicates that the pointer is located at the first character of the first line (L.nn).

Storage of Print Codes

1. To store print codes, enter up to 5 and press [A]. The program will store these codes appropriately. The pointer is adjusted to indicate the next available location. (This operation takes about 9 seconds). Repeat, as desired.

2. When a line of text is completed it is automatically printed.

3. To end a line before 20 characters have been entered, press [A'].

4. To relocate the pointer, specify the line and character position desired (L.nn), and press [E].

5. To print the stored data, press [B]. [B] will only print data up to and including the line indicated by the pointer.

Error Correction

1. Move the pointer to the location of the error as in #4, above.

2. To write over existing text, enter up to 5 print codes and press [A].

   To insert print codes into the existing text, press [C]. (If there is a large amount of text after the insertion this operation may take several minutes!)

   To delete one or more characters, enter the number of deletions desired and press [D]. If zero is in the display, only one pair of codes will be deleted. (If there are a large number of deletions, this operation may take several minutes!)

PRINT CODE PROCESSOR

Loc	Code	Key	Loc	Code	Key	Loc	Code	Key	Loc	Code	Key	Loc	Code	Key	Loc	Code	Key	Loc	Code	Key	Loc	Code	Key
000	76	LBL	060	86	STF	120	52	EE	180	00	0	240	87	IFF	300	42	STO	360	74	SM*	420	04	04
001	17	B′	061	01	01	121	55	÷	181	54	)	241	01	01	301	02	02	361	00	00	421	46	+

PARABOLIC CURVE FIT.- Bob Patton, Arlington, Texas, wrote this program. It does a
------------------ least squares fit to a parabola according to:

$$y = A + Bx + Cx^2.$$

It gathers sums and solves the following equations using PGM 02 of the ML-module:

$$nA + \Sigma xB + \Sigma x^2 C = \Sigma y$$

$$\Sigma xA + \Sigma x^2 B + \Sigma x^3 C = \Sigma xy$$

$$\Sigma x^2 A + \Sigma x^3 B + \Sigma x^4 C = \Sigma x^2 y$$

and the correlation coefficient

$$r^2 = [\ B(\Sigma xy - 1/n\ \Sigma x\Sigma y) + C(\Sigma x^2 y - 1/n\Sigma x\Sigma y)]\ /\ [\Sigma y^2 - 1/n\ (\Sigma y)^2]$$

Recording instructions:
1. Put your calculator in 8 OP 17 partitioning and key in program and data registers.
2. Put the calculator in 6 OP 17 partitioning and record banks 1 and 2 on a mag card.

User instructions:
1. Load sides 1 and 2.
2. Press 2nd A'. Titles are printed.
3. Enter data: x_i, press A; y_i, press B. Repeat this step for all data pairs.
4. To remove data: x_i, press A; y_i, press 2nd B'.
5. Solve determinant, press C. A, B and C printed.
6. Solve correlation factor, press D. R^2 printed.
7. Find predicted values: x, press E, see $\hat{y}$.

PRINT CODE TABLE.- P.D. Forbes of Philadelphia, PA gave me this chart at the recent
---------------   seminar in that city. His comments:"I found that entering print
information was slow enough without the added disadvantage of trying to read the chart
in the manual. Maybe the large print says something about the creeping myopia of age."
          All us older fellas thank you, Don.

	0	1	2	3	4	5	6
00	01	02	03	04	05	06	07
7	8	9	A	B	C	D	E
10	11	12	13	14	15	16	17
—	F	G	H	I	J	K	L
20	21	22	23	24	25	26	27
M	N	O	P	Q	R	S	T
30	31	32	33	34	35	36	37
•	U	V	W	X	Y	Z	+
40	41	42	43	44	45	46	47
x	*	$\sqrt{\ }$	π	$\underline{e}$	(	)	,
⁻50	51	52	53	54	55	56	57
↑	%	:	/	=	$\underline{'}$	x	$\bar{x}$
60	61	62	63	64	65	66	67
$\underline{2}$	?	÷	!	$\amalg$	△	π	Σ
70	71	72	73	74	75	76	77

NEWCOMER'S CORNER.-Subroutines. The Personal Programming manual states on page IV-47:
----------------------------- "Actually, up to six return addresses can be stored
in the subroutine return register at any one time. This means that a subroutine can
contain and use or call a subroutine that can also call a subroutine, etc. up to six
times." Then, towards the bottom of the page comes the most important part:" When a
program part is labeled with a user-defined key, that part can be executed from the
keyboard simply by pressing the applicable key as we have seen.The same thing happens
when one of these keys is encountered in a program - the program pointer goes to that
label and processing continues. These user-defined keys have an automatic SBR instruc-
tion built in. So, if you label a program part with a user-defined key and end it
with INV SBR, that part is processed just as though you had called it with a the SBR
instruction."

I get letters, sometimes up to fifty a day. Among them I find people saying:"I wrote
this program (enclosed) which does a certain routine six times. Then it goes haywire."

Or I teach TI-59 seminars and one of the more advanced students approaches me with
"Somebody at the office wrote this program. It works alright, as long as I don't
tell it to go through the loop more than six times. Otherwise it prints all kinds
of garbage."

In all of these cases you can be sure they are suffering from subroutine trouble. And
especially the kind where user-defined labels are involved.

Take, for example,this sequence:

LBL E ....ADV E (the writer told me that his common label routine always
worked alright: LBL COS ....ADV GTO COS.)

The LBL E routine executes six times and goes to never-never land. Why? Because the
call to E after the ADV is in reality a SBR E call. It is as if you nested six sub-
routines into each other. The remedy? Simply say "GTO E". That is the reason why the
common label version always worked, because it does not say "SBR COS" but "GTO COS.
When the command GTO is used, no address is stored and thus no return register can
get full. But if you say simply E, you say in effect SBR E, and each time the contents
of the subroutine return register is increased by one. When it reaches seven it lets
you know that you broke the rules and refuses to go on.

A mechanical engineer in one of the seminars had a rather involved program, which he
wrote himself. From the coding one immediately recognized an experienced programmer.
He stated he just attended the seminar "to pick up a few pointers here and there."
Ironically, the "pointers" were his downfall. After cutting through all the "hash"
his program boiled down to:

LBL COS..........GTO SIN. Then somewhere else LBL SIN..............GTO +. And finally
at the very end of his program LBL + ...SBR COS.

Again, the same complaint: the program iterates six times, then it stops and winks at
you." The SBR COS is, of course, the offending sequence. Replace it with GTO COS and
the program will iterate till the year 2000 and beyond.

On the other hand, some people believe that, by placing a RTN (INV SBR) instruction at
the end of a branch, they effectively convert it into a subroutine. Wrong!!!

The instruction following immediately after a comparison is always a branch and cannot
be converted into a subroutine by simply placing a RTN at the end:

....RCL NN X>T COS PRT R/S   LBL COS........R/S or GTO NNN or GTO some(common)label.

The COS after the comparison is a branch and, as branches on a tree, never returns to
the main tree. If you insist to make LBL COS a subroutine, go ahead and end it in RTN
But then re-write your main program as

......RCL NN INV X>T PRT SBR COS LBL PRT PRT R/S LBL COS......RTN