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

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

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

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

Members living outside North America: The special A.O. Mail Printed Matter rate we have been allowed to use is much less expensive than outright Air Mail, of course. It differs from continent to continent, in three categories, between 60 and 86 cents per issue. But, during the summer an International Post Office conference was held in Rio de Janeiro, in which it was proposed to increase all Air Mail tariffs. As an example, an ordinary 1 ounce letter would be increased from 31 to 40 cents. We will not try to anticipate the exact increase it will provoke in each of the three A.O. Mail categories. That would be an impossible task. But increase they will, and soon. The post office has already received large amounts of larger-denominations air-mail stamps. To make it simple, may I propose to those members living outside North America to send $ 10.00 extra. If, however, you don't require air mail delivery, $ 5.00 extra, or the equivalent of domestic first class, will do for surface mail.

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

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

<table>
<thead>
<tr>
<th>TABLE OF CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALENDAR PRINTING CONTEST, Palmer O. Hansen ........................... 3</td>
</tr>
<tr>
<td>SOLUTIONS OF 3 EQ IN 3 UNKNOWNS, COMPLEX, Bill Wilson ................. 3</td>
</tr>
<tr>
<td>RADAR CALCULATIONS, Bill Skillman ......................................... 4</td>
</tr>
<tr>
<td>MICROWAVE CIRCUIT DESIGN .................................................. 4</td>
</tr>
<tr>
<td>RADIATION THERAPY, Barry Tugman ........................................... 4</td>
</tr>
<tr>
<td>LAPLACE TRANSFORM, Alan Chu Mao ............................................ 4</td>
</tr>
<tr>
<td>3-D NAVIGATION, Dave Leising ............................................... 4</td>
</tr>
<tr>
<td>TIME BomBS, Richard Snow ................................................... 4</td>
</tr>
<tr>
<td>MOREMINKEREBIODIGRAM, Harold M. Oto ..................................... 4-6</td>
</tr>
<tr>
<td>SELECTIVE INH LIST, Karl-Joseph Meusel ................................. 6-8</td>
</tr>
<tr>
<td>PRATTURO DIGITS, Dave Leising ............................................. 6-8</td>
</tr>
<tr>
<td>SMPSYD FACTOR Finder, Kevin, Hastewm, Caldwell, Williams de-T ...... 7</td>
</tr>
<tr>
<td>TRANSFORMATION REGRESSION, J. Worthington &amp; B. Rayeim ............... 7-30</td>
</tr>
<tr>
<td>GAUSS DIGITS PRINT, Karl-Joseph Meusel .................................. 10</td>
</tr>
<tr>
<td>FAST ALPHA SORT, Richard Snow .............................................. 10-12</td>
</tr>
<tr>
<td>FAST MODMUV LIST, J. Worthington &amp; B. Rayeim .......................... 10-15</td>
</tr>
<tr>
<td>10-DIGIT RES List, Clyde Corbin, Richard Snow ........................... 15</td>
</tr>
<tr>
<td>STRUCTURAL PROGRAMS .......................................................... 17</td>
</tr>
<tr>
<td>DAY OF THE WEEK, Dave Beden .............................................. 17</td>
</tr>
<tr>
<td>ANALYZE COMPLEX LINEAR NETWORKS S.N. Hartman ........................... 17</td>
</tr>
<tr>
<td>PROGRAM LISTING WITH PAPER, John Gasia III ............................ 16</td>
</tr>
<tr>
<td>ML-95 SIMPSON APPROXIMATION, John Gares III ............................ 18</td>
</tr>
<tr>
<td>PRINT CODE TABLES, Dave Leising .......................................... 18-20</td>
</tr>
<tr>
<td>PROGRAMMING PUZZLEs, Vogel, Rosenale, Snow ............................. 20-25</td>
</tr>
<tr>
<td>BRAIN TOSERS, B. Gustavson &amp; C. Boal .................................... 21</td>
</tr>
<tr>
<td>ML SURVIVAL GUIDE, Wayne, Chas, Lewis ................................. 21-22</td>
</tr>
<tr>
<td>FAST MODE MUSE CODE, Dave Leising ...................................... 22-23</td>
</tr>
<tr>
<td>ROUNDS ROUTINE, Alton Craig .............................................. 23</td>
</tr>
<tr>
<td>MARKET GAME, Robert Snow .................................................. 24-25</td>
</tr>
<tr>
<td>THE 12 DAYS OF V-MAS, Bill Skillman .................................... 26</td>
</tr>
</tbody>
</table>


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

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Total price in dollars US</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>12,000</td>
</tr>
<tr>
<td>500</td>
<td>17,000</td>
</tr>
<tr>
<td>1,000</td>
<td>25,000</td>
</tr>
<tr>
<td>2,000</td>
<td>42,000</td>
</tr>
<tr>
<td>5,000</td>
<td>92,000</td>
</tr>
</tbody>
</table>

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

If all this attracts you, please tell me ON A SEPARATE SHEET OF PAPER if you also talk of other subjects in your letter:

What you would like to see on the module, how much money you would like to spend for it (taking in account also that we are going to print extensive documentation in the form of a booklet) in short, anything that comes to your mind and that you think might help the effort.

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

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

Maurice E.T. Swinnen
SOLUTION OF THREE EQUATIONS IN THREE UNKNOWNS. - In v5n8p5 I presented a program by Bill Beebe. If you observed closely, everything was there EXCEPT BILL'S PROGRAM. I wanted to complement Bill's "for calculator only" program by one that also had some print out. So, I reworked the program and then proceeded to publish only the enhanced one. My sincere apologies to Bill. Here is his original one. By the way, just to spite me, he sent me an improved version, two steps shorter than his first one.

Instructions: Initialize, press E. This clears all memories.

Coefficients entered in POLAR form, in order as indicated in v5n8p5. Once entry 12 has been entered, the program automatically computes the value of the determinant, delta, in complex form. It will place the real part in the display, while the imaginary part may be obtained by pressing x:i. Avoid division by zero.

Please re-read the article on v5n8p5 with respect to error recovery.
RADAR CALCULATIONS WITH THE TI-59 PROGRAMMABLE CALCULATOR.- Bill Skillman sent me a tentative outline of a book by that title he is writing. I have seen Bill's programs before and I know of his reputation as a technical writer and teacher. So, the book promises to be a winner. Bill would like to invite other programmers to get into the act. The authors will be given due credit, of course. So, here is your chance to be remembered for posterity. To make it easy on Bill (I know what I am talking about, believe me) send your submissions to Bill preferably on a PPX submission form and include recorded mag cards. Bill's address is: 605 Forest View Road, Linthicum Heights, MD, 21090. Bill is an EE and is employed at Westinghouse in nearby Baltimore.

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

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

AN INEXPENSIVE PROGRAMMABLE CALCULATOR SYSTEM FOR EVERYDAY USE IN A RADIATION THERAPY


Programs are for the TI-59 and the Tektronix model 31 calculators.

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

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

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

TIME BOMBS.- Although the program on v5n7/p12 was only intended as a demonstration of Richard Snow's application of "branching from the keyboard", some members asked me to publish the user instructions for that program. Here they are:

Live bombs will explode if: 1) you cut the red wire, 2) you cut the unknown "deadly" wire, 3) you cut the green wire, THEN cut one of the two primary wires, or 4) you run out of time. (the total of seconds left is displayed)

Live bombs can be defused if you cut the unknown "defuse" wire or you cut one of the two primary wires THEN cut the green wire.

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

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

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

After you have entered your function in E, to start the ball rolling press A. Now enter in order, prompted by the calculator:

X1, the start value of X1, which runs in the same direction as the paper strip.
X2, the start value of X2, which runs 90° with the paper strip.
N1, number of "levels" in the contour graph. (0 through 8) N1 stands for "Niveau."
BA, the number of paper strips, any value possible. BA=Bahnren or strips.
e, a precision factor. (try 1.5 as a start)
DX1, the increment of X1.
DX2, the increment of X2.
N, the number of print lines per paper strip.
N10 through N18, value of each level. For each level you have to enter a specific value. There is no default value available.

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

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

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

--------------------------------------------------------

SELECTIVE INV LIST.- Karl-Joseph Meusch wrote this practical routine. It will list --------- a number of data registers to be chosen by the user. It takes about 2.8 sec per register. It mimics exactly INV LIST.
Instructions:
Enter the number of the highest register you want to list and press A.
Enter the number of the lowest register you want to list and press B.
In case you want only one single register listed, enter 0 A N B.

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

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

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

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

Needless to say that the ML module has to be in place. Maybe somebody gets it to work with another module, maybe with different results?
HOEHENLINIENDIAGRAM.- Sample print-outs.
SPEDY_FACTOR_FINDER.- We started the ball rolling on this subject in v5n6p7 with Bill Skillman's program. Suddenly I receive a deluge of similar programs, all claiming to be faster. (some of them are considerably speedier) I once followed with fascination my German friends "slugging it out" in their newsletter Display with the aim of finding out who could write the fastest factor finder program for the HP-67 and the SR-52. Names like K.P. Frank, B. Hoffrichter and V. Lopasie are forever linked in my mind to ingenious approaches and nice programming tricks. In December 1979 I translated all the articles I could find on this subject and distributed them in our local Washington DC club. I still think that it provided the kindle to "light the fire" of the likes of Bill Skillman and Norman Herzberg. Those two more than anybody else have consistently written programs for Speedy Factor Finders (SFF) one faster than the former one. Now, we got some "new" talent working on the problem:

The first program is by Palmer O. Hanson, who adapted Bill's SFF to Fast Mode. It does our two test numbers in really record time:

987654321 (3*3*17*17*37*721) in 1 min 40 sec as opposed to Bill's original: 2 min 26 s. This one will run also in normal mode at 2 min 53 sec.

103569859 (463*467*479) in 1 min 17 sec, while for Bill's original: 1 min 53 sec. This one also runs in normal mode at 2 min 25 sec.

To run in fast mode: Load side 1, RST R/S. Then re-load side 1, CE, enter any integer and press R/S. The calculator/printer will print the integer and all possible factors. It will stop with the highest factor flashing in the display.

Press CE, enter a new integer, etc.

Palmer also sent me a normal mode SFF program, which is not printed here and which is an enhancement of the Louder/Vanderburgh program in v3n1p4 of S2-Notes. It does our first test number in 2 min 14 sec and the second one in 1 min 45 sec. Palmer is now trying to convert that one to fast mode. I will publish it when ready.

The second program here is by Björn Gustavsson from Smedjebacken in Sweden. He wrote a whole new version and converted it also to fast mode. It is even faster than Palmer's. It does the first test number in 1 min 33 sec and the second one in 1 min 11 sec.

To use it, load side 1, RST R/S. Then re-load side 1, enter the integer and press R/S. The integer and its factors will be printed.

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

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

In recent issues of the HP PPC JOURNAL I found that the HP members were asked whether their "fine-tuned" Fast Factoring Program should be a challenge to TI-59 users. If they accept, why not? I am not sure that the TI-59 is a worthy challenger to the HP-41C in this field, but our fast mode should make up for its deficiencies. I know also that Richard Vanderburgh once talked about the existence of an HP-67/69 SFF program that run our second test number at under one minute. Let's give it a whirl.
The last program is Bill Skillman's version of an SR-56 SFF program. To use it, enter the program, enter the integer, press RST R/S. All factors will be printed. If handheld, replace step 47 with a NOP, step 48 with a +/- and step 80 with R/S. Press R/S for each next factor. The last factor will be displayed as a negative value.
TRANSFORM REGRESSION - by John Worthington and Emil Regelman

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

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

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

III. Entering Data: The X, Y data pairs are stored and printed by entering the X value, pressing [X² T], entering the Y value and pressing [A]. Continue entering pairs of data as desired. The program will accommodate data in groups of 15 pairs.
   a. If 15 or fewer data pairs are stored, the data do not have to be recorded.
   b. When the 16th data pair is entered, the program will ask for a data card by printing "WRT". Insert a blank card and continue entering data. Repeat as many times as necessary. The cards containing the recorded data groups should be sequentially numbered, so that they may be reinserted in the correct order. When all data have been stored, press [E] and insert another blank card to record the final data points.

IV. Deleting Data:
   a. If a data pair just entered was in error, press [A']. The calculator will perform the deletion and print "E-", along with the deleted values. (The X and Y values will be recalled into "T" and the display, respectively.) To defeat the deletion (if desired) press [A] which will reenter the data. Continue to press [A'] to delete as many data pairs as desired. In situations where the deletions extend into another group (previously recorded), the calculator will request the insertion of a card by printing "WRT". When the card is inserted the calculator will record the modified data and request the insertion of the card on which the previous set was recorded by printing "READ". The calculator will then delete the last pair of this group of 15 pairs. Also note that this group has now been modified and must be recorded before computations are done.
   b. If the data to be deleted are not the pair just entered;
      1. Enter the number of the pair to be deleted, and press [D].
      2. Then, press [A']. The deleted data will be printed, as noted earlier.
      3. Additional data sets may be deleted by repeating 1. and 2., or as described in IV. a., above.
      4. When all desired deletions have been done, the following additional operations must be performed to reestablish the appropriate counting memories.
         i. Enter the new number of data pairs, and press [D].
         ii. To this number, add the total number of deletions, and press [SBR][SBR].
      5. It would be useful at this point to list the stored data (see VI) to check for errors. If this is done, step 4 (ii) may be omitted.
V. Performing the Regression Analysis:
   a. If less than 16 data pairs have been entered, press [C] to initiate
      the transform regression (note that the designated transforms are printed at this
      time). To compute another transform regression, specify the new transform (see II)
      and press [C].
   b. If more than 15 data sets have been entered (and recorded), be sure to
      record the last-entered group of data (by pressing [E] and inserting a blank card)
      before proceeding. Insert the first data card and press [C]. Then insert the card
      on which the next group of data pairs was recorded. Continue to insert the cards
      as they are read by the calculator. The cards must be inserted in the order in
      which they were recorded.
   c. As data pairs are processed, each X value is briefly displayed (PAU),
      so that the user can verify that the correct data are being processed; if an
      incorrect data group was inserted, press [R/S] and RST] and then start again.
   d. When the summations are completed, the following will be printed:
      \[ N = \text{the number of summations actually performed.} \]
      \[ \text{SLP.} = \text{the slope of the regression line.} \]
      \[ \text{YINT} = \text{the Y intercept of the regression line.} \]
      \[ \text{XINT} = \text{the X intercept of the regression line.} \]
      \[ R^2 = \text{the correlation coefficient, squared.} \]
   e. Steps II and V may be repeated as many times as needed to find the com-
      bination of transforms that yields the highest \( R^2 \) value, or the most suitable
      regression line.

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

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

/SEE PROGRAM LISTING ON NEXT PAGE./

---

GUARD DIGITS PRINTER.- I am not sure if you'll believe me, but this is the last, the
---------- ultimate, the final 13-digits printer I am going to shove down
your collective throats. Enter the number in the display and press A. The 10th, 11th,
12th and 13th digits are printed in OP 04 space. My apologies for the sarcasm to Karl-
Joseph Meusch, the author of this well-written routine. 3.141592654 5359

<table>
<thead>
<tr>
<th>000</th>
<th>76 LBL</th>
<th>014</th>
<th>58 FIX</th>
<th>028</th>
<th>22 INV</th>
<th>042</th>
<th>69 DP'</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>11 A</td>
<td>015</td>
<td>00 00</td>
<td>029</td>
<td>52 EE</td>
<td>043</td>
<td>06 06</td>
</tr>
<tr>
<td>002</td>
<td>53 &lt;</td>
<td>016</td>
<td>52 EE</td>
<td>030</td>
<td>22 INV</td>
<td>044</td>
<td>92 RTH</td>
</tr>
<tr>
<td>003</td>
<td>42 STD</td>
<td>017</td>
<td>94 +/−</td>
<td>031</td>
<td>59 INT</td>
<td>045</td>
<td>76 LBL</td>
</tr>
<tr>
<td>004</td>
<td>00 00</td>
<td>018</td>
<td>65 ×</td>
<td>032</td>
<td>10 E'</td>
<td>046</td>
<td>10 E'</td>
</tr>
<tr>
<td>005</td>
<td>50 IXI</td>
<td>019</td>
<td>00 0</td>
<td>033</td>
<td>10 E'</td>
<td>047</td>
<td>53 (</td>
</tr>
<tr>
<td>006</td>
<td>65 ×</td>
<td>020</td>
<td>42 STD</td>
<td>034</td>
<td>10 E'</td>
<td>048</td>
<td>24 CE</td>
</tr>
<tr>
<td>007</td>
<td>53 (</td>
<td>021</td>
<td>01 01</td>
<td>035</td>
<td>10 E'</td>
<td>049</td>
<td>65 ×</td>
</tr>
<tr>
<td>008</td>
<td>52 EE</td>
<td>022</td>
<td>01 1</td>
<td>036</td>
<td>43 RCL</td>
<td>050</td>
<td>01 1</td>
</tr>
<tr>
<td>009</td>
<td>55 +</td>
<td>023</td>
<td>52 EE</td>
<td>037</td>
<td>01 01</td>
<td>051</td>
<td>00 0</td>
</tr>
<tr>
<td>010</td>
<td>52 EE</td>
<td>024</td>
<td>07 7</td>
<td>038</td>
<td>69 DP</td>
<td>052</td>
<td>49 PRD</td>
</tr>
<tr>
<td>011</td>
<td>00 0</td>
<td>025</td>
<td>54 )</td>
<td>039</td>
<td>04 04</td>
<td>053</td>
<td>01 01</td>
</tr>
<tr>
<td>012</td>
<td>00 0</td>
<td>026</td>
<td>22 INV</td>
<td>040</td>
<td>43 RCL</td>
<td>054</td>
<td>49 PRD</td>
</tr>
<tr>
<td>013</td>
<td>54 )</td>
<td>027</td>
<td>58 FIX</td>
<td>041</td>
<td>00 00</td>
<td>055</td>
<td>01 01</td>
</tr>
</tbody>
</table>
FAST ALPHA SORT.- In response to the RPN programmer's challenge in v5n8, Richard Snow of Vallejo, California, has written this program. It will sort up to 99 five-letter words. Richard has discovered that a program in fast mode can be stopped with a "hard" display if PAU or PRT is used just before R/S. Steps 019 and 020 reflect this principle. The decimal point trick is used here to start the Shell sort routine. We all know that the Shell sort is not the only method available. I hope that this first entry will encourage other good programmers to try new and innovative methods to speed up the sorting.

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

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

On the next page Richard explains how the program works.
FAST ALPHA SORT. - This program is an enhanced version of the Alphabetical Sort program which was previously published in TI PPC NOTES. The program was re-written in response to the RPN programmers' challenge described in V5N5. A list of 99 words in reverse alphabetical order can be sorted in 6 minutes and 34 seconds. It takes only 3 minutes and 1 second for the program to sort a list of words which is already in alphabetical order. Another 40 seconds are needed to print the list in two columns.

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

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

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

The same optimized Shell sort routine is used in this program as in the original Alphabetical Sort program and is described in V5N4.P5.

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

The contents of the registers are printed in two columns using the same method which is used in the Alphabetical Sort program described in V5N4.P6.

This program version was not designed to sort pre-stored print code unless it is already left justified. The program can be modified slightly if you need to sort print code stored in registers and recorded on magnetic cards. First, change 150 to R/S so the program can be stopped when it is finished. Initialize the program by pressing A and re-enter the mag strip with the R/S modification. Force the magnetic cards with print code into the appropriate banks using -2, -3, and -4. This protects data in the registers from being destroyed and assures that the cards will be read while in the 159, 99 partition. Enter the IRN mode and backstep to step 018. Change steps 018 to 020 to "NOP RCL IND 00". Go out of IRN mode, enter the first register number to be left justified. Press "+/- R/S". The program will sort the registers up to the register which contains a zero.

(over)
INVERSE LIST PRINT ALL - FAST MODE
by John Worthington and Emil Regelman

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

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

*  This initiates the fast mode (see TI PPC notes V5N6P4 on the fast mode limitations).
** For example, to list memories ten through nineteen, enter 10.19.
12-DIGIT REGISTER LIST. - I have seen several of these types of listings that print the contents and the register number each on a separate line. But the following two programs print both on one and the same line. The price paid for this "neatness" is, of course, a slow-executing program.

The first program is by Clyde Durbin, Dallas TX. It lists registers 00 through 85 at 26 sec/line. The instructions are simple: Enter the highest register number and press A. Enter the lowest register number and press R/S. The program halts automatically.

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

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

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

<table>
<thead>
<tr>
<th>SAMPLE OF 12-DIGIT PRINT-OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>002 31 CLR</td>
</tr>
<tr>
<td>003 28 88</td>
</tr>
<tr>
<td>004 66 66</td>
</tr>
<tr>
<td>005 31 31</td>
</tr>
<tr>
<td>006 16 A°</td>
</tr>
<tr>
<td>007 00 00</td>
</tr>
<tr>
<td>008 02 HIR</td>
</tr>
<tr>
<td>009 33 37 37</td>
</tr>
<tr>
<td>010 07 07</td>
</tr>
<tr>
<td>011 02 HIR</td>
</tr>
</tbody>
</table>

| 000 24 CE                    |
| 001 21 R/S                  |
| 002 26 66                   |
| 003 16 A°                   |
| 004 09 09                   |
| 005 31 31                   |
| 006 18 C°                   |
| 007 95 88                   |
| 008 33 37                   |
| 009 00 00                   |
| 010 00 00                   |
| 011 00 00                   |
| 012 00 00                   |
| 013 00 00                   |
| 014 00 00                   |
| 015 00 00                   |
| 016 00 00                   |
| 017 00 00                   |
| 018 00 00                   |
| 019 00 00                   |
| 020 00 00                   |
| 021 00 00                   |
| 022 00 00                   |

<table>
<thead>
<tr>
<th>DATA CARD TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>002 04 04</td>
</tr>
<tr>
<td>003 59 INT</td>
</tr>
<tr>
<td>004 04 04</td>
</tr>
<tr>
<td>005 04 04</td>
</tr>
<tr>
<td>006 04 04</td>
</tr>
<tr>
<td>007 04 04</td>
</tr>
<tr>
<td>008 04 04</td>
</tr>
<tr>
<td>009 04 04</td>
</tr>
<tr>
<td>010 04 04</td>
</tr>
<tr>
<td>011 04 04</td>
</tr>
<tr>
<td>012 04 04</td>
</tr>
<tr>
<td>013 04 04</td>
</tr>
<tr>
<td>014 04 04</td>
</tr>
<tr>
<td>015 04 04</td>
</tr>
<tr>
<td>016 04 04</td>
</tr>
<tr>
<td>017 04 04</td>
</tr>
<tr>
<td>018 04 04</td>
</tr>
<tr>
<td>019 04 04</td>
</tr>
<tr>
<td>020 04 04</td>
</tr>
<tr>
<td>021 04 04</td>
</tr>
<tr>
<td>022 04 04</td>
</tr>
</tbody>
</table>
STRUCTURAL PROGRAMS.- For those who are professionally interested in structural programs such as Simple Span Trusses, Steel Beam Columns and Reinforced Concrete Beams, the firm Engineering Calculations, Box 412, North Baldwin Station, Baldwin, NY 11510 has developed several of them and sells them at prices ranging from $10.00 to $35.00 per program. The programs come with complete listing (Very rare! Usually you get only cards recorded with minus!) sample problem, & recorded mag card(s). The programs are divided into four groups: A. Analysis and Design of Steel Structures, B. Analysis and Design of Reinforced Concrete Structures, C. Two Way Slab Systems 1977 A.C.I. Code, D. Foundations. Prices range from $110.00 to $155.00 per group of programs. Write to the above address if interested.

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

Here are the needed corrections:
Step Should be
71 DIV
177 pi
340 square root
346 DIV
352 =
402 DIV
467 DIV

DAYS OF THE WEEK.- Program 20 of the ML module computes the day of the week for you, but you are still required to interpret the output, a digit between 0 and 6 as meaning Saturday through Friday. Several people have sent me elaborate programs to have the machine actually print out the correct day, rather than a "dumb" digit. Among the many attempts the one from Evan Boden is remarkable for its brevity. It would be very well suited for 58 users. Just enter the date in the usual MMDD.YYYY format and press R/S. The output is one single abbreviation, such as TUE or FRI, printed in OP 02 sector. I could not resist enhancing it a little, giving it a user-defined key, printing date and day on one line and writing it such that program and data fit on one and the same card side. Of course, the price paid for all that is double the amount of program steps. For the enhanced program, enter the date in MMDD.YYYY format and press A. For both programs, make sure the ML module is plugged in.

By the way, the Europeans find it cumbersome to remember our "illogical" MMDD.YYYY format. "Logic" is defined as "state the date in ascending order of time duration." Thus, their format is DDMM.YYYY, for which I have seen several conversion programs, so that PGM 20 may be used.

The first program is Evan Boden's, with data in R10 through R16. The second one is the enhanced one by yours truly, with data in R90 through 96. Key in the program and data in 10 OP 17, then record one card side in 6 OP 17.

```
000 36 PGM
001 20 20 361107. 10
002 14 D 364107. 15
003 95 - 372417. 12
004 01 1 312017. 12
005 00 Y 3723415. 15
006 95 = 213524. 16
007 42 STD
008 19 1
009 75 RC+
010 18 18
011 07 DP TUE
012 02 02 TUE
013 01 01 FRD
014 05 05 SAT
015 01 DST
016 36 LLL
017 54 99.13
018 19 13
019 76 LLI
020 15 3
021 73 RC+
022 60 T
023 19 13
024 69 DF
025 60 T
026 46 34
027 59 98
028 49 FCL
029 39 SDD
030 19 13
031 68 DF
032 69 DF
033 19 13
034 59 98
035 69 DF
036 19 13
037 98 DF
038 06 06
039 19 13
040 39 9
041 06 06
042 19 13
043 06 06
044 19 13
045 06 06
046 06 06
047 06 06
048 06 06
049 06 06
050 06 06
051 06 06
052 06 06
053 06 06
054 06 06
055 06 06
056 06 06
057 06 06
058 06 06
059 06 06
060 06 06
061 06 06
062 06 06
063 06 06
064 06 06
065 06 06
066 06 06
067 06 06
068 06 06
069 06 06
070 06 06
071 06 06
072 06 06
073 06 06
074 06 06
075 06 06
076 06 06
077 06 06
078 06 06
079 06 06
080 06 06
081 06 06
082 06 06
083 06 06
084 06 06
085 06 06
086 06 06
087 06 06
088 06 06
089 06 06
090 06 06
091 06 06
092 06 06
093 06 06
094 06 06
095 06 06
096 06 06
```

---

TI PPC NOTES

---

5/8-10/18
PROGRAM LISTING WITHOUT PAPER.- (Re-v5n7p9) John Garza III, of Texas City, TX, has an explanation as to how this might work: "This is a classic example of 'It's so simple, why didn't I think of it? I am sure you are familiar with the use of the GTO key during program execution as stated in Personal Programming. Now think of what OP 08 does. It prints a list of used labels. But first it must scan program memory to find each 76 key code. I believe what you see is this 'scanning' of program memory at a slower than normal rate."

---

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

Has anybody else had the same troubles? Why does it happen?

---

PRINT CODE TABLES.- I was unaware of PPX's contest to print the fastest print code table until I got a call from from Bill Beebe in Lilburn, Georgia. As I saw later in the PPX newsletter, Bill received an honorable mention with a super speedy program. Unfortunately, the PPX people do not encourage the use of the fast mode. I don't know why, but I surmise it has not been proven yet beyond a shadow of a doubt that this mode is not detrimental to the calculator. I have not had any bad experience with it, nor have I heard any complaints from other TI-59 users about their beloved calculator suddenly going on the blink because of the fast mode. But the caution of TI is proverbial.

Here are a few more programs to print a print code table, all respectfully fast. The authors are Bill Beebe, Bill Skillman and Karl-Joseph Meusch.

The first three are rather simple to use: just enter the program, either from the keyboard or from a mag card. Each fits on one card side. Then press A.

The last one is a little more complex. It uses the fast mode and in this mode really goes to town. To use it, put it first on one side of a mag card. Then enter the card, press RST R/S, which initializes the fast mode. Then enter the card again and see the printing start. To show how slow normal mode is in comparison, press RST, which takes the program out of fast mode. Then press A. The same table is printed, but ever so slowly. Note also that this program prints the extended print code table. The author of this jewel is K-J Meusch, Koenigsdorf, West Germany.

---

(over)
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01234567</td>
<td>01234578</td>
<td>01234567</td>
<td>01234578</td>
</tr>
<tr>
<td>01234567</td>
<td>01234578</td>
<td>01234567</td>
<td>01234578</td>
</tr>
<tr>
<td>01234567</td>
<td>01234578</td>
<td>01234567</td>
<td>01234578</td>
</tr>
<tr>
<td>01234567</td>
<td>01234578</td>
<td>01234567</td>
<td>01234578</td>
</tr>
<tr>
<td>01234567</td>
<td>01234578</td>
<td>01234567</td>
<td>01234578</td>
</tr>
<tr>
<td>01234567</td>
<td>01234578</td>
<td>01234567</td>
<td>01234578</td>
</tr>
<tr>
<td>01234567</td>
<td>01234578</td>
<td>01234567</td>
<td>01234578</td>
</tr>
</tbody>
</table>

**Source:** TI PPC NOTES

**Description:** PRINT CODE TABLES (continued)
PROGRAMMING PUZZLES.— (re-v5n7/p5) I received several solutions to selected puzzles. As you will remember, these puzzles were submitted by George Vogel, Newton Highlands, Mass. The first one went as follows:

1. You design a program which stops in the middle of a subroutine and according to the value displayed, you plan to use your judgement to decide whether to press A or B to complete the computation. Let's say that labels A and B are both within the subroutine, e.g. ... R/S LBL A 2 + LBL B 1 = RTN.... But it doesn't work: the program always stops at the RTN and will not return to the main program. What is wrong? How can you make your idea work?

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

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

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

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

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

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

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

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

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

Morton P. Matthew has this 34-step routine:

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

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

Jeff Rosendale's laconic comment:

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

2. You have a program with a number of OP 06 comments. You want to keep your choice of FIX open for each calculation. You don't want to use FIX IND or assemble codes via HIR 8. Can you think of a simple way of avoiding distortion of the comments, regardless of the FIX chosen?

George says:

Enter each print code into the program as a ten-digit number, starting it 10...

(or any two digits of which the first one is not a zero) and supply two more zeros for each blank position, if any. E.g.: 1000131415 will print ABC regardless of FIX. Both Jeff Rosendale and Richard Snow came up with the same solution:

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

3. Write a histogram program that will accumulate in separate bins (registers) the number of times x occurs in the following steps-of-five ranges:

x≤60, 60<x≤65, 65<x≤70, etc. Program should have 15 steps or less.

George offers this solution:

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

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

```
DIV 5 - 1 = STO 09 X>T 1 SUM IND 09 X=T 002 RTN which is a 16-step solution.
```

When 60 is input, 1 must be added to two different registers since 60 satisfies the first two arguments given in the problem. (continues next page)
PROGRAMMING PUZZLES. (continued)

Karl-Joseph Meusch also sent a similar solution to puzzle 3:

DIV 5 - 11 = STD 59 1 SUM IND 59 R/S RST

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

George Vogel's solution is as follows:

8.7 STD 09 RCL 09 STD IND 09 DSZ 0 005 R/S This takes advantage of the fact that
in extracting the direct address from reg 09 (or any other register) the calculator
ignores the fractional part, and suggests a generally useful trick of including reg.00
in a DSZ procedure, e.g. clearing selectively the contents of registers from, say
8 down to 0 (replace RCL 09 by CLR in the above). Register 00 will be included if
you use as the initial indirect address not 8, but 8 plus some fraction.

Most participants didn't have too much trouble with this one. J. Huntington Lewis
sends this one: 8.7 STD 09 STD IND 09 - 1 ) DSZ 0 005 R/S

Karl-Joseph Meusch writes: 8.7 STD 00 STD IND 00 - 1 = GE 003 R/S

And Richard Snow devised George Vogel's routine to a Tg, so did Charles Williamson.

And finally Morton P. Matthew has one with a PAUSE included, so you will know when
to stop: .7 SUM 59 RCL 59 STD IND 59 PAU 1 GTO 002

5. Try this on the keyboard: 9.999 EE +/- 87 STD 01 STD 02 1 EE +/- 99 SUM 02
   .1 PRD 01 PRD 02 . You now have two small numbers in the two registers. Then,
   still from the keyboard, do RCL 01 X:T RCL 02 X:T . Why the flashing? And why do
   you have to press CE twice to stop the flashing?

George Vogel offers this comment:

You have synthesized and stored in registers 01 and 02 two numbers which are still
legal but differ by less than 1E-99. The comparison is done internally by subtraction
which thus yields an illegal number. (This cannot happen unless two numbers are numer-
ically smaller than 1E-87.) Why must CE (or CLR) be pressed twice? The first push
is interpreted only as the address for the X:T ( or EQ if you prefer ) command, and
only the second push will stop the flashing. (GE is subject to the same rules as EQ)
It is interesting to note that, even when the condition described above causes flas-
hing, the comparison and the corresponding branching are done correctly. It may seem
an unlikely condition to occur, but it can happen and cause a big headache.

Richard Snow writes:

As derived from several articles on computers, it is evident that whenever a compari-
son is made, an internal subtraction is performed. The subtraction from the t-register
results in an underflow error in this sample problem. The first CE does not clear
the error because it completes the instruction " X:T CE " where CE is merely the name
of the label. The second CE is treated as the normal error-clearing instruction.

And Jeff Rosedale? He gave up after puzzle 2. He simply writes:

3, 4, 5?

My thanks to the many contributors. It has been fun.

BRAIN TEASERS.- See also v5n7p5 and v5n8p4. A few more solutions came in:

Björn Gustavsson in Sweden has these two:

For .1415... : INV Σ+ RAD INV COS INV INT ( that's the beginning of Don O'
grady's routine too )

For 197: INV LOG INV LOG X² + CE - LOG ) INT

Meyer Bolland on the other hand goes like this:

For .1415... Σ+ +/- INV COS - X:T - Σ+ = (8 steps)

For 197: Σ+ Σ+ X INV LOG - Σ+ = (7 steps)

MASTER LIBRARY SURVIVAL GUIDE.- Fred Fish who wrote this fantastic manual in 1978,
has a new address:

1346 W. 10th Place, Tempe. Arizona, USA. Tel: (602) 894-6881.
Fred still has a few copies available.
FAST MODE MORSE CODE.- Dave Leising.

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

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

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

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

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

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

| 372317008 | 00 | 074 00 | 0 | 076 00 | 0 | 078 00 | 0 | 080 00 | 0 |
| 3220007212 | 04 | 077 00 | 20 | 079 00 | 20 | 081 00 | 20 | 083 00 | 20 |
| 3164922122 | 0 | 084 00 | 20 | 086 00 | 20 | 088 00 | 20 | 090 00 | 20 |
| 1750000005 | 05 | 085 00 | 44 | 167 00 | 44 | 169 00 | 44 | 171 00 | 44 |
| 4455 | 11 | 086 00 | 44 | 088 00 | 44 | 090 00 | 44 | 092 00 | 44 |
| 24037542 | 07 | 087 00 | 44 | 089 00 | 44 | 091 00 | 44 | 093 00 | 44 |
| 1336662073 | 08 | 088 00 | 44 | 090 00 | 44 | 092 00 | 44 | 094 00 | 44 |
| 1342841122 | 09 | 089 00 | 44 | 091 00 | 44 | 093 00 | 44 | 095 00 | 44 |
| 43526243 | 10 | 090 00 | 44 | 092 00 | 44 | 094 00 | 44 | 096 00 | 44 |
| 5700000000 | 11 | 091 00 | 44 | 093 00 | 44 | 095 00 | 44 | 097 00 | 44 |
| 203242 | 14 | 092 00 | 44 | 094 00 | 44 | 096 00 | 44 | 098 00 | 44 |
| 1732608211 | 16 | 093 00 | 44 | 095 00 | 44 | 097 00 | 44 | 099 00 | 44 |
| 3122569013 | 18 | 094 00 | 44 | 096 00 | 44 | 098 00 | 44 | 098 00 | 44 |
| 2726082432 | 20 | 095 00 | 44 | 097 00 | 44 | 099 00 | 44 | 099 00 | 44 |
| 4765000000 | 22 | 096 00 | 44 | 098 00 | 44 | 098 00 | 44 | 099 00 | 44 |
| 1797450000 | 23 | 097 00 | 44 | 099 00 | 44 | 099 00 | 44 | 099 00 | 44 |
| 32132 | 24 | 098 00 | 44 | 098 00 | 44 | 098 00 | 44 | 098 00 | 44 |
| 3930402427 | 25 | 099 00 | 44 | 099 00 | 44 | 099 00 | 44 | 099 00 | 44 |
| 1951312627 | 26 | 099 00 | 44 | 099 00 | 44 | 099 00 | 44 | 099 00 | 44 |
| 412517064 | 27 | 100 00 | 44 | 100 00 | 44 | 100 00 | 44 | 100 00 | 44 |
| 3762141131 | 28 | 101 00 | 44 | 101 00 | 44 | 101 00 | 44 | 101 00 | 44 |
| 2600375250 | 29 | 102 00 | 44 | 102 00 | 44 | 102 00 | 44 | 102 00 | 44 |
| 191331 | 30 | 103 00 | 44 | 103 00 | 44 | 103 00 | 44 | 103 00 | 44 |
| 151720062 | 31 | 104 00 | 44 | 104 00 | 44 | 104 00 | 44 | 104 00 | 44 |
| 1327210013 | 32 | 105 00 | 44 | 105 00 | 44 | 105 00 | 44 | 105 00 | 44 |
| 157634117 | 33 | 106 00 | 44 | 106 00 | 44 | 106 00 | 44 | 106 00 | 44 |
| 3122550013 | 34 | 107 00 | 44 | 107 00 | 44 | 107 00 | 44 | 107 00 | 44 |
| 2726444591 | 35 | 108 00 | 44 | 108 00 | 44 | 108 00 | 44 | 108 00 | 44 |
| 1726444591 | 36 | 109 00 | 44 | 109 00 | 44 | 109 00 | 44 | 109 00 | 44 |
| 32132 | 37 | 110 00 | 44 | 110 00 | 44 | 110 00 | 44 | 110 00 | 44 |
| 32417200 | 38 | 111 00 | 44 | 111 00 | 44 | 111 00 | 44 | 111 00 | 44 |
| 1000000000 | 39 | 112 00 | 44 | 112 00 | 44 | 112 00 | 44 | 112 00 | 44 |
| 3221002437 | 40 | 113 00 | 44 | 113 00 | 44 | 113 00 | 44 | 113 00 | 44 |

---

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

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

```
LBL A PRT X:T 10 Y^X R/S = X X:T + .5 = INT DIV X:T = PRT R/S
```
MARKET, a game program by Robert Snow.

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

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

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

11. SPLIT: One share of stock is earned for each two full shares held by you.
12. DIVIDEND SUSPENDED: At the last minute, a declared tax or dividend is cancelled.

(Some of these "events" seem rather fiendish)

PROCEDURE:

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

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

   Thus, if you want to sell, say, four shares of gold, you enter as 4.1

To buy, press B. To sell, press C.
3. To obtain the current MARKET NEWS, at all times press R/S.
4. Your account is displayed after every buy or sell. To get a printed statement, press A.
5. Go to steps 2 or 3.

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

The above print-out is an example of what to expect, especially if you cheat, as I tried to do.
Program keyed-in in 4 OP 17, recorded in 6 OP 17.  SEE PROGRAM ON NEXT PAGE.
THE TWELVE DAYS OF CHRISTMAS.- Two years ago Jill Zimmerman wrote a program by that name in Kilobaud. If one has 16 K memory and Basic available, printing that famous Christmas carol is a cinch. But it is of course the idea that counts. Now, when the Snow brothers saw that program and said to themselves: "How about making the TI-59 do it?", that was a real challenge. So, just before Christmas 1979 I received a version by Richard Snow. Once it was suitably distributed among the members of our local club, lots of new versions cropped up. I even wrote one myself and sent it to PXP, not as a submission, but to amuse the my friends the program analysts.

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

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

To run the program, load the four card sides and press A. About a yard of paper will be the result of all the printing. "But it is cute," you'll say.
TI PPC NOTES 1981 SUBSCRIPTION FORM. Please copy this form and mail back.

Distribute a few copies among your friends.

I WANT A 1981 SUBSCRIPTION ONLY [ ] Please check appropriate box(es)

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

I WANT ALL THE BACK ISSUES FOR 1980 [ ]

US, Canada & Mexico @ 20.00
Overseas air mail @ 24.00

FIRST & LAST NAME  ..............................................................
ADDRESS (# & STREET) ............................................................
CITY .................................................................
STATE & ZIP CODE # .........................................................
COUNTRY .................................................................

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

I HAVE $........ DOLLARS US AVAILABLE TO BUY $........ NUMBER OF MODULES.

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

__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________

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

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