PPX Exchange

Vol. 6 Number 5 Copyright 1982 September/October 1982

Base Converter

The number system familiar to most of us is the base 10 number system. This is probably due to the number of fingers (and thumbs) on the hands of early man. To indicate having used each finger as a count, the number 10 was used. But what a convenience (mathematically at least) it would be if man had but 1 finger on each hand! Binary arithmetic - the language of computers in every country - would be as simple as counting on our fingers! Or better yet, if homosapiens had 8 fingers on each hand, counting would be in base 16 (hexadecimal) form. This would be computer shorthand as every number would represent 4 binary digits. But what about 999999 fingers? The programs presented here will make the conversion from base 10 to any other base or from any base to base 10 without making life difficult for glove makers or calculator manufacturers.

continued on page 7

Googol-Eyed

by Lee Keith
Software Design Engineer

There is a certain mystery and aloofness that an individual or group enjoys when others are ignorant of their shortcomings. The fact that everyone knows that President Reagan has a weakness for jellybeans takes away from the aura of perfection that surrounds all presidents. For those of you who ever ever wondered what it is like working around PPX, let me relate a story.

During one of the many brainstorming sessions that PPX staffers sometimes engage in to think of exciting ideas for this newsletter, this software engineer's attention was caught by the ramblings of an obviously deranged undergraduate from a local university. While discussing large (very large) numbers, the undergraduate speculated on the amount of time it would take the TI-59 to count from -9.99999 X 10^{99} to 9.99999 X 10^{99} by ones! To the uninitiated (and some undergrads), this does not sound like a formidable task. Most would agree that it would probably take at least a weekend!

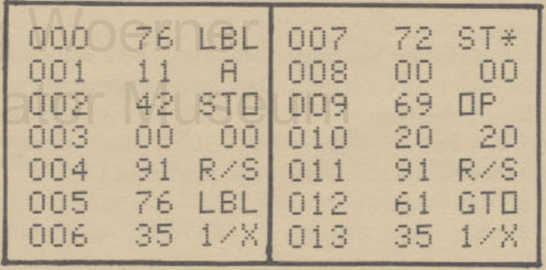
How long *would* it take? The range of integers from -9.99999 X 10^{99} to 9.99999 X 10^{99} contains 18.99999 X 10^{99} integers, or 1.899999 X 10^{100} integers if you prefer. Now things look sticky! We have found that the TI-59 can ex-

continued on page 2

Keeping It Simple

(Editor's Note: Because of the room necessary to explain program development completely, the programming techniques presented in this column will be of little utility save for instructional purposes. The novice programmer can use this information for the development of programming skills, and the more experienced programmer can use the techniques as a refresher course in well structured friendly programming.)

The first article in "KEEPING IT SIMPLE" was a data entry program discussion label transfer addressing and indirect data register addressing, and we arrived at the following program listing:



```
000  76 LBL | 007  72 ST*
001  11  A  | 008  00  00
002  42 STO | 009  69 OP
003  00  00 | 010  20  20
004  91 R/S | 011  91 R/S
005  76 LBL | 012  61 GTO
006  35 1/X | 013  35 1/X
```

The data, now stored in sequential data registers, can be manipulated by another program. The subject of this article will be a sorting program which will sort the values stored in the previous article. In accomplishing this, the program will cover absolute addressing, logical and flag testing, and DSZ looping.

The sorting method chosen is the "bubble sort". This is a brute force method of sorting data, and is one of the least efficient of the sorting methods available. The main advantage of the bubble sort is that it is easy to understand and simple to code.

The desired operation of the program will be outlined, the flowchart symbols used will be defined, the order of programmed events will be flowcharted, and finally, the keystroke implementation will be explained in detail.

Given a sequential list of values, the bubble sort begins by comparing the first value with the second. If the first value is larger than the second, the order of the two values is exchanged; the first value becomes the second and the second value takes the place of the first. Whether an exchange takes place or not, the sort continues by comparing the present second value with the third. The process is repeated until the next-to-last value is compared with the last value and the

continued on page 6

Googol-Eyed continued from page 1

ecute seven additions per second, but even at this rate, it would take 2.7143X 10^{10} seconds to count to 1.899999 X 10^{100} which is equivalent to 4.5238 X 10^{97} minutes or 7.5397 X 10^{95} hours or 3.1416 X 10^{94} days or 8.607 X 10^{91} years or 8.607 X 10^{88} millennia or 8.607 X 10^{85} billion years! WHEW! This number is in excess of 10^{84} times longer than the estimated age of the universe! Even a modern digital computer doing 10^{9} additions per second would take over 6 X 10^{74} billion years to do this task. Needless to say, our undergraduate aged at least that much when the answer came to light.

It turns out that since the TI-59 has an accuracy of only 13 digits, the above calculation, even if programmed, would be impossible. The TI-59, even with all its seemingly magical powers and functions, cannot hold more than the most significant, or leftmost, 13 digits, i.e. to add 1 to 9.999999999999 X 10^{99}:

	SIGNIFICANT DIGITS	TRUNCATED DIGITS
	9.999999999999	00000000···00000000···000000000 X 10^{99}
+	.000000000000	00000000···00000000···000000001 X 10^{99}
	9.999999999999	00000000···00000000···000000001 X 10^{99}

As you can see, the "1" pales into insignificance in comparison to 9.9999999999 X 10^{99}. Lucky for the undergraduate too, for he would have to forego the use of his TI-59 for 8.607 X 10^{85} billion years, which is nearly the length of time it will take him, without the use of his TI-59, to graduate.

Surely everyone has played with their TI-59 and tried to see just how big of a number they could represent without overflowing the capability of the TI-59 (I have). The limit turns out to be 9.999999999999 X 10^{99}, which appears to be just short of that seemingly magic number of 1 X 10^{100}. It is such a nice round number, 1 X 10^{100}. It is so nice, in fact, that it has been named a GOOGOL (pronounced GOOGULL)!

Who do we have to thank for this incredible outgushing of creativity? The credit belongs to Milton Sirotta, who in 1938 at the age of nine coined the name for his uncle Edward Kasner, an American mathematician.

I am sure that many of you do not realize exactly *how* immense 10^{100} really is. Get out your pencil and paper and try to write a googol, 10 followed by 99 zeros: 10,000,000,000,000,000,000,000,000,000,000,000,000-,000,000,000,000,000,000,000,000,000,000,000,000,-000,000,000,000,000,000,000,000,000,000,000! Can *anything* in the real world be compared to a googol? Unfortunately, I could not find anything.

Let's think of some large numbers that mean something to us, and compare them to a googol. How about the National debt? The national debt now is estimated to stand at more than one trillion dollars. That is only 1 X 10^{12}. What about all the grains of sand on all of the beaches of the world? Better yet, what if the whole world was one solid gigantic beach, inside and out? If we assume a grain of sand to be 1mm on a side, then "only" 1.1 X 10^{30} grains of sand could fit in the Earth (remember that this is not one third of a googol but is 10^{-70} of a googol)! Let me go one step further and instead of a pure sand Earth, we use an Earth made of tiny neutrons. Would the Earth contain a googol of neutrons when packed side by side? If the neutron has a diameter of 1.6 X 10^{-13} cm then the Earth can hold only 5.1 X 10^{65} neutrons! As you can see, we have a very long way to go.

How many neutrons would there be if all of the matter in the Universe consisted entirely of neutrons? This is surely not only the largest number with any physical meaning, but must also be larger than a mere 10^{100}! If the observable Universe is taken as having 100 billion galaxies, and each galaxy has 100 billion stars, and if each star is the size of the Sun, holding 6.6 X 10^{71} neutrons, then the Universe would contain 6.6 X 10^{93}neutrons. This number, though immense, is only .00000015 of a googol!

I cannot seem to find a googol of anything in the Universe!

If any PPX member knows of a googol of something, please send it to PPX in care of the Exchange editor, and it will be published in a future issue of the newsletter.

Some readers might ask whether a number as big as a googol is good for anything. Well, the immortal Issac Asimov put it perfectly in his essay "T-Formation" when he said the googol is useful "for its own sheer, abstract beauty." If that is the best *he* can do, what do you expect me to say?

If you were impressed with all that, let me tell you of an even *larger* number than the googol. Imagine, if you will, ten raised to the googol power! That is $_{10}10^{100}$ or one followed by a googol of zeros (this one will not be written for you.) What would one call such a number? A one followed by a googol of zeros is called a googolplex.

When one deals with numbers as large as a googolplex, just writing it becomes literally impossible! While a googol can be written in just the space of a few lines, a googolplex cannot be written no matter how small one makes the zeros. In order to show you the magnitude of such a task, let me bring back Dr. Asimov to explain. In his book, *Asimov On Numbers*, Dr. Asimov points out the futility of having to merely write out a googolplex (keep in mind that a nucleon is either a proton or neutron):

> "... you could not write the number on the entire surface of the Earth, if you made each zero no larger than an atom. In fact, if you represented each zero by a nucleon, there wouldn't be enough nucleons in the entire known Universe or in a trillion like it to supply you with sufficient zeros."*

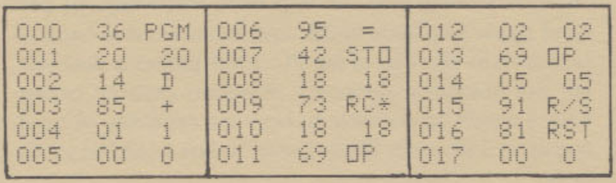
Earlier, I failed to find a googol of neutrons in the known Universe. Let me now take that analysis to the limit by filling the Universe with the little buggers! Neutrons *everywhere*, even in places where they could not exist or should not be; packed so tightly that they touch. The Universe will become like a giant neutron with only 6.6 X 12^{123} neutrons in it! You will notice that we have finally created a number larger than a googol, but had to go to absurd and unimaginable extremes to do it. Yet, after all of that, this new number is still almost zero when compared to a googolplex!

It's enough to make you GOOGOL-EYED!!!

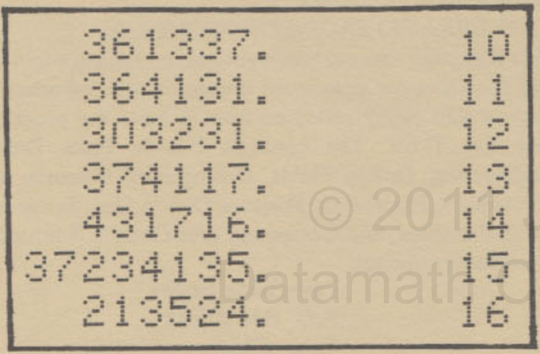
*From the book: ASIMOV ON NUMBERS by Isaac Asimov. Copyright © 1978 by Isaac Asimov. Published by Doubleday and Co., Inc.

from the Analyst's Desk

- Mr. Evan Boden has written us with a short routine for taking the day of the week (0 through 6) as computed by ML-20 and converting it to the print code for the day's abbreviation. The abbreviation is then printed. To run the sequence, key in the 17 step program and load the following print codes in the specified data registers 10-16. You may then enter the date, press [R/S], and the abbreviated day of the week will be printed. The Master Library Module must be installed.



```
000  36 PGM   006  95  =     012  02  02
001  20  20   007  42 STO    013  69 OP
002  14  D    008  18  18    014  05  05
003  85  +    009  73 RC*    015  91 R/S
004  01  1    010  18  18    016  81 RST
005  00  0    011  69 OP     017  00  0
```



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

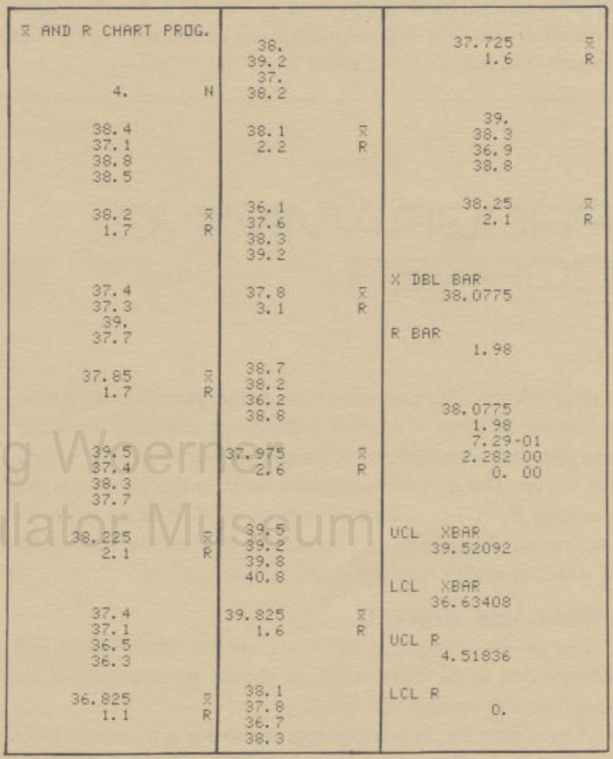
- Those members who are familiar with micro-computers are aware of the chaining operation. Member Andre Verville explains that this operation makes it possible to control the reading of programs on the disk memory and to begin their execution without leaving the running cycle of a main program. Surprisingly, the TI-59 is, in fact, provided with this option.

While in the execution of a program, you already know that you can induce the reading of data from a magnetic card with the instructions N [INV] [2nd] [Write]. This is very practical but what about reading program listings? You can order the reading of a program partition which is the same as the one you are working in. If you are in Bank 2, at step 305, you can put the instructions [CLR] 2 [INV] [2nd] [Write] [GTO] [1/x] at that location. This is a completely new set of instructions on the card that has to be read, and the bank number is 2. The processor will read the card while in program mode, replace steps 240-479 with the new code, find label [1/x], and begin execution from that program location.

With this procedure, you can make a program as long as is needed. The control of the program is strategically placed by using N [INV] [2nd] [Write] instructions within the program, and the user need only to deliver the cards to the calculator. Those members with printers can make the process even more complete by providing the user with printed prompts to enter the correct card side.

By using this chaining operation, a complete programming system is possible, meaning that you never leave program execution and the TI-59 can be used like a micro-computer in a conversational mode.

- Program #278013 "Average (x̄) and Range (R) Control Charts" which is also contained in the "Quality Assurance II/Control Charts" Packette has an error in the sample problem output. Location 049 should be 01 and the sample problem output should read as follows:



```
x̄ AND R CHART PROG.

      4.      N

     38.4
     37.1
     38.8
     38.5

     38.2     x̄
      1.7     R

     37.4
     37.3
     39.
     37.7

     37.85    x̄
      1.7     R

     39.5
     37.4
     38.3
     37.7

     38.225   x̄
      2.1     R

     37.4
     37.1
     36.5
     36.3

     36.825   x̄
      1.1     R
```

```
     38.
     39.2
     37.
     38.2

     38.1     x̄
      2.2     R

     36.1
     37.6
     38.3
     39.2

     37.8     x̄
      3.1     R

     38.7
     38.2
     36.2
     38.8

     37.975   x̄
      2.6     R

     39.5
     39.2
     39.8
     40.8

     39.825   x̄
      1.6     R

     38.1
     37.8
     36.7
     38.3
```

```
     37.725   x̄
      1.6     R

     39.
     38.3
     36.9
     38.8

     38.25    x̄
      2.1     R

X DBL BAR
   38.0775

R BAR
      1.98

   38.0775
      1.98
      7.29-01
    2.282 00
       0. 00

UCL  XBAR
  39.52092

LCL  XBAR
  36.63408

UCL R
   4.51836

LCL R
       0.
```

- Program #398058 listed as "Progressions (Arith., Geom., Harmonic)" should be correctly titled as "Laguerre Polynomials" by Andrew J. Nicola, Jr. The program calculates the Laguerre polynomials for N = 1 through 10. The program has 784 steps.

ADDRESS CHANGES

In order to ensure uninterrupted service, please submit address changes to PPX at lease six weeks prior to the effective date of the change. Send your name, membership number, old and new addresses to:

PPX
P.O. Box 109
Lubbock, TX 79408

On Revisions

We would like to remind all members that are submitting revisions to PPX for approval that a revision is a complete program and must be submitted as such. We must require this so that we (as well as members purchasing the program) are assured that a full program is contained in our files. In the future, a partial program submitted as a revision will be returned to the author along with a request that the complete program be submitted. If the complete revision is accepted, the author will receive magnetic cards in the stated 4:1 ratio and an order form good for one free program of their choice.

We would also like to remind you that the option of revising a program contained in the Software Catalog lies with the original author. Any program revisions submitted by someone other than the original author will be referred by PPX to the original author for possible revision.

PROGRAMMING CORNER

(Editor's Note: This column serves a dual purpose. It informs members of what non-PPX software is currently available and also lists descriptions of programs our members would like to see. The non-PPX software listed in this column is not available from Texas Instruments or PPX, and all inquiries concerning the pricing and availability of these items should be directed to the contacts listed below.)

PROGRAMS WANTED

The program requests for this issue are listed below. All submissions to fill these requests should be postmarked no later than November 15, 1982.

- A program for non-linear regression for a tri-exponential curve that will be compatible with PPX #000014 "Regression Analysis and Multivariate Statistical Methods" programming system.
- A program to derive magnetic headings, ground speeds, leg times, and fuel used for navigation legs determined by magnetic course and distance or from points described by latitude and longitude using rhumbline navigation to first derive the true course and distance between points. The program should also keep track of the actual time or elapsed time at each turn point and the fuel remaining at that point. The program should compute the turn required at each turn point for the given airspeed, ground speed, and bank angle, rollout point latitude/longitude, the arc distance flown during the turn, the time required to fly the turn, and the true course and distance from the roll out point to the next turn point. The arc distance and the distance from the roll out point will be added to determine the total distance flown from the last turn point to the next turn point. Program should also be compatible with the Vortac-point to Point and Rhumbline Navigation programs which can be used to calculate the latitude/longitude of a point given the magnetic or true course and distance to a point from a known point.
- A program to compute the local times of sunrise, noon and sunset from any place in the world. The input should be the altitude and latitude of the place, the difference between local time and GMT, and the date.
- A program to determine the direction of the city of Mecca in Saudi Arabia from any place in the world. The direction may be with reference to the North and also with reference to the shade of the sun cast by a vertical stick in the ground at any particular time. The inputs should be the altitude, latitude, local time, difference between local time and GMT, and the date.
- A program to calculate the pole and zero locations of a "nth" order low pass elliptic filter. Given is the filter order "n'; cut-off frequency, fc; stop band frequency, fs; pass band ripple amplitude in dB; and minimum stop band attenuation in dB. Use of the PC-100A printer is desirable, but not absolutely necessary. Program capacity permitting, it would be desirable to have the capability of handling up to 14th order filters. Lesser orders would be acceptable.

SOFTWARE AVAILABLE

- *TI-59 DRILLING ENGINEERING MANUAL*

  This 232 page manual has twenty-seven drilling engineering programs for petroleum engineers and introductory step-by-step explanations on how to input the programs into the TI-59. The contents include Basic Drilling Engineering, Drilling Fluids, Drilling Fluid Viscosity and Circulation, Hydrostatic Pressure Due to Gas, Surge and Swab Pressures, and Cementing Well Control. For further information contact:

  Penn Wells Books
  P.O. Box 21288
  Tulsa, Oklahoma 74121

- A CASE FOR PORTABILITY

  The power and compactness of the TI—58/59, PC-100A(C) combination make a good case for using it to solve problems in the field. When you are among those users carrying it between daily sales calls or to distant work sites, you would say there is also a good case for a GOOD CASE. The case pictured below is manufactured specifically for carrying the calculator/printer combination. The dark brown leather-like vinyl is handsome and durable, and the shock absorbent foam lining provides protection for the equipment. Heavy duty zippers and a

PC cont'd

positive latching device prevent the lid from accidentally opening during transit, and a detachable shoulder strap provides "ease of handling" when your hands are full.

The carrying case is available to PPX members through H/S Enterprises at a special low price of $46.50 each (includes shipping and handling within the continental United States.) To order, be sure to specify the number of cases wanted, your correct mailing address, and mail your check or money order for the exact amount to:

H/S Enterprises
11902 Jones Road, Suite L262
Houston, Texas 77070

For orders to be shipped outside the continental U.S., please add $10.00 per case for shipping and handling charges, and allow four weeks for delivery. To avoid additional delays, be sure to include your mailing address.

Letters to the Editor

(Editor's Note: Do you have any comments or questions on the Exchange of other aspects of PPX that might benefit other members? We have always welcomed letters from our membership, and therefore, we provide this space in the newsletter for you to share your views with your fellow members. Approximately 2-4 letters dealing with issues of general interest will be featured as space permits.)

Dear Editor:

On occasion, I have a need for a subroutine for the remainder function Y modulo X yet I have not seen this function mentioned in the *Exchange* nor have I noticed it in the programs I have happened to order. Hence, I am attaching a routine which I find useful.

If Y and X are generated in the program and are stored in R_{14} and R_{15}, calling the subroutine produces Y modulo X.

```
000  76 LBL | 007  43 RCL | 014  94 +/-
001  71 SBR | 008  15  15 | 015  85  +
002  53  (  | 009  54  )  | 016  43 RCL
003  53  (  | 010  59 INT | 017  14  14
004  43 RCL | 011  65  ×  | 018  54  )
005  14  14 | 012  43 RCL | 019  54  )
006  55  ÷  | 013  15  15 | 020  92 RTN
```

As an example, store 27 in R_{14} and 5 in R_{15}. Press [SBR] [SBR] and 2 is displayed equaling 27 module 5. Of course the routine can easily be altered so that X or Y or both can be entered from the keyboard.

Sincerely,
W.W. Buechner
Arlington, MA

Dear Editor:

In the May/June 1982 issue of the PPX *Exchange* a letter from John A. Lawlor appears concerning the article "Root Finding: A Natural Application." Mr. Lawlor found some improvements to the Regula Falsi method; however, as I experimented with the revised program, I found that there are times when the increment on the variable is made zero, and as the value of F(x) is not within the range specified, the program will continue running indefinitely.

I have made a very small modification to the revised program that seems to overcome the problem. Using as a sample problem:

$$F(x) = x^5 - 125 x^4 - 23 x^2 - 125 x - 128$$

which has only one real root, the program will go no further than:

$$x = 125.1337988$$
$$F(x) = - 0.086441433$$

The program will stop when the increment on x is zero. Below you will find the re-revised listing of the program.

Yours truly,
Andres Maurer
Mexico City, Mexico

```
000  76 LBL | 059  42 STO | 118  16 A'  | 177  81  81
001  11  A  | 060  06  06 | 119  42 STO | 178  61 GTO
002  42 STO | 061  43 RCL | 120  11  11 | 179  00  00
003  01  01 | 062  06  06 | 121  33 X²  | 180  67  67
004  22 INV | 063  16 A'  | 122  85  +  | 181  43 RCL
005  86 STF | 064  87 IFF | 123  43 RCL | 182  06  06
006  01  01 | 065  01  01 | 124  08  08 | 183  91 R/S
007  93  .  | 066  17 B'  | 125  95  =  | 184  61 GTO
008  00  0  | 067  42 STO | 126  34 ΓX  | 185  01  01
009  01  1  | 068  10  10 | 127  32 X:T | 186  65  65
010  42 STO | 069  43 RCL | 128  43 RCL | 187  25 CLR
011  04  04 | 070  05  05 | 129  04  04 | 188  35 1/X
012  91 R/S | 071  16 A'  | 130  77  GE | 189  43 RCL
013  76 LBL | 072  29 CP  | 131  00  00 | 190  07  07
014  12  B  | 073  67  EQ | 132  28  28 | 191  91 R/S
015  42 STO | 074  01  01 | 133  43 RCL | 192  76 LBL
016  02  02 | 075  68  68 | 134  11  11 | 193  16 A'
017  91 R/S | 076  42 STO | 135  65  ×  | 194  42 STO
018  76 LBL | 077  09  09 | 136  43 RCL | 195  00  00
019  15  E  | 078  65  ×  | 137  09  09 | 196  65  ×
020  42 STO | 079  43 RCL | 138  95  =  | 197  53  (
021  04  04 | 080  10  10 | 139  29 CP  | 198  24 CE
022  91 R/S | 081  95  =  | 140  77  GE | 199  33 X²
023  76 LBL | 082  29 CP  | 141  01  01 | 200  33 X²
024  13  C  | 083  77  GE | 142  54  54 | 201  75  -
025  42 STO | 084  14  D  | 143  43 RCL | 202  43 RCL
026  03  03 | 085  43 RCL | 144  07  07 | 203  00  00
027  91 R/S | 086  09  09 | 145  42 STO | 204  65  ×
028  43 RCL | 087  55  ÷  | 146  06  06 | 205  24 CE
029  07  07 | 088  53  (  | 147  43 RCL | 206  33 X²
030  91 R/S | 089  24 CE  | 148  11  11 | 207  65  ×
031  76 LBL | 090  75  -  | 149  42 STO | 208  01  1
032  14  D  | 091  43 RCL | 150  10  10 | 209  02  2
033  43 RCL | 092  10  10 | 151  61 GTO | 210  05  5
034  01  01 | 093  54  )  | 152  00  00 | 211  75  -
035  42 STO | 094  65  ×  | 153  85  85 | 212  43 RCL
036  05  05 | 095  53  (  | 154  43 RCL | 213  00  00
037  85  +  | 096  43 RCL | 155  07  07 | 214  33 X²
038  43 RCL | 097  06  06 | 156  42 STO | 215  65  ×
039  03  03 | 098  75  -  | 157  05  05 | 216  02  2
040  95  =  | 099  43 RCL | 158  43 RCL | 217  03  3
041  42 STO | 100  05  05 | 159  11  11 | 218  85  +
042  01  01 | 101  54  )  | 160  42 STO | 219  43 RCL
043  42 STO | 102  85  +  | 161  09  09 | 220  00  00
044  06  06 | 103  43 RCL | 162  61 GTO | 221  65  ×
045  42 STO | 104  05  05 | 163  00  00 | 222  07  7
046  07  07 | 105  75  -  | 164  85  85 | 223  08  8
047  32 X:T | 106  48 EXC | 165  25 CLR | 224  04  4
048  43 RCL | 107  07  07 | 166  35 1/X | 225  75  -
049  02  02 | 108  95  =  | 167  91 R/S | 226  01  1
050  77  GE | 109  29 CP  | 168  43 RCL | 227  02  2
051  00  00 | 110  67  EQ | 169  05  05 | 228  05  5
052  61  61 | 111  01  01 | 170  91 R/S | 229  54  )
053  87 IFF | 112  87  87 | 171  14  D  | 230  75  -
054  01  01 | 113  33 X²  | 172  76 LBL | 231  01  1
055  01  01 | 114  42 STO | 173  17 B'  | 232  02  2
056  65  65 | 115  08  08 | 174  24 CE  | 233  08  8
057  86 STF | 116  43 RCL | 175  67  EQ | 234  95  =
058  01  01 | 117  07  07 | 176  01  01 | 235  92 RTN
```

Keeping It Simple continued from page 1

final decision has been made to exchange or not.

Each iteration of the sort is called a 'pass'. The above discussion is of the first pass. After a little reflection it becomes clear that the largest value in the list will reside in the last place in the list after the first pass. This should hint that it is not necessary to include this value in future passes, as it will not need to be exchanged.

The second pass must now check one less pair of values for the possibility of exchange. The comparisons are made for the second pass, and the next largest value in the list will appear in the next to last position in the data list. Similarly, the third pass will have still one less pair of values to check and will end with the third highest value in the third from the last position in the list.

When the final pass is made, there will only be one comparison left to make. The first and second values are compared and exchanged if necessary completing the sort. The characteristic of the bubble sort is that the smallest value rises to the top of the list by stages as the larger values sink to the bottom. It is the behavior of the smaller values that gives this sort its name. This particular implementation of the bubble sort sorts the values in ascending order. To sort the values in descending order using the bubble sort, the order of the values are switched if the first value is smaller than the second.

It would be desirable for the program to stop when the list is sorted whether or not the last pass has been executed. For instance, it would be inefficient to execute a second pass if no exchanges have taken place in the first. It would be equally inefficient to continue after any pass resulted in no exchanges. An intelligent program would incorporate such a check.

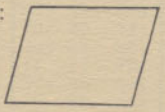
The program should now be sufficiently defined. The next step is to draw a flow-chart using the familiar program symbols. Each of the symbols used is given below along with its name and a short description.

The symbols used in the flowchart below conform to the International Organization for Standardization (ISO) International Standard 1028 - "Information Processing - Flowchart Symbols and Their Usage in Information Processing," ANSI X3.5 - 1970.

Process:

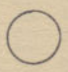
Any processing function; defined operations causing change in value, form, or location of information.

Input/Output:



General I/O function; information available for processing (input), or recording or processed information (output).

Connector:

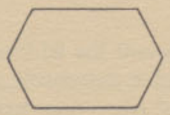


Exit to, or entry from another part of chart.

Arrowheads and Flowlines:

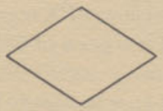
In linking symbols, these show operations sequence and dataflow direction. Arrowhead required if path on any linkage is not left-to-right or top-to-bottom.

Preparation:



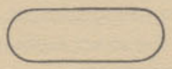
Instruction modification to change program - set a switch, modify an index register, initialize a routine.

Decision:



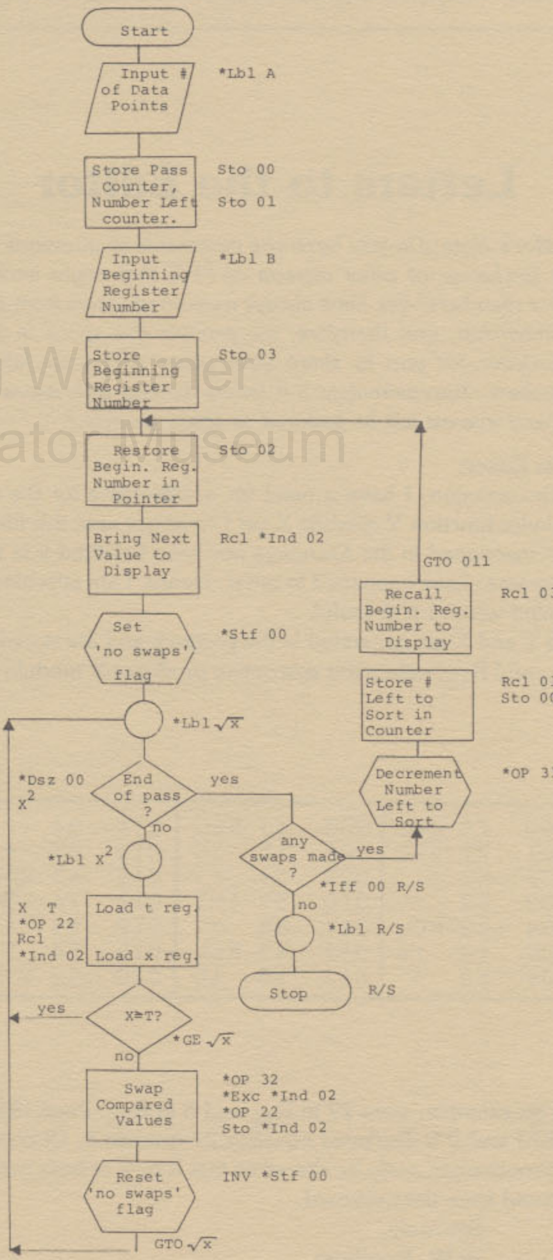
A decision or switching -type operation that determines which of a number of alternative paths followed.

Terminal, Interrupt:



A terminal point in a flowchart. Start, stop, halt, delay, or interrupt.

The development of the flowchart proceeds as a step-by-step solution of the already defined problem. Keystrokes appear beside symbols for clarity.



continued on page 8

Base Converter continued from page 1

Key in the two programs and record each program on bank one of a magnetic card.

ENTER	PRESS	DISPLAY	COMMENT
Read in bank one of card one.			
3256719	A	3256719.	N_{10}
16	B	16.	Target base
	C	0.	Calculate N_{16}
	D	0.	Display first digit
	R/S	0.	Display second digit
	R/S	0.	Display third digit
	R/S	0.	Display fourth digit
	R/S	3.	Display fifth digit
	R/S	1.	Display sixth digit
	R/S	11.*	Display seventh digit
	R/S	1.	Display eighth digit
	R/S	8.	Display ninth digit
	R/S	14.**	Display tenth digit
	R/S	"9.999999 99"	Task complete

The complete answer is $31B18E_{16}$.

The reverse process is accomplished as follows:

ENTER	PRESS	DISPLAY	COMMENT
Read in bank one card side two.			
	A	0.	Initialize
0	B	0.	Digit 1
0	R/S	0.	Digit 2
0	R/S	0.	Digit 3
0	R/S	0.	Digit 4
3	R/S	3.	Digit 5
1	R/S	1.	Digit 6
11	R/S	11.*	Digit 7
1	R/S	1.	Digit 8
8	R/S	8.	Digit 9
14	R/S	14.**	Digit 10
16	C	16.	Source base
	D	3256719.	n_{10}

*Decimal representation for hexadecimal B

**Decimal representation for hexadecimal E

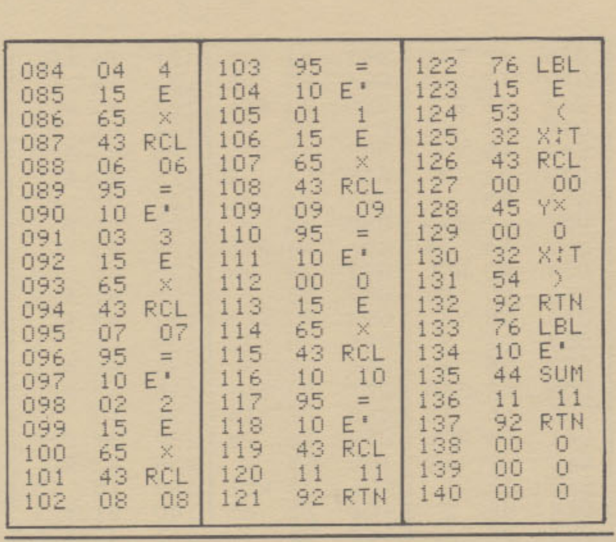
PROGRAM 1
From base 10 to desired base

```
000  76 LBL   015  09   9   030  95   =
001  11   A   016  95   =   031  42 STO
002  42 STO   017  42 STO   032  08  08
003  02  02   018  04  04   033  43 RCL
004  92 RTN   019  43 RCL   034  03  03
005  76 LBL   020  03  03   035  45 Y^x
006  12   B   021  45 Y^x   036  06   6
007  42 STO   022  08   8   037  95   =
008  03  03   023  95   =   038  42 STO
009  92 RTN   024  42 STO   039  10  10
010  76 LBL   025  06  06   040  43 RCL
011  15   E   026  43 RCL   041  03  03
012  43 RCL   027  03  03   042  45 Y^x
013  03  03   028  45 Y^x   043  05   5
014  45 Y^x   029  07   7   044  95   =
```

```
045  42 STO   083  42 STO   121  76 LBL
046  12  12   084  25  25   122  14   D
047  43 RCL   085  43 RCL   123  43 RCL
048  03  03   086  02  02   124  05  05
049  45 Y^x   087  42 STO   125  91 R/S
050  04   4   088  24  24   126  43 RCL
051  95   =   089  76 LBL   127  07  07
052  42 STO   090  95   =   128  91 R/S
053  14  14   091  43 RCL   129  43 RCL
054  43 RCL   092  24  24   130  09  09
055  03  03   093  55   ÷   131  91 R/S
056  45 Y^x   094  73 RC*   132  43 RCL
057  03   3   095  25  25   133  11  11
058  95   =   096  95   =   134  91 R/S
059  42 STO   097  59 INT   135  43 RCL
060  16  16   098  72 ST*   136  13  13
061  43 RCL   099  26  26   137  91 R/S
062  03  03   100  65   ×   138  43 RCL
063  45 Y^x   101  73 RC*   139  15  15
064  02   2   102  25  25   140  91 R/S
065  95   =   103  95   =   141  43 RCL
066  42 STO   104  94 +/-   142  17  17
067  18  18   105  44 SUM   143  91 R/S
068  43 RCL   106  24  24   144  43 RCL
069  03  03   107  02   2   145  19  19
070  42 STO   108  44 SUM   146  91 R/S
071  20  20   109  25  25   147  43 RCL
072  01   1   110  44 SUM   148  21  21
073  42 STO   111  26  26   149  91 R/S
074  22  22   112  43 RCL   150  43 RCL
075  92 RTN   113  25  25   151  23  23
076  76 LBL   114  32 X:T   152  91 R/S
077  13   C   115  03   3   153  25 CLR
078  15   E   116  22 INV   154  35 1/X
079  05   5   117  77  GE   155  91 R/S
080  42 STO   118  95   =   156  25 CLR
081  26  26   119  25 CLR   157  61 GTO
082  04   4   120  91 R/S   158  14   D
```

PROGRAM 2
From given base to base 10

```
000  92 RTN   028  07  07   056  08   8
001  76 LBL   029  91 R/S   057  15   E
002  11   A   030  42 STO   058  65   ×
003  47 CMS   031  08  08   059  43 RCL
004  25 CLR   032  91 R/S   060  02  02
005  29 CP    033  42 STO   061  95   =
006  81 RST   034  09  09   062  10 E'
007  76 LBL   035  91 R/S   063  07   7
008  12   B   036  42 STO   064  15   E
009  42 STO   037  10  10   065  65   ×
010  01  01   038  91 R/S   066  43 RCL
011  91 R/S   039  25 CLR   067  03  03
012  42 STO   040  35 1/X   068  95   =
013  02  02   041  91 R/S   069  10 E'
014  91 R/S   042  76 LBL   070  06   6
015  42 STO   043  13   C   071  15   E
016  03  03   044  42 STO   072  65   ×
017  91 R/S   045  00  00   073  43 RCL
018  42 STO   046  92 RTN   074  04  04
019  04  04   047  76 LBL   075  95   =
020  91 R/S   048  14   D   076  10 E'
021  42 STO   049  09   9   077  05   5
022  05  05   050  15   E   078  15   E
023  91 R/S   051  65   ×   079  65   ×
024  42 STO   052  43 RCL   080  43 RCL
025  06  06   053  01  01   081  05  05
026  91 R/S   054  95   =   082  95   =
027  42 STO   055  10 E'    083  10 E'
```



```
084   04   4      103   95   =      122   76 LBL
085   15   E      104   10 E'       123   15   E
086   65   ×      105   01   1      124   53   (
087   43 RCL      106   15   E      125   32 X⇂T
088   06  06      107   65   ×      126   43 RCL
089   95   =      108   43 RCL      127   00  00
090   10 E'       109   09  09      128   45 Y×
091   03   3      110   95   =      129   00   0
092   15   E      111   10 E'       130   32 X⇂T
093   65   ×      112   00   0      131   54   )
094   43 RCL      113   15   E      132   92 RTN
095   07  07      114   65   ×      133   76 LBL
096   95   =      115   43 RCL      134   10 E'
097   10 E'       116   10  10      135   44 SUM
098   02   2      117   95   =      136   11  11
099   15   E      118   10 E'       137   92 RTN
100   65   ×      119   43 RCL      138   00   0
101   43 RCL      120   11  11      139   00   0
102   08  08      121   92 RTN      140   00   0
```

Keeping It Simple continued from page 6

*Lbl A — As discussed in the last newsletter, the user defined keys make data input easy. The display value is made available to the first program step following Lbl A. The input is the number of data points to be sorted.

Sto 00 — Stores the number of data points to be sorted in the present pass in register 0. This register contains the number of values still left to compare within the present pass.

Sto 01 — Stores the number of data points for the next pass in register 1. This register contains the number of total values in the present pass.

R/S — Stops the program to allow entry using user-defined label B.

*Lbl B — User defined key entry of the beginning register number.

Sto 03 — Stores the entered value (beginning register number) permanently in register 3 for future reference.

Sto 02 — Stores the display value for use as a pointer in the comparison loop and exchange block.

Rcl *Ind 02 — Recalls the contents of the register pointed to by register 2 into the display.

*Stf 0 — Initialize 'no exchanges made' flag.

*Lbl $\sqrt{x}$ — Common label, used as beginning of inner loop within each pass.

*Dsz 0 X^2 — Decrements register 0 by 1, tests register 0. If register 0 does not contain 0, execution of the program transfers to the step following Lbl X^2. Otherwise, program execution continues to the next step following the Dsz 0 X^2 block.

*Iff 0 R/S — Tests flag 0. If flag 0 is set, execution of the program transfers to the step following Lbl R/S. Otherwise, program execution continues to the next step following the Iff 0 R/S block.

*Op 31 — Decrement counter register 1. One less comparison is left to be made.

Rcl 01 Sto 00 — Stores number of comparisons to be made in the next pass in register 0.

Rcl 03 — Recalls the contents of register 3 (the beginning register number) to the display.

Gto 011 — Transfer to the absolute address (step 11) where the display is stored into the pointer register. This starts a new pass.

*Lbl X^2 — Common label marking the beginning of the comparison block.

X ≷ T — The display will contain the value corresponding to the register pointer 2. X ≷ T stores the contents of the display in the T register.

*Op 22 — Increment the register pointer 2 by 1.

Rcl *Ind 02 — Move the contents of the register pointed to by the contents of the pointer register 2 to the display.

*GE $\sqrt{x}$ — Compare X (display) register with the T register. If X is greater than or equal to T, execution of the program transfers to the step following Lbl $\sqrt{x}$. Otherwise, program execution continues to the next step following the GE $\sqrt{x}$ block, where the values are exchanged.

*Op 32 — Beginning of the exchange block. The display contains value to be exchanged. The pointer register will now point to the register where the value is to be stored.

*Exc *Ind 02 — The lower of the two values is stored in the register pointed at by register 2, and the higher value (in the register pointed at by register 2) is brought into the display.

*Op 22 — The pointer register will now point to the register where the greater of the two values will be stored.

Sto *Ind 02 — The greater of the two is stored in the proper register.

Inv *Stf 0 — Clears flag 0. An exchange has been made.

Gto $\sqrt{x}$ — Go back to label $\sqrt{x}$ to continue the present pass.

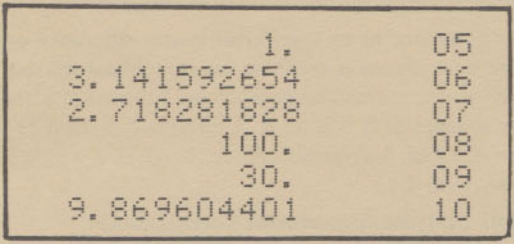
*Lbl R/S
R/S — Label R/S is used to exit the program from the 'no exchanges made' check.

(* denotes 2nd function)

The LED display of your TI59 is controlled by the contents of a memory in the calculator. This memory is called the display register. For instance, when the key strokes [STO] [01] are entered into the calculator, the content of the display register is stored in register 1. When the calculator is running a program in memory, the LED display will not contain the contents of the display register (unless a PAU instruction is encountered or the PAU key is held down on the keyboard) but the contents of the display register is still available to the program. For instance, when the instruction [RCL] [00] is encountered in the program, the contents of the register 0 is transferred to the display register.

SAMPLE PROBLEM

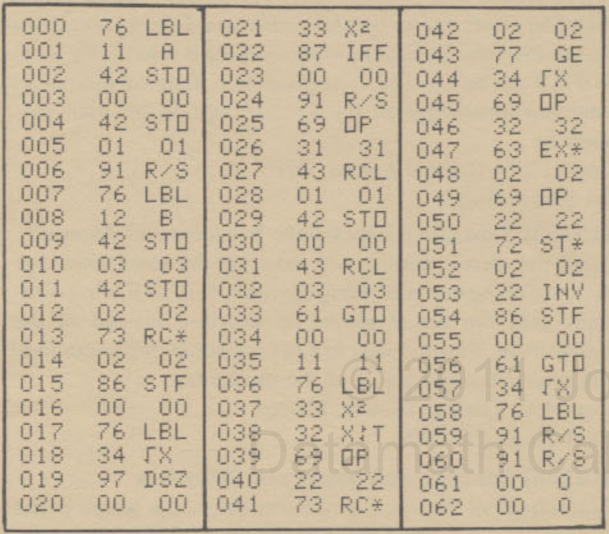
1) Load the following values in the registers 5-10 by using the program from the previous newsletter or by using the STO instruction.



```
          1.        05
3.141592654         06
2.718281828         07
        100.        08
         30.        09
9.869604401         10
```

2) Enter the number of points to be sorted and press A.
3) Enter the starting register number and press B.
4) The points are now sorted and can be found in ascending order in registers 5-10.

Note: The lowest register available is register 4. Registers 0-3 are reserved by the sorting program and are not available for data storage.



```
000  76 LBL    021  33 X²     042  02  02
001  11  A     022  87 IFF    043  77  GE
002  42 STO    023  00  00    044  34 ΓX
003  00  00    024  91 R/S    045  69 OP
004  42 STO    025  69 OP     046  32  32
005  01  01    026  31  31    047  63 EX*
006  91 R/S    027  43 RCL    048  02  02
007  76 LBL    028  01  01    049  69 OP
008  12  B     029  42 STO    050  22  22
009  42 STO    030  00  00    051  72 ST*
010  03  03    031  43 RCL    052  02  02
011  42 STO    032  03  03    053  22 INV
012  02  02    033  61 GTO    054  86 STF
013  73 RC*    034  00  00    055  00  00
014  02  02    035  11  11    056  61 GTO
015  86 STF    036  76 LBL    057  34 ΓX
016  00  00    037  33 X²     058  76 LBL
017  76 LBL    038  32 X⇄T    059  91 R/S
018  34 ΓX     039  69 OP     060  91 R/S
019  97 DSZ    040  22  22    061  00   0
020  00  00    041  73 RC*    062  00   0
```

potpourri

- We are finding that many members are not receiving their newsletters and catalog updates. The reason, more often than not, is that members have had a change of address which has never been forwarded to us. The newsletters and addendums to the catalog are mailed third class bulk rate, and are therefore returned to PPX when the labeled address is not correct. As mailing costs continue to rise, we at PPX are making every effort to refrain from passing these costs on to you, but we need your help. Please notify our membership coordinator at P.O. Box 109 as soon as possible with any change of address.
- We would like to remind our international members that we can accept payment in U.S. Dollars only. All checks and money orders must be drawn on a United States bank or a U.S. correspondent bank. The orders that we receive in any other currency will be returned unfilled, and to avoid the inconvenience and time lag, we suggest that you send all international payments in money orders.
- In recent months, we have noticed a sizeable increase in the number of orders which have not included the required $2.00 postage and handling fee and applicable state and local taxes. PPX is required by state and local regulations to pay taxes in all states except Alaska, Montana, New Hampshire, Oregon, and Delaware. We strive at all times to keep our costs down so that we may offer our members, at the lowest possible price, membership in the Exchange, programs, and accessories. When members neglect to pay the postage and handling fees and their state and local taxes, which we must pay regardless, it hurts everyone. Beginning immediately, all orders which do not include the postage and handling fee and all applicable state and local taxes will be returned unfilled requesting that the required amounts be remitted.
- In the transition to our new automated system, there have been many duplicate membership renewal cards sent to our members. If you have renewed your membership and receive a duplicate renewal, please ignore it. We are working on this problem, and hope to have it remedied in the near future.

Précis

This column presents the abstracts of some of the new PPX programs which have been recently accepted. The programs were selected by our analysts as being ones that would be of special interest to our members. You can purchase these programs at a cost of $4.00 each. Send your order to: Texas Instruments: PPX Department, P.O. Box 109, Lubbock, TX 79408. Include an additional $2.00 for postage and handling plus applicable state tax.

If you have a need for a specific program, send a note to PPX. There is a chance that the program may have already been written. If it has, we will put the abstract in the next issue of the Exchange. Requests for programs not yet written will be placed in the "Programming Corner" column.

058013I Weekly Payroll Deductions Hawaii (Single)
Calculates and prints deductions (FICA, Federal Withholding, Hawaii State Withholding, and three more user defined and labeled deductions) for a weekly payroll period for single employees including heads of households.
H. Doug Matsouka, Honolulu, HI
797 Steps, PC-100A

058014I Weekly Payroll Deductions Hawaii (Married)
Calculates and prints deductions (FICA, Federal Withholding, Hawaii State Withholding, and three more user defined and labeled deductions) for a weekly payroll period for married employees.
H. Doug Matsouka, Honolulu, HI
792 Steps, PC-100A

128025I Statement Savings Daily Compounding (ATM)
Most Automatic Teller Machine banking systems compound interest daily but credit it only monthly. This program checks all bank statement entries.
Serge Borodin, Brooklyn, NY
315 Steps, PC-100A, Mod 1

128026I ATM Checking Account Random Reconciliation
Automatic Teller Machines display checking account balances but do not tell you which of your outstanding transactions have been processed. This program considers all possible combinations of transactions using a binary sequence.
Serge Borodin, Brooklyn, NY
404 Steps, PC-100A

148021I Consumer Price Index Conversion II
Table of CPI's as of the end of the month, store, check, correct, and update. The CPI can be calculated for any date in the table, and dollars can be converted from current dollars to constant dollars.
John E. Binns, Stuart, FL
396 Steps, Mod 1

218072I ANOVA for Repeated of Unrepeated Lattices
Handles the ANOVA for simple lattice designs (whether the basic design is repeated or not) containing up to 64 treatments or 64 blocks and any number of replications (r) and plots per block (k). Input: parameters, raw data according to blocks in the field plan, treatment totals, block number and treatment number, C-values (printed earlier by program), and replication totals. Output: A, B, and C values and complete ANOVA (including degrees of freedom, sums of squares and mean squares). An optional part provides adjusted treatment totals and means.
Abdollah Bassiri, Shiraz, Iran
479 Steps, PC-100A

228073I Bonferroni Correction (Multiple Corrections)
Bonferroni Correction of Student's T in multiple comparisons is not valid statistically. This method is used after an Analysis of Variance and provides a correction for the fact that multiple comparisons are being made. The data must be reentered after the analysis of variance for each sample desired, or a complete listing of the data points is needed for each comparison otherwise. If the mean for each group is known, it can be entered directly, saving several steps.
Guy H. Nelson, Richmond, VA
786 Steps, PC-100A, Mod 2

248007I Random Date Generator
User enters first and last dates of the population period and a random seed number. Program computes desired number of random dates within the period. If desired, user may opt to have the program perform any one or more of the following operations: reject all Saturdays, reject all Sundays, or reject up to 35 user-defined holidays.
Barbara C. Hevener, Columbia, SC
277 Steps, Mod 1

358023I Continuous State Equations (Discrete Form)
Converts a coupled set of linear, constant-coefficient differential equations to an equivalent matrix difference equation. The equivalence is exact for the case where the driving functions are piecewise-constant inputs. Program will handle systems as large as 8 X 8 on a TI-59 or 5 X 5 on a TI-58.
Stephen J. Gold, Lafayette, LA
80 Steps, Mod 1

788066I Ecliptic/Equatorial Conversions
Easily convert ecliptic coordinates to equatorial or from equatorial to ecliptic.
John J. Garner, Grand Portage, MN
256 Steps

418131I Balancing Chemical Equations
Balances chemical reactions by matrix methods and converts non-integer quantities of the initial reactants and final products in the reaction to integer quantities. Program has subroutines to eliminate the effect of near zero values due to rounding errors and to by-pass zero value during the conversion to integers. Equations which require a matrix size up to an 8 X 8 can be handled. The procedure does not require an extensive knowledge of the applied field to be used successfully.
Robert C. McQuattie, Lorain, OH
234 Steps, Mod 1

418132I Complete Electronic Structure of Atoms
Given the atomic number will print out the exact electronic structure of the corresponding element. (S,P,D,F, etc.). Program uses the actual electronic structure instead of the hypothetical, thus the user can predict some of the chemical and physical properties of all the elements.
Jose M.G. Garcia, Tijuana, Mexico
1038 Steps, PC-100A

418133I Specific and Molecular Rotation
By using either the volume of the solution, the length of the polarimeter tube, the concentration, the rotation observed, the molecular weight or the specific rotation, calculates any of the following: grams in the solution, the specific rotation or the molecular (or molar) rotation. Printer is optional.
Jose M.G. Garcia, Tijuana, Mexico
195 Steps

418134I Density as a Function of the Temperature
Given the temperature in degree celsius, calculates the density of the following solvents: acetone, acetic acid, benzene, toluene, phenol, n-propanol, iso-propanol, chloroform, carbon tetrachloride, and ether. Printer is optional.
Jose M.G. Garcia, Tijuana, Mexico
125 Steps, Mod 1

418135I Molar Polarization
Given the molecular weight, the density and the dielectric constant, this small program will calculate the molar polarization. Printer is optional.
Jose M.G. Garcia, Tijuana, Mexico
130 Steps

418136I Water Properties IX: Liquid Surface Tension
Given the temperature, calculates the surface tension of liquid water and is more accurate than graphical methods.
Jose M.G. Garcia, Tijuana, Mexico
202 Steps, PC-100A, Mod 1

418137I Water Properties X: Liquid Termal Conductivity
Calculates the thermal conductivity of liquid water when the temperature is given.
189 Steps, PC-100A, Mod 1

588006I Diet Calculation for Kidney Patients
Using nutritional information stored on magnetic cards, calculates the content of those food elements of most interest to patients with chronic kidney failure, namely, water, protein, calories, carbohydrate sodium, potassium, calcium, and phosphorus. The protein chemical score (and index of protein quality) is computed by the method of Block and Mitchell (*Nut. Abst. Rev.* 16: 249-278, 1946). Individual scores of the essential amino acids are provided as well. Computation is on a meal-by-meal basis.
Stephen Dubin, Philadelphia, PA
612 Steps, PC-100A, Mod 1

668191I Boiler Efficiency, Fuel to Pound Steam
Calculation of the boiler efficiency by using the enthalphy of steam, boiler water, and feed water. Input: pound of steam produce and BTU input of gas, oil, or both. Output: boiler efficiency, cost of gas, cost of oil, total cost, and cost per 1000 pounds steam.
Roland R. Cameron, Ft. Wayne, IN
254 Steps, PC-100A

668192I Fusion Temperature of Coal Ash
In selecting coals for steam generators, it is important to consider the characteristics of the ash formed in the combustion process. The ash fusion temperature is a characteristic which has the greatest effect on design and daily operations. Knowing the constituents of the ash, the program estimates the ash fusion temperature, the base/acid ratio, and the iron/calcium ratio.
Rober J. Smogor, Jr., Homer City, PA
120 Steps, Mod 1

678019I Shelter Computation
Provides essential data required for shelter management in a nuclear survival situation. Results are based on reading dosimeters to yield RADS absorbed, maximum allowable dose, time allowable outside, protection factor, and inside rates. Printer is optional.
Ross D. Litman, II, Ft. Wainwright, AK
133 Steps

698036I Loads Due to Pitch and Roll Motions of a Ship
Given the ship's motion behavior characteristics (center of rotation, pitch angle, pitch period, roll angle, and roll period) and the weight and center of gravity of the object being investigated, calculates both the individual horizontal and vertical load components and the total resultant horizontal and vertical loans due to the ship's pitch and roll motions. Also calculated is the direction of the total resultant horizontal load.
John F. Hancock, South Weymouth, MA
596 Steps, PC-100A

708013I Percussive Skin Formulae
Computes frequency for plate (m and n) modes. Makes a neat mathematics game. It uses digression on rectangular plate frequency theory. Using stiffness and damping coefficients, it also calculates damped oscillation factors. The music student, acoustician or instrument designer/manufacturer will be able to alter modes without manually graphing a multitude of mathematical manipulations. Printer is optional.
Ross D. Litman, II, Ft. Wainwright, AK
131 Steps

788067I Satellite of Jupiter
Calculates the configuration of Jupiter's four satellites: Io, Europa, Ganymede, and Callisto relative to Jupiter for any date at any time. Also can plot the satellites' positions at that time for use in observations, or it can plot the satellites positions at twelve hour intervals for any specified number of days to graphically show the satellites' motion around Jupiter.
S.T. Bradley, Coucil Bluffs, IA
847 Steps, PC-100A, Mod 1

868023I SAE Grade Oil Viscosities and Specific Gravities
By entering the temperature in degree Fahrenheit and the SAE grade of oil (5,10,20,30,40, or 50), calculates the absolute viscosity in Reyn from the kinematic viscosity in Saybolt Universal Second (SUS). For the given temperture in Reyn, specific gravity, and kinematic viscosity in centistokes will also be calculated.
Danny L. Luey, Flushing, NY
114 Steps

918317I Checkers 3-Move Generator
Will select an opening at random to begin a checker game. Any one of 142 American Checker Federation approved openings can be generated or selected eliminating the need for a 3-move deck. A recommended response is also given for each opening.
John R. Gibson, Colorado Springs, CO
160 Steps, Mod 10

998052I Dovetail Joint Measurements
Program provides dimensions for marking the cuts to be made for joining wood with dovetail joints. Allows for regular or irregular spacing of the pins and tails, depending on the esthetic effect desired.
Robert S. McGihon, Alexandria, VA
457 Steps, PC-100A

The PPX Exchange is published bimonthly and is the only newsletter published by Texas Instruments for TI-59 owners. Members are invited to contribute articles and items of general interest to other TI-59 users. Authors of accepted feature articles for the newsletter will receive their choice of either a one year complimentary PPX membership or a Solid State Software^{TM} module. Please double-space and type all submissions, and forward them to:

Texas Instruments, PPX
P.O. Box 53
Lubbock, Texas 79408
Attn: PPX Exchange Editor

Classified Ads

If you have something to sell or trade, or if you are in need of an item that you are unable to locate commercially, PPX will be offering to PPX members a classified ad section for hardware only. With over 14,000 members, both foreign and domestic, PPX can offer you an outlet for advertising your equipment. We will publish a minimum of five ads per newsletter, and the cost to the member is 15^{c} per word with 100 words per ad as the maximum. This price applies to a single issue of the newsletter, and you are welcome to submit your ads as often as you like. Send your ads in care of the PPX Exchange Editor, P.O. Box 109, Lubbock, Texas 79408 and be sure to include your membership number on all ads.

Membership Renewals

Is your membership about to expire? To ensure that you will miss no newsletters, catalogs, or ordering privileges, check the renewal table to find out if your membership will expire soon. (If your number is not included in the range of the table, it is not time for you to renew). The next issues of the Exchange will list additional renewal dates.

A renewal card and reminder will be sent to each member before the time to renew. Return the card promptly to PPX with your check or money order for $20.00. Please do not procrastinate in returning your renewal material as our membership coordinator must remove delinquent members from our computer listing. Be sure to include your membership number on both your card and your check and mail to: Texas Instruments PPX Department, P.O. Box 109, Lubbock, TX 79408.

Membership Number	Renewal Due
910896-911973	November 30
921595-922334	November 30
928271-928718	November 30
932936-933132	November 30
900001-901982	December 31
911974-912576	December 31
922335-922787	December 31
928719-929148	December 31
933133-933485	December 31

TI-59 Programming Seminar

There may be a seminar coming to your area. These seminars are open to anyone with a TI-59 regardless of programming background. The seminars provide both beginning and intermediate programming training on the TI-59 in a "hands on" fashion. Tuition for the two day class is $150.00 per person. This includes the instruction, workbook, and luncheon for the two days. You should supply your own TI-59. To register send your check for $150.00 payable to Texas Instruments to:

TI-59 Seminar
Texas Instruments
P.O. Box 10508 MS 5820
Lubbock, TX 79408

If you have any further questions regarding the seminars or if you would like information on setting up a company seminar, please contact Professional Calculator Division at 806-741-2202. The schedule of the upcoming seminars is listed below.

Seminar Dates	Location
October 5-6	Chicago, IL
October 11-12	Detroit, MI
October 14-15	Salt Lake City, UT
October 21-22	Milwaukee, WI
October 25-26	Montreal, Canada
October 28-29	Rochester, NY
November 4-5	Albany, NY
November 11-12	Philadelphia, PA
November 18-19	Birmingham, AL
November 22-23	Akron, OH
December 2-3	Indianapolis, IN
December 9-10	New York City, NY

TEXAS INSTRUMENTS
INCORPORATED
PPX • P.O. Box 53 • Lubbock, Texas 79408
U.S. CALCULATOR PRODUCTS DIVISION

ADDRESS CORRECTION REQUESTED

BULK RATE
U.S. POSTAGE
PAID
Permit No. 476
Lubbock, Texas