PPX EXCHANGE

Vol. 3 Number 5 Copyright 1979 September 1979

Do your Christmas shopping early. PPX is again offering The Sourcebook for Programmable Calculators at a sale price of $9.95 (compared to $12.95) through the end of December. This ideal gift provides excellent examples of problem solutions and serves as a TI-59 learning aid. Topics which are written on the college level include: Business and Operations Research; Statistics; Mathematics; Biomedical, Electrical, and Systems Engineering; Music Theory; Economics; Biology; Physics; and Astronomy. To order a copy, enter "Sourcebook" on your PPX-59 order form and include your **check or money order (only)** for $9.95, plus $1.00 postage and handling, and applicable sales tax. Order now as this offer ends January 31, 1979. **This offer is good only in the U.S., Canada, and Mexico.**

MEMBERSHIP RENEWALS

Is your membership about to expire? To ensure continued receipt of Addendums, newsletters, and ordering privileges, make certain that this is not the case.

Below is printed the renewal table for members whose one year memberships will soon expire. To find your renewal date, check your membership number against the table shown below. Your membership number corresponds with your required renewal date.

Membership number	Must be postmarked by
59910094-59910894	September 15
59910895-59912576	November 15
59912577-59913803	December 15

Members with numbers greater than those listed above will be informed of their renewal dates in a future issue of the PPX Exc hange.

A renewal subscription card and reminder will be sent to each member in ample time to renew. The subscription card must be returned with a check or money order for $15. Be sure to include your membership number on both your subscription card and check.

PPX-59 PROGRAMMING CORNER

This column is devoted to PPX-59 programming suggestions. If you have a program(s) that you would like to see made available through PPX-59, send your suggestions to PPX. In this way, members who enjoy programming are made aware of your programming needs. PPX-59 is not staffed to do custom programming; therefore, member suggested programs will become available only if another member of PPX-59 comes to the rescue.

Our members would like to see:

- Put Option programs (buying and selling).
- A program giving lagged cross correlations between x(t) and y(t) for lags of +/-6 or more (an autocorrelation option for x(t) would also be useful).
- A program to pay Backgammon.

Now available . . . More specialty Software for your TI-59 calculator. The following software packages have been written by specialists in their respective fields. Order these packages directly from the addresses shown below. **These items are not available through PPX, nor does Texas Instruments manufacture or guarantee them in any way.** PPX hopes the following information may be helpful to you.

- Attention solar energy architects, engineers, and consultants! The first book covering solar energy calculations by Dr. R. W. Graeff, has been published by Solarcon, Inc. The book describes over 40 typical solar energy calculations from solar radiation and collector efficiencies to system evaluation. Included in these descriptions are the user instructions and sample problem output of Solarcon's programmed solution. (The actual program must be ordered directly from Solarcon.) Also included in *Solar Energy Calculations* are other chapters which cover how to select commercially available programs, how to use them most efficiently, and how to write your own problems.

The price of *Solar Energy Calculations* is $25. Each copy includes a $20 coupon, redeemable when buying any Solarcon program. The book can be ordered from:

Dr. Roderich W. Graeff
Solarcon, Inc.
607 Church Street
Ann Arbor, MI. 48104
Tele: (313) 769-6588

- "Forecasting Computer System Reliability with a Handheld Programmable Calculator" by PPX member Ronald Zussman appeared in the March 1979 issue of Computer Design Magazine. This article addresses the understanding, evaluation and improvement of computer hardware reliability using the TI-59. Instructions and Program listings for the TI-59 are included. Free copies of this article and an accompanying appendix are available to PPX-59 members who write to:

The Editor
Computer Design
11 Goldsmith St.
Littleton, Ma. 01460

PPX POTPOURRI

1. **Program Memos.** For those PPXers who have never used a PPX Program Memo form, let us explain what they are and how they are used. This form is included with every order filled by PPX-59. It is intended as a convenient vehicle by which PPX receives both positive and negative user comments on programs ordered. If you receive a program which you feel is extraordinarily good, send us your comments and we will be glad to pass your compliments to the

author. When you send a program memo to PPX be sure to put your membership number and correct address on it. This will help to insure a prompt reply to any problems which you have encountered with the program.

2. **Utility is in the eyes of the beholder.** Are you about to submit a program whose function can be done as easily without using a program or is unlikely to be of interest to others? Unfortunately, we have recently noticed an increase in the number of programs returned to the authors due to "questionable utility". Prior to the submission see page 3 of your Member's Guide to determine if your program meets all of the acceptance criteria of PPX-59. Admittedly, there is a certain amount of room for analyst subjectivity in the area of whether or not the "utility" criteria has been met. As this is one of the most difficult decisions an analyst must make, we ask that each author become familiar with submission acceptance criteria.

3. **New Members:** If you would like to reinforce your knowledge of those subjects presented in the TI-59 Personal Programming Manual, PPX has the answer for you. The **"TI-59 Workbook"** offers sample problems and corresponding exercises (specifically referencing each exercise with pages in the Personal Programming Manual). This book is written on an **elementary** level. To obtain your copy, enter **"TI-59 Workbook"** on your PPX-59 order form and enclose $4.95, plus tax and handling.

4. **No purchase orders, please!** We are receiving numerous purchase orders. Please be advised that PPX does not accept purchase orders for any reason as we are unable to invoice our members. All transactions **must** be prepaid with a check or money order.

5. **International members —** PPX has received several letters recently concerning the difficulty of sending payment with an order. As stated above, all orders must be prepaid. International members should use a personal check or international money order. The order volume we experience makes us unable to match a payment with an order when they arrive separately. Orders received without payment are immediately returned to the sender.

6. **Our mistake!** The last (blue) page in Addendum D, titled "Programming Systems", should be numbered 7-1. An error was made in printing and the page number was omitted.

7. **Additional listing sheets —** If you find that your program's length exceeds the space provided on the PPX-59 Listing Sheets, please put the remaining steps on white bond paper.

A WALK THROUGH TI-59 HARDWARE

Editor's Note: This article originally appeared in the March 1978 issue of the PPX Exchange. Due to the response we received when it was first released, we thought our newer members would enjoy a look at some of the basic aspects of the TI-59. It is not written as an official design specification and should not be construed as such. It is a rapid walk through the workings of the TI-59.

Although many owners of the TI-59 do not realize it, they possess a tool containing the equivalent of approximately 100,000 Metal-Oxide-Semiconductor (MOS) Transistors. Because the average user lacks the technical background to comprehend the hardware of the TI-59, the following discussion is presented in the hopes that it will further each user's understanding of his TI-59.

All TI-59 hardware is associated with two major functions; Support Functions and Logic Functions (with many subsections included within each):

A. Support Functions
   1. Switching Regulator Power Supply
   2. Clock
   3. Light Emitting Diode (LED) Display with Drivers
   4. Keyboard
   5. Magnetic Card Read/Write Assembly
   6. Read Amplifier
   7. Motor Control Circuit

B. Logic Functions
   1. Scanning Read Only Memory (SCOM)
   2. Read Only Memory (ROM)
   3. Arithmetic Logic Unit (ALU)
   4. Multi-Register Memory
   5. Magnetic Input/Output (Mag I/O) Interface Processor
   6. Constant Read Only Memory (CROM)

The Support Functions provide the MOS Logic with the necessary inputs and outputs for communication with the user. Additionally, they supply the essential power and timing signals required for the operation of the TI-59.

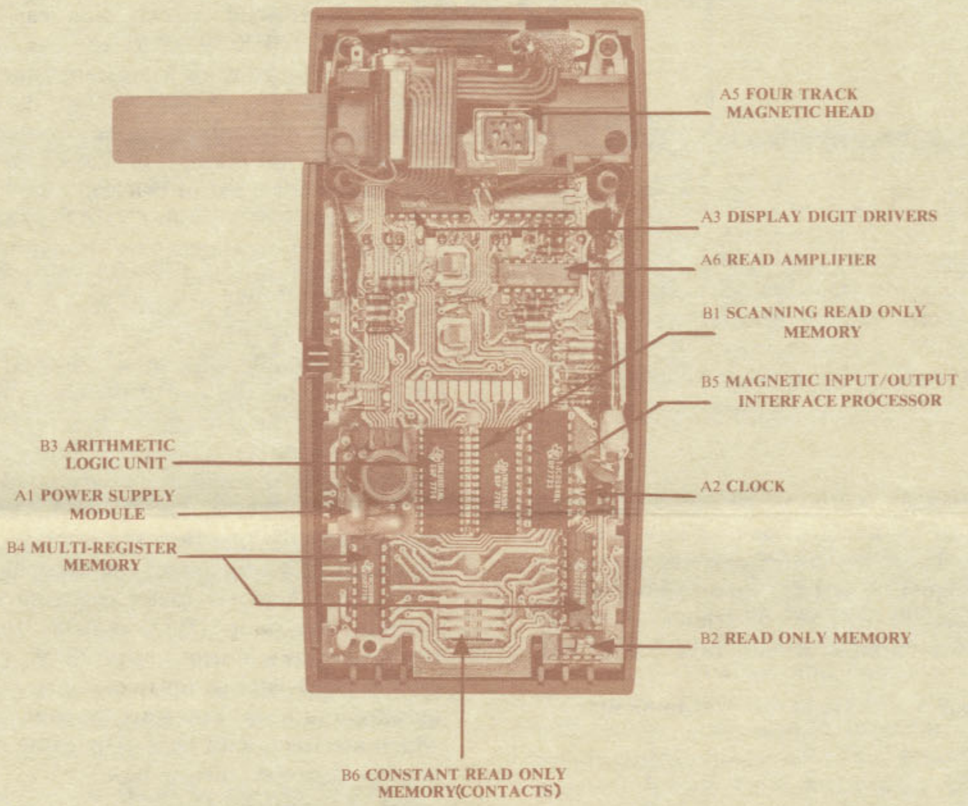
A1. **The Switching Regulator Power Supply** is used to convert the low battery voltage to the higher voltage required by the MOS. The nickel cadmium battery pack provides a relatively constant output voltage of 3.75 volts to the power supply. The power supply then produces two outputs of approximately 10 volts and 16 volts that are distributed to every MOS device as well as the Clock and Read Amplifier.

A2. **The Clock** is the source of the timing signals distributed to all of the logic functions. These signals synchronize the distribution and execution of all logic functions. Additionally, by monitoring the activity occuring in the logic functions, the Clock slows the system down during inactive periods thus increasing battery life.

A3. **A 7-Segment Group of Light Emitting Diodes** form each character in the display. (There are twelve 7-Segment Characters across the display.) Depending on the segments illuminated in each 7-Segment Character, a digit or minus sign is displayed. (For example, all seven segments are illuminated when an 8 appears in the display.) The segment information is provided by the Arithmetic Logic Unit (ALU) and the digit information is provided by the Scanning Read Only Memory (SCOM). By working together, the ALU and SCOM illuminate the character by a process called multiplexing. Because the display voltage differs from the ALU and SCOM voltage, a pair of digit driver integrated circuits provide the level translation.

A4. **The Keyboard (not labeled in the picture)** enables the entry of numbers and functions into the TI-59. This device consists of an x-y matrix of switch contacts attached to a plastic board. The horizontal lines (i.e., x-axis) consist of sequentially addressed digit signals which are used to drive the display digit drivers. The vertical lines (i.e., y-axis) consist of key select lines. When a key is depressed, a digit signal is connected to a key select line. The Arithmetic Logic Unit senses the particular digit signal and decodes the key entry.

A5. **The Magnetic Card Read/Write Assembly** sets the TI-59 apart from most programmable calculators in that it provides a means of permanent program or data storage. A four track magnetic head, similar to that used in tape recorders, is used to read and write data on a card. The head is mounted within a transport housing. The transport housing is molded from a glass filled plastic with very precise tolerances, in some cases to within one thousandth of



an inch. The housing contains a small motor, gearbox, and drive wheel which are used to drive the magnetic card across the read/write head.

A6. **The Read Amplifier** increases the low amplitude magnetic head read signal to allow the Magnetic Input/Output Interface Processor (logical function) to recognize it. The four channel integrated circuit Amplifier provides a signal that is about five hundred times larger in amplitude than the signal originating from the read head.

A7. **The Motor Control Circuit (not labelled in the picture)** maintains the card speed to within a window of 2.1 and 2.5 inches per second. The circuit supplies a constant voltage to the motor under various loads, thus maintaining a uniform card speed as the card passes through the mechanism. This circuit is adjustable so that the card speed will match the requirements of the individual mechanism installed in each calculator. This circuit is made up of discrete components located throughout the calculator.

All necessary computations, manipulations, and data storage are accomplished in the logic functions of the calculator. Ten MOS devices comprise the six distinct logic functions.

B1. **Two Scanning Read Only Memory [SCOM]**, each containing 2.5K words of Read Only Memory (ROM) and 64 words of Random Access Memory (RAM), compose one logic function. The ROM stores most of the TI-59 algorithms. The RAM stores pending operations, t register, key push register, etc. In addition to containing ROM and RAM (memory), the SCOM contains the digit select circuitry with 16 digit time outputs for display multiplexing and key selection.

B2. **The Read Only Memory [ROM]** is programmed with the algorithm overflow from the two SCOM chips. In effect, it is a 1K work extension of the total ROM available in the calculator.

B3. **The Arithmetic Logic Unit [ALU]** is the most complex of the logic functions. The primary task of the ALU is to perform the required mathematical operations using instructions supplied by the SCOM or ROM. Additionally, the ALU provides the direct segment drive to the seven segment display and interprets the keyboard entries appearing on its key select lines.

B4. **Four Multi-Register Memories**, each containing 1920 bits of information, retain the program storage and data memory. Each Multi-Register Memory has thirty shift registers capable of storing a 16 digit number of 4 binary bits. To increase the flexibility of the TI-59, each shift register is capable of storing data or program steps (dependent upon the setting of the partition).

B5. **The Magnetic Input/Output [Mag I/O] Interface Processor** is responsible for data transfer to and from a magnetic card. This device directly controls the magnetic head during write operations and interprets information from the head during read operations. The Mag I/O Interface Processor also corrects and verifies the read information from the mag head and controls the read/write timing (using Clock signals) and motor start function.

B6. **The Constant Read Only Memory [CROM] Chip**, otherwise known as the Solid State Software™ Module, performs the final logic function. This is a ROM that is programmed with software applicable to a specific discipline and capable of being easily removed and replaced. Up to 5,000 program steps (or the equivalent of 40,000 bits of data) can be stored in this compact module.

This ends our walk through the workings of the TI-59. We hope you found it interesting!

ANWENDERFREUNDLICHKEIT

Maurice E.T. Swinnen

If I want someone's attention in the office, I don't use the words "CAUTION" or "ATTENTION" on my signs anymore. People stopped heeding them a long time ago. Instead, I write "Achtung". People smile, say "Jawohl," and read the rest of the sign. Thus, I have achieved my goal: Attention.

In the same vein the "ANWENDERFREUNDLICHKEIT" title of this article is a long but harmless attention-getter. Don't let it scare you. Do you recognize the word "Freund" (i.e., friend) in it? This long title means nothing more than "friendliness towards the user". Which in my opinion, every programmer should have in mind when writing a PPX program.

To illustrate what I mean, I'll give you some examples of "unfriendly" instructions.

1. *Repartition to 239.89.* Veterans of the TI-59 would have no trouble interpreting this instruction. But for those people whose TI-59 is still new to them, it could cause unnecessary trouble. Why not say "Press 9 Op 17"? (Always give clear concise instructions as errors are easily made.) Better yet, why not record all of your magnetic cards with the turn-on partition (479.59) and let each program repartition itself as needed? For example, in an initialization routine, write: Lbl E' 9 Op 17 . . . R/S.

2. *Enter the value for the following variables in this order: time, press D; temperature, press C; pressure, press A'; and volume, press D'.* Note that the user has to remember the order in which the variables are to be entered, as well as the order in which the user defined keys have to be pressed. This results in a lot of unnecessary key pushing and memory work. You can make friends of the user by placing input routines at the end of your initialization routine: Lbl E' . . . R/S STO 01 R/S STO 02 R/S. Input values may now be entered in order by pressing R/S. By placing descriptors alongside each input, "friendliness" to the user again increases. That is, LBL E' R/S STO 01 **37243017 Op 04 RCL 01 Op 06** R/S STO 02 **37173033 OP 04 RCL 02 OP 06** R/S . . .

3. *Enter variable integers 1 through 9 in this manner. Enter "-1.1" for a 1, "-1.2" for a 2, and so on up to 5. For an input value of 6, enter "1.6", for 7 enter "1.7" and so on up to 9.* Here the author's intent was to have a program branch to one of nine possible routines depending upon the input value. Knowledge of the "GTO Indirect" sequence could make the input method much more friendly.

This sequence is:

```
000  76 LBL
001  12  B
002  65  x
003  01  1
004  01  1
005  95  =
006  42 STO
007  08  08
008  83 GO*
009  08  08
```

By examining this sequence, it becomes clear that if you enter a 1, an 11 will be placed in register 08. Then the program will begin execution at location 011. A 2 will send it to location 022, etc.

By writing the appropriate execution routines at locations 011, 022, 033, 044, and so on, the program will execute quickly because of the absence of conditional branching. It will also be economical in program steps. If you need more than 11 steps in your routines, just change the number 11 at locations 003 and 004. Any unused program steps in these routines should be filled with Nops.

4. *Do not forget to enter all input values. If all input values are not entered you will get erroneous results.* True, the user should know enough to always enter all values. But the TI-59 can easily be programmed to protect him from a possible input oversight. By including a decimal point sensor routine under a user defined label, such oversights can be prevented. For example, assuming using label E' as our user defined label, the following "decimal point senser" routine will be executed each time a new variable is entered (using the R/S keystroke): LBL E' . . . R/S . A' STO 01 R/S . x=t A' STO 02 R/S, etc. (It is assumed that program execution is to be initiated by pressing R/S and that there is a zero in the T-register.)

The result of the decimal point sensor routine will be: If you leave the former value in the display (i.e., you forget to enter a new value), the TI-59 will interpret your "non-entry" as a zero, resulting in a branch to A'. Somewhere in the program, define A' as follows (preferrably at the beginning of the program for speed): LBL A' 1/x Prt R/S. The value printed by A' (flashing 9.9999999 99 in this case) will make you notice your omission. Of course, you can always make the A' routine more elaborate. You can print statements ranging from a polite "Please check entry" to a more or less insulting "Watch it, turkey!"

5. *To use this program with a printer, insert the Prt instruction at the appropriate places.* Most programs have enough spare program steps to accommodate one of the many printer sensing routines available. The program will behave "normally" with the TI-59 and halt execution at each R/S so that the user can copy the results. But when the printer is connected to the TI-59, all output values will be printed without halting program execution.

A simple printer sensing routine is this one: LBL E' 20 Op 07 Op 19 CLR . . . R/S. This routine will set flag 7 only if the printer is connected. The output routines can now be written as follows: LBL C . . . STO 09 37243017 Op 04 RCL 09 Op 06 Ifflg 7 EE R/S LBL EE . . . STO 10 . . . Iff 7 CE R/S LBL CE . . .

If the printer is connected, the first output value, with a descriptor, will be printed. Then because flag 7 is set, the program will jump to LBL EE and continue to compute the next output value, print it and jump to LBL CE and so on.

If you want to reduce still more of the needed "button pushing", don't even write LBL C, but omit the last R/S of the input routines and link the C routine to it. Then, after entering your last input value, the program will start computing immediately, saving both execution time and your having to remember to push one more button.

As you can see, it is possible to write a program with "Anwenderfreundlichkeit". To this same end, when you prepare your documentation for submission to PPX, please type or at least hand print everything. Handwritings might be charming or peculiar, just ask any pharmacist. There is nothing more aggravating than trying to decipher somebody elses scrawls. I even have trouble with my own hieroglyphics. So, as a written form of communication, I rigorously stick to typing, even if it is only of the "two-finger" variety.

FROM THE ANALYST'S DESK

• PPX member Elmer B. Clausen of Buffalo, New York, shares a "Number Entry Routing" routine which will solve for the unknown variable in a multivariable formula. The known variables **must be non-zero** as the routine senses whether a variable has been entered prior to pressing an input key. *(Continued on back page)*

TABLE SUMMATION

TABLE SUMMATION

This program adds the number entered into the display register to a value identified by one of the user defined keys (A - E) when the specified user defined key is depressed. This allows the user to sum each row or column of a data table and retain the sums for later processing. The user also has the capability to multiple all sums by a constant and to display the grand total of all the column/row sums. After all the user defined keys have been used to identify a column/row, a redefine function is provided to extend the number of columns/rows available to the user. The user may also return to a previously defined group of five columns/rows. A total number of seventy-eight columns/rows may be processed by this program.

PPX wishes to thank the author of "Table Summation", Herbert D. Rice, for his excellent program.

USER INSTRUCTIONS:

1. Partition to 239.89 by pressing 9 Op 17. Enter program (record on card side 1).
2. Enter register contents R_0 to R_{10} (record on card side 4). The contents of these registers are constants. Enter the constant and then press STO nn, where nn is the appropriate register number.

Constant	Register	Constant	Register
5.	00	11.	06
11.	01	12.	07
11.	02	13.	08
0.	03	14.	09
15.	04	15.	10
11.	05		

3. Associate each column/row of data with one of the user defined keys (A - E).
4. Enter each number in each column/row depressing the associated user defined key after each entry.
5. To move to a new group of 5 columsn/rows, press E'. The new group number will be displayed.
6. To return to the next lower number group, press A'. The next lower group number will be displayed.
7. To total all column/row totals, press D'. The grand total will be displayed.
8. To display current group number, press C'.
9. To multiply all column/row sums by a constant, enter the constant and press B. (To clear all summations, enter 0 and press B.)

EXAMPLE:

Find the sum of each column in the following table. Then multiply each sum by 2 and calculate the resulting grand total.

167	41	161	0	84	93
64	29	44	51	100	101
75	36	63	81	12	13

Enter	Press	Display	Comments
167	A	167	
64	A	231	
75	A	306	1st total
41	B	41	
29	B	70	
36	B	106	2nd total
161	C	161	
44	C	205	
63	C	268	3rd total
0	D	0	
51	D	51	
81	D	132	4th total
84	E	84	
100	E	184	
12	E	196	5th total
	E'	2	Reprogram user defined keys
93	A	93	
101	A	194	
13	A	207	6th total
2	B'	2	multiply all totals
	A'	1	next lower group
	CLR C	536	view 3rd total
	D'	2430	grand total
0	B'	0	clear all summation registers.

TI-59 LISTING

```
000  76 LBL|022  14   D |044  43 RCL|066  43 RCL|088  64 PD*|110  43 RCL|132  70 RAD|154  76 LBL
001  11  A |023  74 SM*|045  00  00|067  00  00|089  02  02|111  04  04|133  43 RCL|155  80 GRD
002  74 SM*|024  09  09|046  22 INV|068  75  - |090  43 RCL|112  32 X:T|134  05  05|156  43 RCL
003  06  06|025  73 RC*|047  44 SUM|069  43 RCL|091  02  02|113  00   0|135  42 STO|157  00  00
004  73 RC*|026  09  09|048  06  06|070  05  05|092  67  EQ|114  42 STO|136  02  02|158  44 SUM
005  06  06|027  91 R/S|049  22 INV|071  95  = |093  90 LST|115  03  03|137  43 RCL|159  06  06
006  91 R/S|028  76 LBL|050  44 SUM|072  55  ÷ |094  01   1|116  76 LBL|138  03  03|160  44 SUM
007  76 LBL|029  15   E|051  07  07|073  43 RCL|095  44 SUM|117  98 ADV|139  91 R/S|161  07  07
008  12  B |030  74 SM*|052  22 INV|074  00  00|096  02  02|118  73 RC*|140  76 LBL|162  44 SUM
009  74 SM*|031  10  10|053  44 SUM|075  95  = |097  61 GTO|119  02  02|141  10 E' |163  08  08
010  07  07|032  73 RC*|054  08  08|076  91 R/S|098  99 PRT|120  44 SUM|142  43 RCL|164  44 SUM
011  73 RC*|033  10  10|055  22 INV|077  76 LBL|099  76 LBL|121  03  03|143  04  04|165  09  09
012  07  07|034  91 R/S|056  44 SUM|078  17 B' |100  90 LST|122  43 RCL|144  32 X:T|166  44 SUM
013  91 R/S|035  76 LBL|057  09  09|079  42 STO|101  43 RCL|123  02  02|145  43 RCL|167  10  10
014  76 LBL|036  16 A' |058  22 INV|080  03  03|102  05  05|124  67  EQ|146  10  10|168  61 GTO
015  13  C |037  43 RCL|059  44 SUM|081  43 RCL|103  42 STO|125  70 RAD|147  22 INV|169  18 C'
016  74 SM*|038  05  05|060  10  10|082  04  04|104  02  02|126  01   1|148  67  EQ|
017  08  08|039  32 X:T|061  76 LBL|083  32 X:T|105  43 RCL|127  44 SUM|149  80 GRD|
018  73 RC*|040  43 RCL|062  18 C' |084  76 LBL|106  03  03|128  02  02|150  43 RCL|
019  08  08|041  06  06|063  43 RCL|085  99 PRT|107  91 R/S|129  61 GTO|151  00  00|
020  91 R/S|042  67  EQ|064  06  06|086  43 RCL|108  76 LBL|130  98 ADV|152  44 SUM|
021  76 LBL|043  18 C' |065  85  + |087  03  03|109  19 D' |131  76 LBL|153  04  04|
```

FROM THE ANALYST'S DESK *(cont'd.)*

The following routine, which solves for the unknown variable in Ohm's law, illustrates the routine (i.e., E=IR, I=E/R, or R=E/I).

```
000   76 LBL | 014   95   = | 028   01  01 | 042   67  EQ
001   11   A | 015   42 STO | 029   55   ÷ | 043   00  00
002   93   . | 016   01  01 | 030   43 RCL | 044   51  51
003   00   0 | 017   92 RTN | 031   03  03 | 045   43 RCL
004   29 CP  | 018   76 LBL | 032   95   = | 046   01  01
005   22 INV | 019   12   B | 033   42 STO | 047   55   ÷
006   67  EQ | 020   93   . | 034   02  02 | 048   43 RCL
007   00  00 | 021   00   0 | 035   92 RTN | 049   02  02
008   15  15 | 022   29 CP  | 036   76 LBL | 050   95   =
009   43 RCL | 023   22 INV | 037   13   C | 051   42 STO
010   02  02 | 024   67  EQ | 038   93   . | 052   03  03
011   65   × | 025   00  00 | 039   00   0 | 053   92 RTN
012   43 RCL | 026   33  33 | 040   29 CP  |
013   03  03 | 027   43 RCL | 041   22 INV |
```

Each label is followed by .0 which has no effect on keyed in entries by generates a zero if no number is keyed in prior to pressing the input label. A test for zero follows which determines whether storage for known variables or calculation of the unknown variable should take place. For example:

Enter	Press	Display	Comments
6	A	6	E
2	B	2	I
	C	3	R

Although the input variables must be non-zero keyboard entries (i.e., not the result of previous calculations), this scheme can be useful in your own programming.

THE BIG "D"

No this is not about the world famous Dallas Cowboys! This is about the other big D. The big D addendum (July 1979), which you should have received by now. We would like to take this opportunity to tell you about a few of the programs which struck our fancy; as our space is limited, we can only mention a few:

- "Small Business Accounting" (PPX-59,#108005) stores up to 38 account balances given the daily transactions (i.e., debits and credits). Available at any time are a trial balance, profit and loss statement, and balance sheet.
- Constant adjustments to a large amount of numbers can be performed quickly and easily using "Constant Arithmetic" (PPX-59 #398118). The following operations can be performed using a constant c and variable x: c+x, c-x, x-c, c*x, c/x, x/c, c^x and x^c. Once the constant is entered, an entire set of numbers can be adjusted by simply entering the variables (an example of this would be the computation of sales tax for a list of prices).
- "Density of Seawater" (PPX-59 #478003). Now here's a program no oceanographer should be without. Given the depth, temperature and salinity, this program computes the density of seawater in either metric or English units. This should be a big time saver.
- The trouble with most Decimal/Binary conversion programs is that the size of the binary number is limited by the display of the calculator. "Binary/Decimal Conversion" (PPX-59 #638011) ignores this limitation. By allowing the user to input or recall the binary number in segments, this program allows the conversion of binary numbers that would normally overflow the display. The maximum data entry is a 33 bit binary or the decimal number 527287.
- PPX Analysts loved checking out the challenging game of "Peg Jump" (PPX-59 #918110). This game provides hours of entertainment on the TI-59/PC-100A. The object of this game is to jump over a peg with an adjacent peg, landing in a hole. You start the game with 16 holes, 15 of which are filled with pegs. The game is over when the one peg is left in the starting hole.

The PPX $\boxed{\text{Exc}}$hange is published every other month and is the only newsletter published by Texas Instruments for TI-59 owners. You are invited to submit items you feel are of general interest to other TI-59 users. Inputs should be limited to 3 double-spaced typed pages. Please forward your newsletter inputs and any questions to:

TEXAS INSTRUMENTS PPX
P. O. Box 53
Lubbock, TX 79408
Attn: PPX Exchange Editor

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

BULK RATE
U.S. POSTAGE
PAID
Permit No. 1
Lubbock, Texas