PPX
EXCHANGE

Vol. 2 Number 4 Copyright 1978

July 1978

PPX is doing it again! Until September 30, while supplies last, SR-52 Games Libraries will be available for the unbelievable low price of $14.95. (Suggested retail price is $29.95.) Limit is 1 Library per member. And that's not all . . . 3-roll packages of PC-100(A) thermal paper will be sold for $7.20 each. This is a $3 savings compared to the suggested retail price of $10.20 per 3-roll package. Limit is 3 packages per member. To order, simply use a blank order form and enclose your check or money order. All thermal paper orders received after August 1 (before knowledge of the sale) will be filled at the sale price and a $3 credit in the form of 1 Complimentary Program will be given. Orders not remitting proper taxes and handling charges will be returned unfilled to the sender.

PPX POTPOURRI

1. The month of July brings the release of two software catalogs — the PPX-59 Software Catalog and the PPX-52 Software Addendum. The PPX-59 Catalog contains approximately 350 new abstracts of programs, bringing the total program offering of PPX-59 to over 600 programs. After two years, PPX-52 boasts of over 1700 programs, with approximately 175 new programs featured in the July Addendum.

2. The PPX-52 July Addendum marks the close of SR-52 program submissions. Orders for programs featured in past Catalogs and Addendums will continue to be filled.

3. **If you are a PPX-52 member whose membership number is greater than 108778, you need not renew your membership.** You are already entitled to newsletters through the end of 1978 and continued ordering privileges.

4. On the same note . . . PPX-59 members need not worry about membership renewal until an announcement is made in a future issue of the PPX Exchange.

5. Return to sender . . . Lack of **current addresses** continues to plague PPX service. Please let PPX know your new address as soon as possible to insure the best possible service.

6. Speaking about the mail . . . It has come to our attention that PPX-59 order forms are being sent back to some of our members instead of being sent to PPX. To avoid this situation, please put the word FROM next to the return address label on your order form.

7. New members . . . Your "3 Introductory Programs" order form is to be used for **individual** programs. It cannot be used as credit towards a Pakette (or any other supplies). All such orders will be returned unfilled to the sender.

8. The question has been raised concerning the purpose of the Professional Category Table. The table was set up as a guideline for possible areas of program development and does not imply that there are programs in every category. With the addition of the PPX-59 July Catalog, PPX-59 offers programs in 73 categories. PPX Exchange's newest column, "The Programming Corner" (beginning this issue), is presented in an effort to encourage program submissions in all 99 categories.

TI-59, THE FIRST CALCULATOR WITH A "TRUE" MEMORY

Donald R. Lambert

The concept of memory in programmable calculators can be likened to a box. One can place a number in it (store), look at its contents (recall), increase or decrease its contents (register arithmetic), and so on. Some calculators have the ability to hold the contents of the box intact when turned off (continuous memory), but no calculator could actually remember what had been in the box (memory) after the contents were changed.

That is, no calculator had "true" memory until the TI-59 was introduced in 1977! Under certain circumstances the TI-59 will remember what was in its program memory for a limited time after it has been replaced (overwritten) by a new program.

The discovery of this remarkable feature of the TI-59 came purely by accident. I was working on a long stock market data-processing application for one of my clients that required a combination of steps and memories beyond the capacity of the TI-59 in any partition, so I decided to split the program into several parts. Since the end user of the program was not familiar with the calculator, it seemed best to have all but the first part of the program loaded into the machine under program control. Therefore, I programmed the following sequence into bank #2 of the first part of the program: *Lbl, SBR, 1, INV, *Write, RST. Execution of this sequence caused the overwriting of bank #1 and the resetting of the program counter to location 000. During the evolution of my program, the above sequence of steps were accidently shifted into bank #1 of the program. The program still ran! Although the RST instruction was overwritten by an entirely different keystroke, it was executed.

I began checking this capability out. My experiments with the TI-59 indicate that program steps are called from main program memory in blocks of eight steps and stored in a "buffer" (temporary) memory pending execution. After a block of eight steps is executed, the next block is called, and so on.

In my particular program, INV *Write shared the same block as RST. Although RST had been replaced in program memory by the execution of INV *Write, it was still in the "buffer" awaiting execution.

Further work with the TI-59 showed that when a bank overwrites itself, up to seven steps (the first step being Write) may remain in the buffer to be executed. These steps will be executed provided that a transfer (conditional or unconditional) to another part of the program is not effected. When a subroutine is called, the return will be to the next step in the new program not to the next step stored in the buffer.

To illustrate the above, try the following: Record a completely blank bank #1 on a card, then press GTO 116 LRN *Lbl A 1 INV *Write 1 2 3 INV SBR 4 5 INV SBR LRN. Now Press A, insert the blank card (the INV *Write will execute) and you'll see "123" in the display. Press R/S, you'll see "45" in the display. Now press LRN and backstep through program memory. It is completely blank. That was "true" memory in operation!

PPX-59 PROGRAMMING CORNER

This column is devoted to PPX-59 programming suggestions. If you have a program(s) that you would to see made available through PPX-59, send your suggestion(s) to PPX. In this way, members who enjoy programming are made aware of your program 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:

- Personal Income Tax programs.
- A program to calculate yield of a wrap-around mortgage.
- Programs dealing with Insurance and Estate Planning and in particular a program that will compute the true cost of an insurance policy.
- Civil Engineering programs dealing with hydraulics and sanitary engineering subjects.
- Seismology programs and in particular a program that will do an interpretation of a sonic log.

Editor's Note: In the last issue of the PPX Exchange, we requested that our members notify us about their program needs. Program requests were sent, however many did not offer enough detail. In order for your needs to be filled, other members need to know what specific problem you would like solved (not what areas you would like to see developed).

SOLUTION TO GONE

The solution to "GONE" (PPX Exchange, May 1978) is based on the sequence: RCL 00 + STO 60 1 = (negative number stored in register 00). Execution of this sequence will reset the calculator (i.e., wipe it out). If you check through the keycode in "GONE", you will not be able to find this sequence. Mr. Pangburn has cleverly programmed the calculator to modify the code so that the sequence is eventually placed into steps 111 through 119 and executed.

THE SR-52 AND SELF MODIFICATION OF CODE

Lawrence R. Pangburn

Editor's Note: The following article is presented to explain the process of self-modifying code used in the program "GONE" (published in PPX Exchange, May 1978). The principle behind self modification can be applied to the TI-59 as well. Page VII-1 of the Personal Programming Manual shows the relationship of registers to programming steps. However, in order to modify TI-59 code in a register, the program must repartition itself to access that register.

One of the interesting features of the SR-52 is that you can design a program that will modify itself during execution. This is possible because the program code is in accessible registers (which can be used exactly like data registers). Eight program steps are located in each of the program registers numbered 70 through 97. The first eight steps are in register 70, the second eight steps are in register 71, and so on.

Before we proceed, there are a few points that must be clarified. The **display register** contains 2 signs, a 2-digit exponent, and either a 12 or 13-digit mantissa (depending upon which model of the SR-52 you have). To check if your calculator's display register has a 12 or 13 digit mantissa, key in the following sequence (out of learn mode): 3, *1/x,- 3, *1/x, =. If the displayed answer is zero, your calculator has a 12-digit mantissa.

Although the display register has a 12 or 13-digit mantissa, only 10 of the digits can appear on the calculator's display. For this reason, to view the full content of your display register it is necessary to move the contents to your calculator's display in two moves.

To proceed . . . in learn mode, key in the program code shown in Figure 1. Now observe the program code stored in register 70 by pressing RCL 70. The number in your display represents the program code in the last four steps of register 70. See Figure 2. Next, we will look at the program code in the first four steps of register 70. In order to see these steps we must use a trick to bring the remaining contents of the display register into the 10-digit calculator display. The method is:

CLR 1 + STO 60 RCL 70 =

The number in the display represents the program code in the first four steps of register 70 (See Figure 2). Two things must be mentioned about this representation.

- The second code digit of step number 1 depends upon the combination of the arithmetic signs of the mantissa and the exponent (See Figure 3).
- For 12-digit calculators, there will always be a zero in the left digit position of step number 2 (Internally the calculator knows what to execute).

Before we demonstrate arithmetic modification of code, we will set up the data registers for LBL A (in Figure 1). This is accomplished by pressing 50 STO 01 20 STO 02. Now, press A and observe that the result happens to be 120. We are ready to modify the code in LBL A. First, perform the following out of LRN mode: 1 EE 14 +/- SUM 70 INV EE. This operation changes code 85 to 75 at program location 007, (i. e. it changes the "+" to a "-" by arithmetically modifying the number (code) in register 70). The result may be verified by going to LRN mode and observing the program code in locations 000 through 007. The effect may be directly

REGISTER	STEP	CODE	KEY	LOCATION
70	1	46	*LBL	000
	2	11	A	001
	3	02	2	002
	4	65	x	003
	5	43	RCL	004
	6	00	0	005
	7	01	1	006
	8	85	+	007
71	1	43	RCL	008
	2	00	0	009
	3	02	2	010
	4	95	=	011
	5	56	*rtn	012
	6	46	*LBL	013
	7	15	E	014
	8	25	CLR	015
72	1	94	+/-	016
	2	46	*LBL	017
	3	14	D	018
	4	01	1	019
	5	52	EE	020
	6	01	1	021
	7	04	4	022
	8	94	+/-	023
73	1	44	SUM	024
	2	07	7	025
	3	00	0	026
	4	22	INV	027
	5	52	EE	028
	6	56	*rtn	029
	7	00	0	030
	8	81	HLT	031

Figure 1. SAMPLE CODE

–	8.5	01	00	43	65 – 14	CODE
	8	7	6	5		STEP

Last 4 steps in reg 70
(all calculators)

CODE	– 10043	65	02	1 – 1	4
STEP		4	3	2	1

First 4 steps in reg 70
(13-digit calculators)

CODE	– 1 0 0 4 3	6 5	0 2	0 – 1	4
STEP		4	3	2	1

First 4 steps in reg 70
(12-digit calculators)

Figure 2. PROGRAM CODE IN DISPLAY REGISTER

If Mantissa &	If Exponent	Then Digit
+	+	0
–	+	2
+	–	4
–	–	6

Figure 3. REPRESENTATION OF SECOND DIGIT OF FIRST STEP

verified by pressing A, which produces the new result, 80. The code that you originally set up for Figure 1 has this arithmetic operation under LBL D, and the inverse operation under LBL E. If you press D you will find that in LBL A, the "-" is changed to "x" and pressing A now results in 2000 (as opposed to the 120 obtained earlier). If you press E the "x" returns to "-". Thus, we have routines D and E which modify the arithmetic operation in LBL A.

We could also generate a routine that would change the numbers in LBL D (program locations 021 and 022) such that D and E would then modify the other arithmetic operation in LBL A (program location 003). Try it! In fact you can easily modify any code in steps 2 through 8 in register 70 by using any legal arithmetic operations.

Now that we have a basic understanding of changing the code, we should note the limitations which exist. They are:

(1) The first digit in step number 8 **should not be zero.** If it is a zero, any manipulation of code produces major changes in code.

(2) During arithmetic operations you should not push the code from a program register into the pending operations stack. If you do, the 13th digit, the first digit of step number 2, will become zero. (12-digit calculators already have a zero in this position.)

(3) The second digit in step number 1 should be 0, 2, 4, or 6 if it is to be unaffected by arithmetic operations. If it is 1, 3, 5, or 7, attempted operations will change it to the next lower digit. If it is 8 or 9, it will **usually become zero.** In some cases when it is 9 you will encounter an overflow, which means that you cannot manipulate the code.

These above limitations may sometimes by used to your advantage. For example, in (3) above, if you wanted to change the first step in register 71 from RCL to STO you need only RCL 71 STO 71 to accomplish this. This operation changes the 3 in code 43 to 2 because the RCL 71 forces the code into the display register as a number whereby that digit is represented by the arithmetic signs of the mantissa and the exponent.

As another example, suppose you wanted to change SUM to LBL in the first step of register 73. This is accomplished by 1 +/- *PROD 73 which changes the sign of the mantissa thereby changing the second digit from 4 to 6. See figure 3. (Note that in Figure 1, program location 031, step number 8, was made a HLT to avoid the zero digit limitation discussed in (1) above.)

To demonstrate (2) above, press RCL 72 + 0 = STO 72. This pushes the arithmetic representation of the code from register 72 into the pending operation stack (i.e. RCL 72 +). If you go to LRN mode and observe program location 017, you will find that it is now 06 instead of 46. The 4 became a zero.

As you may have guessed by now, you can generate many codes completely by arithmetically combining the appropriate digits and storing the result in a program register. This is what enabled the program "GONE" to wipe itself out without a trace of evidence.

91 − [ ] = 38   7 × 25 =   50   64   6 + 53 =   65   90 − 30 = 60

ARITHMETIC TUTOR

48 + 29 = [ ]

21 − 7 =   30   9 − 4 =   29   31   28 + 75 = [ ]

The following two programs are presented as educational tools for children. Through the use of these programs, children will be able to better their math skills and at the same time add enjoyment to the learning experience. A joint effort between the parent and child is necessary only in the beginning. Once the child has been shown how the program works, he can operate the calculator on his own. A pad of paper and pencil will be necessary as the child must write the problem down to solve it.

PPX wishes to thank the author of the SR-52 program, Dennis White, and the author of the TI-59 program, Chris Wegehenkel, for their excellent programs.

DESCRIPTION (SR-52): This program generates a user-determined number of different problems for a pupil to solve. This problem is presented in the format XX.YY (where XX is the first number and YY is the second number to be operated on). The user specifies the maximum value that either XX or YY can assume. If an answer is incorrectly given, the problem will be repeated. When all the problems have been answered, the SR-52 displays the percentage of correct answers. All answers are positive integers.

USER INSTRUCTIONS (SR-52)

1. Enter program.
2. Enter a seed number, 0 to 1, press C'.
3. Enter maximum value of numbers to be generated (0<max<100), press B'. The difficulty of problems generated is dependent upon the maximum value assigned to the numbers. If a default value of 12 is desired then skip this step.
4. Enter number of problems desired. Choose the operation to be performed. Press B for addition, C for subtraction, D for multiplication, and E for division. Two numbers will be generated in the format XX.YY (where XX is the first number and YY is the second number to be operated on).
5. Enter answer and press A.
6. If answer is correct, a new problem is generated. Repeat Step 5.
7. If answer is incorrect, the problem is again displayed, preceded by a minus sign. Repeat Step 5.
8. After the specified number of problems have been correctly answered, the percentage of correct answers is displayed (incorrectly answered problems are counted in the base upon which this percentage is figured).
9. Repeat Step 3 or 4.

NOTE: If you enter an answer into the display and wish to again view the problem (assuming A has not been pressed yet), press A'.

EXAMPLE (SR-52): Initialize using a seed of .625. Set the maximum number size to 50. Then generate 5 addition problems and 3 multiplication problems.

Enter	Press	Display	Comments
.625	C'	12.	Enter seed
50	B'	50.	Enter maximum value of numbers
5	B	44.11	Enter # problems and operation
55	A	25.25	Correct, next problem
225	A	-25.25	Incorrect, try again
50	A	42.21	Correct, next problem
63	A	32.19	Correct, next problem
41	A	-32.19	Incorrect, try again
51	A	30.38	Correct, last problem
68	A	71.4	Percent correct
3	D	34.31	Enter # problems and operation
954	A	-34.31	Incorrect, try again
1054	A	28.32	Correct, next problem
896	A	24.10	Correct, next problem
240	A	75.0	Percent correct

SR-52 LISTING

```
000  44 SUM|024  02   2|048  95   =|072  44 SUM|096  65   ×|120  01   1|144  42 STO|168  88  2'|192  75   -
001  00   0|025  46 LBL|049  42 STO|073  00   0|097  93   .|121  44 SUM|145  00   0|169  46 LBL|193  43 RCL
002  02   2|026  19  D'|050  09   9|074  00   0|098  00   0|122  00   0|146  00   0|170  17  B'|194  00   0
003  56 RTN|027  25 CLR|051  08   8|075  46 LBL|099  01   1|123  05   5|147  04   4|171  42 STO|195  05   5
004  44 SUM|028  04   4|052  65   ×|076  89  3'|100  95   =|124  44 SUM|148  41 GTO|172  09   9|196  95   =
005  00   0|029  85   +|053  43 RCL|077  22 INV|101  44 SUM|125  00   0|149  88  2'|173  09   9|197  55   ÷
006  01   1|030  43 RCL|054  09   9|078  58 DSZ|102  00   0|126  04   4|150  46 LBL|174  56 RTN|198  43 RCL
007  56 RTN|031  09   9|055  09   9|079  77  4'|103  01   1|127  43 RCL|151  14   D|175  46 LBL|199  00   0
008  49 PRD|032  08   8|056  75   -|080  01   1|104  46 LBL|128  00   0|152  47 CMS|176  18  C'|200  04   4
009  00   0|033  95   =|057  93   .|081  44 SUM|105  16  A'|129  01   1|153  42 STO|177  47 CMS|201  65   ×
010  02   2|034  45  YX|058  05   5|082  00   0|106  25 CLR|130  94 +/-|154  00   0|178  42 STO|202  01   1
011  56 RTN|035  59   π|059  95   =|083  04   4|107  43 RCL|131  81 HLT|155  00   0|179  09   9|203  00   0
012  90 IFZ|036  95   =|060  52  EE|084  19  D'|108  00   0|132  46 LBL|156  08   8|180  08   8|204  00   0
013  00   0|037  75   -|061  22 INV|085  42 STO|109  01   1|133  12   B|157  41 GTO|181  01   1|205  95   =
014  02   2|038  53   (|062  52  EE|086  00   0|110  56 RTN|134  47 CMS|158  88  2'|182  02   2|206  57 FIX
015  00   0|039  57 FIX|063  57 FIX|087  01   1|111  46 LBL|135  42 STO|159  46 LBL|183  42 STO|207  01   1
016  49 PRD|040  00   0|064  02   2|088  42 STO|112  11   A|136  00   0|160  15   E|184  09   9|208  81 HLT
017  00   0|041  75   -|065  56 RTN|089  00   0|113  75   -|137  00   0|161  47 CMS|185  09   9|
018  01   1|042  93   .|066  46 LBL|090  02   2|114  43 RCL|138  25 CLR|162  42 STO|186  56 RTN|
019  56 RTN|043  05   5|067  88  2'|091  19  D'|115  00   0|139  41 GTO|163  00   0|187  46 LBL|
020  19  D'|044  54   )|068  42 STO|092  36 IND|116  02   2|140  88  2'|164  00   0|188  77  4'|
021  41 GTO|045  52  EE|069  00   0|093  51 SBR|117  95   =|141  46 LBL|165  01   1|189  43 RCL|
022  00   0|046  22 INV|070  03   3|094  00   0|118  90 IFZ|142  13   C|166  02   2|190  00   0|
023  01   1|047  52  EE|071  01   1|095  03   3|119  89  3'|143  47 CMS|167  41 GTO|191  04   4|
```

Listing produced with a TI-59/PC-100A using the program "SR-52 Program Listing" (PPX-59 #908010).

DESCRIPTION (TI-59): The TI-59 generates 10 math problems in the format XX.YY (where XX is the first number and YY is the second number to be operated on). The skill level and operation to be performed are user specified. User is given three chances to answer each problem correctly. If three incorrect answers are given, the TI-59 displays the correct answer. All answers are positive integers.

EXAMPLE (TI-59): Generate addition problems on skill level 3. Initialize using seed 5555.

Enter	Press	Display	Comments
	C'	3.	Skill level 3
	A	1.	Addition
5555	E'	36.30	Enter seed, first problem
66	R/S	38.19	Correct, next problem
57	R/S	9.11	Correct, next problem
20	R/S	38.24	Correct, next problem
62	R/S	3.15	Correct, next problem
18	R/S	50.19	Correct, next problem
69	R/S	25.14	Correct, next problem
39	R/S	17.26	Correct, next problem
33	R/S	(.2222222222)	Incorrect (flashing)
		17.26	Try again
43	R/S	34.46	Correct, next problem
70	R/S	(.2222222222)	Incorrect (flashing)
		34.46	Try again
80	R/S	24.30	Correct, next problem
54	R/S	10.	All 10 correct

USER INSTRUCTIONS (TI-59)

1. Enter program (partition 479.59).
2. Choose skill level. This skill level determines the upper and lower limit of the numbers to be generated. Press A' (skill level #1) for numbers between 0 and 13; B' (skill level #2) for numbers between 0 and 26; C' (skill level #3) for numbers between 0 and 51; or D' (skill level #4) for numbers between 0 and 100.
3. Choose the operation to be performed. Press A for addition, B for subtraction, C for division, or D for multiplication.
4. Enter a seed number, 0 to 199017, press E'. Two numbers will be displayed in the format XX. YY (where XX is the first number and YY is the second number to be operated on).
5. Enter answer and press R/S.
6. If answer is correct, new pair of numbers is generated. Repeat Step 5.
7. If answer is incorrect, one of the following occurs: .2222222222 flashes, followed by the problem re-displayed, indicating 2 remaining tries left; .1111111111 flashes, followed by the problem re-displayed, indicating 1 remaining try left; correct answer flashes, indicating no remaining tries left (press R/S for new problem). Repeat Step 5.
8. After the 10th problem, the number of problems answered correctly within three tries is displayed. Press E to begin new set of problems.

NOTE: Skill level or operation may be changed at any time, however a new set of problems must be generated after the change. A new seed is not necessary.

TI-59 LISTING

```
000 86 STF  039 08 08   078 76 LBL  117 76 LBL  156 06 06   195 10 10   234 02 02   273 00 00   312 09 9
001 40 IND  040 65 ×    079 60 DEG  118 45 Yˣ   157 32 X:T  196 32 X:T  235 61 GTO  274 02 2    313 09 9
002 07 07   041 05 5    080 92 RTN  119 43 RCL  158 43 RCL  197 43 RCL  236 25 CLR  275 06 6    314 00 0
003 22 INV  042 22 INV  081 76 LBL  120 05 05   159 05 05   198 11 11   237 76 LBL  276 42 STO  315 01 1
004 58 FIX  043 28 LOG  082 15 E    121 55 ÷    160 87 IFF  199 22 INV  238 43 RCL  277 01 01   316 07 7
005 43 RCL  044 95 =    083 58 FIX  122 43 RCL  161 01 01   200 67 EQ   239 03 3    278 02 2    317 42 STO
006 07 07   045 59 INT  084 02 02   123 06 06   162 23 LNX  201 33 X²   240 42 STO  279 92 RTN  318 08 08
007 92 RTN  046 55 ÷    085 00 0    124 95 =    163 87 IFF  202 03 3    241 03 03   280 76 LBL  319 61 GTO
008 76 LBL  047 05 5    086 42 STO  125 42 STO  164 02 02   203 42 STO  242 00 0    281 18 C'   320 15 E
009 88 DMS  048 22 INV  087 04 04   126 10 10   165 24 CE   204 03 03   243 35 1/X  282 22 INV  321 76 LBL
010 02 2    049 28 LOG  088 01 1    127 22 INV  166 87 IFF  205 69 OP   244 22 INV  283 58 FIX  322 11 A
011 04 4    050 65 ×    089 00 0    128 59 INT  167 03 03   206 24 24   245 58 FIX  284 00 0    323 01 1
012 02 2    051 53 (    090 42 STO  129 29 CP   168 32 X:T  207 76 LBL  246 43 RCL  285 42 STO  324 42 STO
013 09 9    052 43 RCL  091 02 02   130 67 EQ   169 65 ×    208 70 RAD  247 10 10   286 00 00   325 07 07
014 08 8    053 01 01   092 03 3    131 25 CLR  170 32 X:T  209 97 DSZ  248 91 R/S  287 05 5    326 81 RST
015 65 ×    054 75 -    093 42 STO  132 69 OP   171 95 =    210 02 02   249 25 CLR  288 01 1    327 76 LBL
016 43 RCL  055 43 RCL  094 03 03   133 36 36   172 61 GTO  211 34 √X   250 58 FIX  289 42 STO  328 12 B
017 09 09   056 00 00   095 76 LBL  134 61 GTO  173 35 1/X  212 22 INV  251 02 02   290 01 01   329 02 2
018 85 +    057 54 )    096 34 √X   135 45 Yˣ   174 76 LBL  213 58 FIX  252 61 GTO  291 03 3    330 42 STO
019 09 9    058 85 +    097 71 SBR  136 76 LBL  175 23 LNX  214 43 RCL  253 70 RAD  292 92 RTN  331 07 07
020 09 9    059 43 RCL  098 88 DMS  137 61 GTO  176 85 +    215 04 04   254 76 LBL  293 76 LBL  332 81 RST
021 09 9    060 00 00   099 42 STO  138 71 SBR  177 32 X:T  216 92 RTN  255 16 A'   294 19 D'   333 76 LBL
022 09 9    061 95 =    100 05 05   139 52 EE   178 95 =    217 76 LBL  256 22 INV  295 22 INV  334 13 C
023 01 1    062 59 INT  101 71 SBR  140 76 LBL  179 61 GTO  218 33 X²   257 58 FIX  296 58 FIX  335 03 3
024 95 =    063 92 RTN  102 88 DMS  141 25 CLR  180 35 1/X  219 22 INV  258 00 0    297 00 0    336 42 STO
025 55 ÷    064 76 LBL  103 42 STO  142 43 RCL  181 76 LBL  220 97 DSZ  259 42 STO  298 42 STO  337 07 07
026 43 RCL  065 52 EE   104 06 06   143 06 06   182 24 CE   221 03 03   260 00 00   299 00 00   338 81 RST
027 08 08   066 43 RCL  105 87 IFF  144 55 ÷    183 75 -    222 43 RCL  261 01 1    300 01 1    339 76 LBL
028 95 =    067 05 05   106 02 02   145 01 1    184 32 X:T  223 22 INV  262 03 3    301 00 0    340 14 D
029 22 INV  068 32 X:T  107 61 GTO  146 00 0    185 95 =    224 58 FIX  263 42 STO  302 00 0    341 04 4
030 59 INT  069 43 RCL  108 87 IFF  147 00 0    186 61 GTO  225 09 9    264 01 01   303 42 STO  342 42 STO
031 65 ×    070 06 06   109 03 03   148 85 +    187 35 1/X  226 35 1/X  265 01 1    304 01 01   343 07 07
032 43 RCL  071 22 INV  110 55 ÷    149 43 RCL  188 76 LBL  227 65 ×    266 92 RTN  305 04 4    344 81 RST
033 08 08   072 77 GE   111 61 GTO  150 05 05   189 32 X:T  228 43 RCL  267 76 LBL  306 92 RTN
034 95 =    073 60 DEG  112 25 CLR  151 95 =    190 43 RCL  229 03 03   268 17 B'   307 76 LBL
035 42 STO  074 48 EXC  113 76 LBL  152 91 R/S  191 10 10   230 95 =    269 22 INV  308 10 E'
036 09 09   075 05 05   114 55 ÷    153 42 STO  192 76 LBL  231 66 PAU  270 58 FIX  309 42 STO
037 55 ÷    076 42 STO  115 71 SBR  154 11 11   193 35 1/X  232 66 PAU  271 00 0    310 09 09
038 43 RCL  077 06 06   116 52 EE   155 43 RCL  194 42 STO  233 58 FIX  272 42 STO  311 01 1
```

TI ANNOUNCEMENT

• Texas Instruments' has released a new TI-59 Library, Business Decisions, designed to help today's businessman by providing aids to more accurate market forecasting, improved inventory utilization, and estimation of financing requirements. Programs in the Library include: Long Term Financing Requirements; Short Term Financing Requirements; Planning and Budgeting; Breakeven Analysis; Investment Evaluation; Economic Reordering and Production Runs; Reorder Timing; Facility Scheduling; Assembly Line Balancing; Demand Forecasting; and Facility Capacity. The Business Decisions Library may be ordered through PPX for $35.00 plus tax and handling.

FROM THE ANALYST'S DESK

• Two PPX-59 programs have been rewritten due to errors in previous versions. If you have purchased "Tic Tac Toe" (PPX-59 #918016) or "Baseball" (PPX-59 #918012), please send your copy in for a replacement.

• If you have purchased "Biorhythms" (PPX-59 #918908), please make the following changes to your program listing. Steps 262 through 269 and step 454 should read:

262	71	SBR	267	97	DSZ
263	43	RCL	268	09	09
264	68	NOP	269	02	02
265	68	NOP	454	49	49
266	22	INV			

• Our latest, July 1978, PPX-59 Catalog features many inventive and versatile programs. A few of these programs caught the fancy of the analysts but are by no means the only programs worthy of note.

"Perspective, 3D Option- Illustrator's Aid" (PPX-59 #698004) by Morton P. Matthew allows the user to produce perspective views of up to 25 x, y, z points by printing the x, z points on the visual plane. Multiple views and a variable degree of depth effect can be accomplished without re-entry of the x, y, z coordinates. PC-100(A) required.

"Critical Path Method (CPM)" (PPX-59 #098006) by Michael Shanok is a network schedule planning program applicable to any type of project planning. Events which follow the critical path are those which have zero float (number of extra days) and must be completed before another activity can begin. This program is used to assign start and finish dates, days to complete and float for a network schedule. Mod 1 and PC-100(A) required.

"Poem Machine" (PPX-59 #918030) by William G. Bryson was voted the funniest program by PPX Analysts. It will crack a smile on the face of even the most serious of programmers. The "Poems" printed are grammatically correct random sentences using user-supplied nouns, verbs, pronouns, and adjectives. A great program to have running during a party.

• PPX was notified of three limitations to the Hierarchy operation (PPX Exchange, May 1978) by Mr. Lawrence Pangburn of Hauppauge, New York. The limitations are:

1. If an arithmetic operation (Sum, Prd, Inv Sum, Inv Prd) is performed into a pending operations register using a number less than 1 (one), the number must be in EE format in order to obtain correct results.

2. If any operation (Sto,Sum,Prd,Inv Sum, Inv Prd) is performed into a pending operations register that is holding a subtraction (e.g., it was established via 25 -) the operation will be changed to addition. A register holding addition is unchanged (it is still addition) by operations into the register.

3. If any operation (Sto,Sum,Prd,Inv Sum,Inv Prd) is performed into a pending operations register that is holding a division (e.g., it was established via 25 ÷) the operation will be changed to multiplication. A register holding multiplication will perform correctly after operations into the register.

• Mohamed Sherif Tawfik shares the following: A disadvantage of Program 18 in the Applied Statistics Library is the limitation on the number of sets of data that can be entered. Unlimited data entry is possible with the following modifications. Store these modifications on a magnetic card. Instead of calling Program 05 to enter data, enter the magnetic card then proceed with data entry according to the instructions in the manual.

Modification of Multiple Regression Program

```
000  76 LBL|017  03  03|034  54   )|051  26  26
001  10 E' |018  91 R/S|035  44 SUM|052  54   )
002  47 CMS|019  76 LBL|036  06  06|053  44 SUM
003  91 R/S|020  12   B|037  43 RCL|054  11  11
004  76 LBL|021  42 STO|038  03  03|055  53   (
005  11   A|022  24  24|039  91 R/S|056  43 RCL
006  42 STO|023  44 SUM|040  76 LBL|057  24  24
007  25  25|024  01  01|041  13   C|058  65   ×
008  44 SUM|025  33 X² |042  42 STO|059  43 RCL
009  04  04|026  44 SUM|043  26  26|060  26  26
010  33 X² |027  02  02|044  44 SUM|061  54   )
011  44 SUM|028  53   (|045  07  07|062  44 SUM
012  05  05|029  43 RCL|046  53   (|063  12  12
013  01   1|030  24  24|047  43 RCL|064  43 RCL
014  44 SUM|031  65   ×|048  25  25|065  03  03
015  03  03|032  43 RCL|049  65   ×|066  91 R/S
016  43 RCL|033  25  25|050  43 RCL|
```

(LIMITATION: R^2 cannot be calculated).

Modification of Nonlinear Regression Program

```
000  76 LBL|018  42 STO|036  25  25|054  26  26
001  10 E' |019  25  25|037  33 X² |055  54   )
002  47 CMS|020  44 SUM|038  44 SUM|056  44 SUM
003  91 R/S|021  05  05|039  02  02|057  11  11
004  76 LBL|022  43 RCL|040  43 RCL|058  53   (
005  11   A|023  05  05|041  03  03|059  43 RCL
006  42 STO|024  42 STO|042  91 R/S|060  25  25
007  24  24|025  01  01|043  76 LBL|061  65   ×
008  44 SUM|026  53   (|044  13   C|062  43 RCL
009  04  04|027  43 RCL|045  42 STO|063  26  26
010  01   1|028  24  24|046  26  26|064  54   )
011  44 SUM|029  65   ×|047  44 SUM|065  44 SUM
012  03  03|030  43 RCL|048  07  07|066  12  12
013  43 RCL|031  25  25|049  53   (|067  43 RCL
014  03  03|032  54   )|050  43 RCL|068  03  03
015  91 R/S|033  44 SUM|051  24  24|069  91 R/S
016  76 LBL|034  06  06|052  65   ×|
017  12   B|035  43 RCL|053  43 RCL|
```

(LIMITATION: R^2 cannot be calculated).

The PPX Exchange is published every other month and is the only newsletter published by Texas Instruments for SR-52 and TI-59 owners. You are invited to submit items you feel are of general interest to other SR-52 or 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