PPX EXCHANGE

Vol. 3 Number 2 Copyright 1979 March 1979

Attention PPX-59 Members — PPX is now offering The Sourcebook for Programmable Calculators at a sale price of $9.95 during the months of April and May. This book usually sells for $12.95. The Sourcebook provides excellent examples of problem solutions, while also serving as a TI-59 learning aid. Topics, 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 obtain your copy simply 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 your state's applicable sales tax. Act now as this sale ends May 31, 1979.

PPX POTPOURRI

1. **The Phantom Strikes Again —** PPX has been receiving mail that does not include **the sender's name, address, and PPX-59 membership number.** If you have not received a reply to your correspondence, it's probably because we don't know who or where to write. Please be sure to include all of this information on all correspondence, orders, programs, and program memos. If your membership is in your company's name, please be sure to note this. When opening our large volumes of mail, envelopes are often separated from letters. Therefore, if you do not include an inside return address, there is a chance your letter will not be answered.

2. **All Sales Are Final —** All submitted programs are verified in accordance with the Member's Guide and program abstracts are edited for completeness before appearing in PPX-59 Catalogs and Addendums. As it is impossible to include every aspect of a program in the abstract, PPX **cannot guarantee** that each program ordered will meet **all** of your requirements and include **all** of the enhancements you would like to see. For these reasons and due to the nature of the materials involved (i.e., printed material), PPX cannot honor requests for replacements or refunds. A PPX Program Memo is included with each filled order for constructive criticism by the program user. These comments are passed to the author as described on page 2 of the Member's Guide.

3. On the subject of Program Memos — PPX often receives program memos from users, regarding a program's workability. Much of the time, the problem lies with the user and not the program. Ask yourself the following questions before you send a Program Memo:

(a) Is my calculator properly partitioned? The partition is stated on the User Instructions page. Further information regarding partitioning can be found in your Personal Programming Manual, pp. V-42 and VII-1.

(b) If a library module is required, is it inserted? The Submission Abstract and User Instructions pages note if a module is required.

(c) Are all alphanumeric codes and required constants stored in their appropriate registers? These requirements are usually noted in the User Instructions, Program Description, or Listing pages.

(d) Do all of my keycodes match those of the author's program listing? If you have trouble entering keycodes which have been produced on a PC-100A listing, please consult your Personal Programming Manual, pp. IV-44 and VI-6.

If you still have problems with a program after answering the above questions, please **clearly document your problems on a PPX Program Memo form.** This form is provided **specifically** for this purpose.

4. The January 1979 issue of the PPX Exchange contained information concerning renewal of PPX-59 memberships. We are pleased that so many members are choosing to continue receiving the benefits offered by PPX-59.

The renewal table is reprinted for those members whose memberships are about to expire. To find your renewal date, check the last four digits of your membership number against the table shown below. Your membership number corresponds with your original membership date.

Membership number	Must be postmarked by
0001-4842	February 15
4843-6014	March 15
6015-7056	April 15
7057-7525	May 15
7526-8257	June 15
8258-8923	July 15

Members with numbers greater than 8923 will be informed of their renewal dates in a future issue of PPX Exchange.

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 **$15 check or money order (only).** Be sure to include your membership number on **both** your subscription card and check.

WASHINGTON, D.C.'ERS DISCERN PROGRAMMER'S 'POINTS OF VIEW'

Last year, ten PPX-59 members located within the Washington, D.C. area formed a local TI-59 Programming club under the coordination of Maurice E.T. Swinnen. Since then their membership has risen to include 25 professionals who meet once a month to share programming ideas, routines, and tricks. Although they all share a common interest in the TI-59, their professions vary widely. Members of the club include: 14 engineers, 5 professional programmers, a surveyor, an insurance salesman, 2 physicists, a psychologist, and a newpaper reporter.

If you live in the Washington, D.C. area and would like to join this group, you may contact:

Mr. Maurice E.T. Swinnen
9213 Lanham Severn Rd.
Lanham, Md. 20801
Home (301) 459-5458 / Work (301) 427-5040

As one of their many activities, this group put together a "TI-59 programmer evaluation test". It's based on the premise that individuals solve programming problems differently depending upon their profession and programming experience. This quick test provided us with alot of laughs at PPX-59.

To discover your programming 'point of view' solve the following problem, then, turn to "What's Your Programming Point of View?" on the inside page of this newsletter.

The display contains either a 1 or a 2. Write a program that leaves a 1 in the display if the previous content was a 2, and vice-versa.

Don't peek until you have worked out a solution.

Editor's Note: Please do not send your results to PPX-59. The above is presented for your enjoyment only.

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 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:

- A program to calculate necessary design parameters for Audio H Pad Attenuators.
- A program to analyze a four bar linkage. Given the crank displacement, the program would calculate the angle, position, velocity, and acceleration of the follower.
- A comprehensive diagnostic program for the TI-59 such as "Diagnostic A & B" (PPX-52 #900087D) was for the SR-52.
- Using Crandall's method, a program to balance latitudes, departures, and course length without changing the bearing of the course.

CATALOG AND PROGRAM UPDATE

We would like to keep our members abreast of the latest corrections regarding the PPX-59 Software Catalog and PPX-59 Programs. Whereas the C Addendum added new programs, this article describes those programs deleted from the PPX-59 Library. **Please note these deletions in your July 1978 Catalog:**

088003B	Annual Property Operating Data
208002A	Regression X'X,Y/Multiple Linear Regression
208007A	Multiple Linear Regression-5 Ind. Variables
918001A	Mortar Fire
938001A	Robbins Distance Formula

The following program corrections were obtained through the PPX-59 Program Memo system. All existing copies of these programs have been updated at PPX; however, some members may have a program which was ordered before these corrections were obtained. If your copy is not updated, we ask that you update it and thank you for doing so. (Please see pages IV-21, V-48, and 51 of Your Personal Programming Manual to aid you in this task.)

228011B Fisher's Exact 2x2 Test
Pages 6 and 7, to handle when one of the n's = 0, change all x=t instructions to $x \geq t$ instructions at the following locations: 130, 153, 179, 207, 234, 256, 283, 311, and 338.

348005B 20 Point Gaussian Numerical Integration
Page 4, step 3 should be press B (instead of A); Page 5, register 10's first four digits should be .4617 (instead of .4671).

398013B Solve A/B=C/D
Page 5, the instruction at location 048 should be 03 (instead of 04).

618004B Hydraulic Loop Calculations (Flow)
Page 7, the instruction at location 046 should be $x \geq t$ (instead of x=t).

638003A EBCDIC Code Converter
Pages 3 and 5, insert the folowing at the beginning of the instructions (i.e., before pressing D): GTO 18 LRN 2 LRN.

638004A ASCII Code Converter
Page 8, the instruction at location 464 should be RST.

738001A Heating Load Calculations
Page 7, the instruction at location 062 should be 08 (instead of 09). This correction will result in changing the output obtained in the Sample Problem — the final answers should be RCL 11 (23886.44 displayed) and RCL 13 (24194.99778 displayed).

778910A Horizontal Curve Design
Page 12, insert the instructions GTO 00 48 between locations 225 and 226, the instruction at location 256 should be 93 (instead of 94).

918020B Skydiving
Page 7, to make the descent curve more realistic, insert the instruction INV between locations 163 and 164. This change will alter the sample problem results.

918033B Space Chase
Page 8, the instructions at locations 160 and 165 should be 09 (instead of 02), delete the instructions LBL E at locations 169 and 170, insert the instructions LBL E between locations 163 and 164; Page 10, register 45's contents should be 1617363735.

918038B Horse Race
Page 7, the instruction at location 046 should be x (instead of CLR).

DATA ENTRY MADE SIMPLE
(THE ART OF UNPACKING THE DISPLAY)

Marvin Hinshaw

The article "The Art of Data Packing" (November 1978, PPX Exchange) dealt with a method to pack and unpack data in storage registers. The same idea can be applied to enable easy data entry.

I developed a routine to ease the tedium of averaging grades (using a 0-9 scale). Previously, if there were ten grades to be averaged, I had to enter each grade individually and press R/S. With this routine, up to ten grades, in the format "$g_1\ g_2\ g_3\ \ldots\ g_{10}$" ($g_1 \neq 0$), may be entered as a single number. By pressing A, the program automatically unpacks the grades and displays the average. The routine is:

```
000  76 LBL
001  11  A
002  29 CP
003  32 X:T
004  42 STO
005  01  01
006  42 STO
007  02  02
008  32 X:T
009  76 LBL
010  55  ÷
011  55  ÷
012  01  1
013  00  0
014  95  =
015  42 STO
016  03  03
017  22 INV
018  59 INT
019  65  ×
020  01  1
021  00  0
022  95  =
023  44 SUM
024  02  02
025  69 OP
026  21  21
027  43 RCL
028  03  03
029  59 INT
030  22 INV
031  67  EQ
032  55  ÷
033  43 RCL
034  02  02
035  55  ÷
036  43 RCL
037  01  01
038  95  =
039  92 RTN
```

For example, if the grades are 2,4,6,0, and 8, simply enter "24608", press A, and the average "4" will be displayed. The algorithm used is simple. The number "24608" is divided by 10 to give "2460.8" (begins at step 009) which is stored in register 03. The ".8" is then peeled off and multiplied by 10 to restore it to its original form, "8". This grade is summed into register 02 and register 01 is incremented by 1. The Int instruction then truncates "2460.8" (in register 03) to "2460" (step 027) and the procedure is repeated until all grades are unpacked, added, and counted (step 030), at which time the average is computed.

The advantages of entering many individual numbers at one time need not be restricted to single digit positive integers or to averaging grades. For example, let's say we have some **unordered** data which we would like to store in registers 10, 11, 12 . . . .

To do this, we must first define the characteristics of the data. Let x denote the maximum number of digits to the left of the decimal point and y denote the maximum number of digits to the right of the decimal point in a given set of numbers. Please note that:

(1) Either x or y can be defined as 0.

(2) If a number's x or y characteristic is less than the maximum x or y, leading or trailing zeroes must be added.

(3) The sum of x and y could equal 1 (allowing 10 separate numbers as one entry), 2 (allowing 5 separate numbers as one entry), or 3 (allowing 3 separate numbers as one entry). These combinations are dictated as the display can only contain up to 10 digits at one time.

(4) This particular routine will not handle ordered data.

For example, if our data consisted of the following set of numbers: **2.6, 3, .4, 7.8, 3.2, .03,** then x=1 and y=2. Since x+y=3, a maximum of 3 separate numbers is allowed in one entry. As our data consists of six numbers, we may enter the data with 2 entries. The following routine should be entered in LRN mode:

```
000  76 LBL
001  11  A
002  29 CP
003  01  1
004  00  0
005  42 STO
006  00  00
007  00  0
008  42 STO
009  01  01
010  42 STO
011  02  02
012  91 R/S
013  76 LBL
014  12  B
015  55  ÷
016  03  3
017  22 INV
018  28 LOG
019  95  =
020  42 STO
021  03  03
022  22 INV
023  59 INT
024  65  ×
025  01  1
026  22 INV
027  28 LOG
028  95  =
029  72 ST*
030  00  00
031  69 OP
032  20  20
033  43 RCL
034  00  00
035  59 INT
036  32 INV
037  67  EQ
038  12  B
039  92 RTN
```

Press A to initialize the program, then enter "**260300040**", press B, enter "**780320003**", and press B again. The six numbers are now stored in registers 10-15. To check this, recall the contents of each register. They should appear as:

```
  0.4   10
    3.  11
  2.6   12
 0.03   13
  3.2   14
  7.8   15
```

To use this routine to enter other groups of unordered data, only two steps need be changed. Step 016 is the x+y value and step 025 is the x value of the data to be entered. (In addition, locations 003 and 004 define the first register into which the data is to be stored.)

Clearly, the advantages of this routine are inversely proportional to the sum of x and y. However, for data containing one or two digits, it can be a real time saver. Hopefully, you can incorporate a similar routine into your programs to relieve the tedium of data entry.

WHAT'S YOUR PROGRAMMING POINT OF VIEW?

(1)

```
000  76 LBL
001  11  A
002  82 HIR
003  01  01
004  03  3
005  42 STO
006  01  01
007  82 HIR
008  11  11
009  50 I×I
010  59 INT
011  42 STO
012  00  00
013  76 LBL
014  37 P/R
015  69 OP
016  31  31
017  97 DSZ
018  00  00
019  37 P/R
020  43 RCL
021  01  01
022  91 R/S
```

(2)

```
000  29 CP
001  75  -
002  01  1
003  95  =
004  67  EQ
005  00  00
006  09  09
007  01  1
008  91 R/S
009  02  2
010  91 R/S
```

(3)

```
000  76 LBL
001  11  A
002  35 1/X
003  22 INV
004  38 SIN
005  85  +
006  06  6
007  00  0
008  95  =
009  38 SIN
010  35 1/X
011  91 R/S
```

(4)

```
000  76 LBL
001  15  E
002  42 STO
003  01  01
004  03  3
005  22 INV
006  44 SUM
007  01  01
008  43 RCL
009  01  01
010  50 I×I
011  91 R/S
```

(5)

```
000  76 LBL
001  11  A
002  22 INV
003  52 EE
004  22 INV
005  57 ENG
006  22 INV
007  58 FIX
008  59 INT
009  50 I×I
010  42 STO
011  00  00
012  75  -
013  01  1
014  95  =
015  29 CP
016  67  EQ
017  42 STO
018  43 RCL
019  00  00
020  75  -
021  02  2
022  95  =
023  67  EQ
024  43 RCL
025  00  0
026  35 1/X
027  92 RTN
028  76 LBL
029  42 STO
030  02  2
031  92 RTN
032  76 LBL
033  43 RCL
034  01  1
035  92 RTN
```

(6)

```
000  76 LBL
001  11  A
002  75  -
003  01  1
004  95  =
005  42 STO
006  00  00
007  00  0
008  32 X:T
009  43 RCL
010  00  00
011  67  EQ
012  87 IFF
013  75  -
014  01  1
015  95  =
016  67  EQ
017  88 DMS
018  91 R/S
019  76 LBL
020  87 IFF
021  02  2
022  91 R/S
023  76 LBL
024  88 DMS
025  01  1
026  91 R/S
```

(7)

```
000  76 LBL
001  11  A
002  32 X:T
003  01  1
004  67  EQ
005  12  B
006  91 R/S
007  76 LBL
008  12  B
009  02  2
010  91 R/S
```

(8)

```
000  76 LBL
001  11  A
002  75  -
003  03  3
004  54  )
005  94 +/-
006  92 RTN
```

(9)

```
000  76 LBL
001  11  A
002  42 STO
003  00  00
004  01  1
005  42 STO
006  01  01
007  02  2
008  42 STO
009  02  02
010  43 RCL
011  01  01
012  32 X:T
013  43 RCL
014  00  00
015  67  EQ
016  00  00
017  26  26
018  43 RCL
019  02  02
020  32 X:T
021  43 RCL
022  00  00
023  67  EQ
024  00  00
025  29  29
026  43 RCL
027  02  02
028  92 RTN
029  43 RCL
030  01  01
031  92 RTN
```

(10)

```
000  76 LBL
001  11  A
002  52 EE
003  22 INV
004  52 EE
005  29 CP
006  75  -
007  01  1
008  95  =
009  67  EQ
010  00  00
011  17  17
012  22 INV
013  67  EQ
014  00  00
015  16  16
016  01  1
017  02  2
018  91 R/S
```

(11)

```
000  76 LBL
001  11  A
002  42 STO
003  01  01
004  43 RCL
005  01  01
006  75  -
007  01  1
008  95  =
009  42 STO
010  00  00
011  43 RCL
012  00  00
013  29 CP
014  67  EQ
015  12  B
016  43 RCL
017  00  00
018  29 CP
019  75  -
020  02  2
021  95  =
022  67  EQ
023  13  C
024  01  1
025  91 R/S
026  76 LBL
027  12  B
028  86 STF
029  00  00
030  61 GTO
031  14  D
032  76 LBL
033  13  C
034  86 STF
035  01  01
036  61 GTO
037  14  D
038  76 LBL
039  14  D
040  87 IFF
041  00  00
042  15  E
043  87 IFF
044  01  01
045  10 E'
046  91 R/S
047  76 LBL
048  15  E
049  02  2
050  91 R/S
051  76 LBL
052  10 E'
053  01  1
054  91 R/S
```

(12)

```
000  76 LBL
001  11  A
002  33 X²
003  42 STO
004  00  00
005  01  1
006  00  0
007  69 OP
008  17  17
009  02  2
010  32 X:T
011  43 RCL
012  00  00
013  77  GE
014  00  00
015  19  19
016  43 RCL
017  99  99
018  91 R/S
019  43 RCL
020  98  98
021  91 R/S

1.    98
2.    99
```


(1) **JOB SECURITY MINDED.** A true work of art. It is also incomprehensible to anyone but the originator.

(2) **FORMER SR-56 OWNER.** Although this is a very short and efficient routine, the user-defined keys have not yet been discovered. Every TI-59 program starts with RST R/S.

(3) **SURVEYOR.** Begins every problem by first converting to bearing or azimuth. Upon completion, the documentation carefully hides all clues of what the program is doing.

(4) **RECENT CONVERT FROM RPN.** Fondness for the E (Enter) key is demonstrated by the use of LBL E (versus LBL A). Note the obvious absence of parentheses and equal signs.

(5) **SYSTEMS PROGRAMMER.** This is actually a very good program. Every conceivable error trap has been included in case the user makes a mistake. This type of programmer will probably move on to an IBM 370.

(6) **OLD-LINE SR-52 PROGRAMMER.** This programmer still needs several steps for every test. Moving up to a 960 step programmable may still not solve his continuous quest for more memory!

(7) **SEMI-ADVANCED PROGRAMMER.** It's apparent that this programmer is about halfway through the owner's manual and has a good grasp of the basics. Most of his waking hours are spent trying to find uses for the TI-59.

(8) **MATHEMATICIAN.** This is a concise algorithmic solution to the problem in which optimization is the name of the game.

(9) **COMPUTER SCIENTIST.** This is likely to be the most sophisticated solution to the problem. One can only hope that once fully documented some light will be shed on what has been done.

(10) **PROGRAMMING INSTRUCTOR.** You will notice a number of key points demonstrated by this program. Chances are that this is the first program ever written from beginning to end. Unfortunately, it will not always work.

(11) **BEGINNER.** Knows a little about programming and is very proud of the sophisticated use of the flags. Thank goodness for user-defined keys!

(12) **ENGINEER.** Insists upon finding a way to put a square peg in a round hole. Doing this, often involves making the problem more complicated than it really is.

NUMERICAL INTEGRATION

This program may be used to approximate the integral, I, of a function defined by the user, over an interval x_o to x_n, using either Simpson's Rule, Trapezoidal Rule, or the Gaussian Quadrature Method.

$$I = \int_{x_o}^{x_n} f(x)dx$$

The function f(x) must be expressed as a sequence of keystrokes entered into program memory prior to execution. The Master library module is required. Run time depends upon the number of sub-intervals used.

PPX wishes to thank Walter Luke Jr. for his excellent program.

USER INSTRUCTIONS:

1. Enter program.
2. Go to location 296 by pressing GTO E'.
3. Enter f(x) as a sequence of keystrokes, ending with INV SBR. Do not use =, CLR, or registers 0-14.
4. Exit Learn mode.
5. Enter lower limit $f(x_o)$, press A.
6. Enter upper limit $f(x_n)$, press B.
7. Enter number of sub-intervals to be used and compute the integral approximation by pressing C for Simpson's Rule, D for Trapezoidal Rule, or E for Gaussian Quadrature Method.
8. For a new number of sub-intervals or for an approximation of the interval using one of the remaining methods, repeat steps 5 through 7.

EXAMPLE:

Evaluate $\int_0^{\pi/2} \frac{1}{\cos x + 2}\,dx$ using two subintervals

Enter	Press	Display	Comments
	GTO E'	0.	
	LRN	296 00	
	Rad*	297 00	Radian mode
	(cos	299 00	f(x)
	+ 2)	302 00	
	1/x	303 00	
	INV SBR	304 00	
	LRN	0.	
0	A	.3333333333	x_o
π÷2=	B	.5	x_n
2	C	.604998903	Simpson's Rule
0	A	.3333333333	x_o
π÷2=	B	.5	x_n
2	D	.6173737946	Trapezoidal Rule
0	A	.3333333333	x_o
π÷2=	B	.5	x_n
2	E	.6045997881	Gaussian Quadrature (Time: 30 sec.)

**Radian Mode must be set when trigonometric functions are used.*

TI-59 LISTING

000	47	CMS	033	10	10	066	03	03	099	42	STO	132	76	LBL	165	06	6	198	03	3	231	04	4	264	42	STO
001	91	R/S	034	61	GTO	067	95	=	100	02	02	133	52	EE	166	01	1	199	06	6	232	09	9	265	06	06
002	76	LBL	035	78	Σ+	068	81	RST	101	16	A'	134	93	.	167	42	STO	200	00	0	233	02	2	266	22	INV
003	16	A'	036	76	LBL	069	76	LBL	102	44	SUM	135	05	5	168	10	10	201	07	7	234	04	4	267	97	DSZ
004	61	GTO	037	89	π	070	68	NOP	103	14	14	136	65	×	169	93	.	202	06	6	235	42	STO	268	05	05
005	02	02	038	22	INV	071	53	(	104	92	RTN	137	71	SBR	170	04	4	203	01	1	236	11	11	269	43	RCL
006	96	96	039	86	STF	072	43	RCL	105	76	LBL	138	68	NOP	171	06	6	204	05	5	237	71	SBR	270	76	LBL
007	76	LBL	040	01	01	073	02	02	106	13	C	139	42	STO	172	07	7	205	07	7	238	78	Σ+	271	42	STO
008	78	Σ+	041	92	RTN	074	87	IFF	107	42	STO	140	12	12	173	09	9	206	03	3	239	43	RCL	272	43	RCL
009	43	RCL	042	76	LBL	075	00	00	108	05	05	141	86	STF	174	01	1	207	42	STO	240	04	04	273	02	02
010	12	12	043	87	IFF	076	69	OP	109	36	PGM	142	00	00	175	03	3	208	11	11	241	65	×	274	42	STO
011	65	×	044	02	2	077	75	-	110	09	09	143	93	.	176	09	9	209	71	SBR	242	43	RCL	275	01	01
012	43	RCL	045	22	INV	078	43	RCL	111	13	C	144	05	5	177	03	3	210	78	Σ+	243	12	12	276	85	+
013	10	10	046	49	PRD	079	01	01	112	36	PGM	145	65	×	178	04	4	211	93	.	244	95	=	277	43	RCL
014	85	+	047	14	14	080	95	=	113	09	09	146	71	SBR	179	06	6	212	09	9	245	92	RTN	278	03	03
015	43	RCL	048	76	LBL	081	92	RTN	114	14	D	147	68	NOP	180	42	STO	213	03	3	246	76	LBL	279	95	=
016	13	13	049	44	SUM	082	76	LBL	115	81	RST	148	42	STO	181	11	11	214	02	2	247	15	E	280	42	STO
017	95	=	050	43	RCL	083	69	OP	116	76	LBL	149	13	13	182	71	SBR	215	04	4	248	42	STO	281	02	02
018	16	A'	051	03	03	084	85	+	117	14	D	150	22	INV	183	78	Σ+	216	06	6	249	05	05	282	71	SBR
019	65	×	052	44	SUM	085	43	RCL	118	42	STO	151	86	STF	184	93	.	217	09	9	250	35	1/X	283	52	EE
020	43	RCL	053	01	01	086	01	01	119	05	05	152	00	00	185	06	6	218	05	5	251	65	×	284	44	SUM
021	11	11	054	43	RCL	087	95	=	120	35	1/X	153	00	0	186	06	6	219	01	1	252	71	SBR	285	06	06
022	95	=	055	01	01	088	92	RTN	121	65	×	154	42	STO	187	01	1	220	04	4	253	68	NOP	286	97	DSZ
023	44	SUM	056	16	A'	089	76	LBL	122	71	SBR	155	04	04	188	02	2	221	02	2	254	42	STO	287	05	05
024	04	04	057	44	SUM	090	11	A	123	68	NOP	156	93	.	189	00	0	222	42	STO	255	03	03	288	42	STO
025	87	IFF	058	14	14	091	42	STO	124	42	STO	157	02	2	190	09	9	223	10	10	256	85	+	289	76	LBL
026	01	01	059	97	DSZ	092	01	01	125	03	03	158	03	3	191	03	3	224	93	.	257	43	RCL	290	43	RCL
027	89	π	060	05	05	093	16	A'	126	01	1	159	08	8	192	08	8	225	01	1	258	01	01	291	43	RCL
028	86	STF	061	44	SUM	094	42	STO	127	22	INV	160	06	6	193	06	6	226	07	7	259	95	=	292	06	06
029	01	01	062	43	RCL	095	14	14	128	44	SUM	161	01	1	194	05	5	227	01	1	260	42	STO	293	81	RST
030	01	1	063	14	14	096	92	RTN	129	05	05	162	09	9	195	42	STO	228	03	3	261	02	02	294	76	LBL
031	94	+/-	064	65	×	097	76	LBL	130	61	GTO	163	01	1	196	10	10	229	02	2	262	71	SBR	295	10	E'
032	49	PRD	065	43	RCL	098	12	B	131	87	IFF	164	08	8	197	93	.	230	04	4	263	52	EE			

FROM THE ANALYST'S DESK

• In the last issue of the PPX Exchange, we recognized 5 PPX-59 members for their extensive program contributions. Since then, it has come to our attention that 3 other members deserve equal recognition for their efforts. For this reason, we would like to congratulate and present complimentary one year memberships to: Milton L. Brown, Chorman W. Ching, and David Rodabaugh.

• PPX would like to apologize to Charles Kluepfel for not including his program "Sunrise/Sunset Eclipse Limits" (PPX-59 #788020) in the C Addendum to the PPX-59 Software Catalog. This program calculates the latitudes and longitudes of points on the earth's surface where an eclipse is beginning or ending at sunrise or sunset. Uses the Besselian eclipse elements produced by PPX-59 Programs #788006B and #788007B. This program is **currently** available.

• With the publication of the December 1978 Addendum, many new and interesting programs crossed the Analyst's Desk. We would like to share (with you) a couple of the programs which caught our fancy.

"Precision Fraction to Decimal Conversion" (PPX-59 #398054C) calculates the decimal representation of a fraction whose numerator and/or denominator contains up to 13 digits. The answer can be carried to **unlimited places** in groups of 5 digits. If the fraction's numerator and/or denominator exceeds 13 digits then "Precision Division" (PPX-59 #398085C) is available which will produce a decimal representation up to **45 digits in length.**

"Roots of a Quartic Equation" can be computed easily using PPX-59 Program #398070C. Utilizing Ferrari's method, this program computes all four roots, whether real or imaginary.

"Binary Distillation" (PPX-59 #618018C) will perform tray to tray binary distillation calculations allowing for both ideal and nonideal systems. This program will save the user time and improve accuracy over graphical and "short cut" analytical methods.

If you enjoy playing games which require grids, "Utility Grids" (PPX-59 #908054C) will be an asset. Drawing grids freehand will be a thing of the past.

The subject of hierarchy has been covered in recent issues of the PPX Exchange. PPX now offers a program, which provides **easy** access to the hierarchy registers, called "Hierarchy Register Functions" (PPX-59 #908063C).

As you are aware, PPX has numerous programs dealing with specific conversions. However, if you have certain conversions which are frequently encountered in your profession, you can customize your own conversions with "Expandable Conversion Program" (PPX-59 #908093C) by storing the necessary conversion factors.

If you are a betting man, "Automatic Blackjack with Options" (PPX-59 #918056C) will provide you with a comprehensive blackjack game.

PPX provides many programs to aid your children in developing and testing their arithmetic skills. **Now it's your turn** — "Arithmetic Game" (PPX-59 #918070C) provides a tough and challenging set of problems on the adult level. Can you score as well as your children do?

• In order to achieve a full sequence of 199017 numbers in the random number generators in the Master Library, Applied Statistics Library, and Math/Utilities Library (i.e., ML-15, ST-02, and MU-12), Step 1 of the User Instructions must be **replaced** as follows:

STEP	PROCEDURE	ENTER	PRESS
1a.	Initialize		RST
1b.	Enter Learn Mode		LRN
1c.	Enter Appropriate Keystrokes as Shown Below		(See Below)
1d.	Exit Learn Mode		LRN

All subsequent user instruction steps remain unchanged.

MASTER LIBRARY KEYSTROKES

000	76	LBL	018	85	+	036	15	15	054	36	PGM
001	10	E'	019	93	.	037	11	A	055	15	15
002	36	PGM	020	05	5	038	92	RTN	056	71	SBR
003	15	15	021	54	)	039	76	LBL	057	00	00
004	10	E'	022	59	INT	040	12	B	058	90	90
005	92	RTN	023	42	STO	041	36	PGM	059	92	RTN
006	76	LBL	024	09	09	042	15	15	060	76	LBL
007	15	E	025	36	PGM	043	12	B	061	13	C
008	36	PGM	026	15	15	044	92	RTN	062	71	SBR
009	15	15	027	71	SBR	045	76	LBL	063	88	DMS
010	15	E	028	00	00	046	18	C'	064	36	PGM
011	92	RTN	029	02	02	047	70	RAD	065	15	15
012	76	LBL	030	67	EQ	048	71	SBR	066	71	SBR
013	88	DMS	031	88	DMS	049	88	DMS	067	00	00
014	29	CP	032	92	RTN	050	42	STO	068	59	59
015	53	(	033	76	LBL	051	08	08	069	92	RTN
016	43	RCL	034	11	A	052	71	SBR			
017	09	09	035	36	PGM	053	88	DMS			

APPLIED STATISTICS LIBRARY KEYSTROKES

000	76	LBL	018	85	+	036	02	02	054	36	PGM
001	10	E'	019	93	.	037	11	A	055	02	02
002	36	PGM	020	05	5	038	92	RTN	056	71	SBR
003	02	02	021	54	)	039	76	LBL	057	00	00
004	10	E'	022	59	INT	040	12	B	058	89	89
005	92	RTN	023	42	STO	041	36	PGM	059	92	RTN
006	76	LBL	024	09	09	042	02	02	060	76	LBL
007	15	E	025	36	PGM	043	12	B	061	13	C
008	36	PGM	026	02	02	044	92	RTN	062	71	SBR
009	02	02	027	71	SBR	045	76	LBL	063	88	DMS
010	15	E	028	00	00	046	14	D	064	36	PGM
011	92	RTN	029	02	02	047	70	RAD	065	02	02
012	76	LBL	030	67	EQ	048	71	SBR	066	71	SBR
013	88	DMS	031	88	DMS	049	88	DMS	067	00	00
014	29	CP	032	92	RTN	050	42	STO	068	59	59
015	53	(	033	76	LBL	051	11	11	069	92	RTN
016	43	RCL	034	11	A	052	71	SBR			
017	09	09	035	36	PGM	053	88	DMS			

MATH/UTILITIES LIBRARY KEYSTROKES

000	76	LBL	007	93	.	014	12	12	021	11	A
001	11	A	008	05	5	015	11	A	022	36	PGM
002	29	CP	009	54	)	016	67	EQ	023	13	13
003	53	(	010	59	INT	017	11	A	024	12	B
004	43	RCL	011	42	STO	018	92	RTN	025	92	RTN
005	09	09	012	09	09	019	76	LBL			
006	85	+	013	36	PGM	020	12	B			

The PPX Exchange 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