



PPX Exchange

Vol. 5 Number 2 Copyright 1981

March/April 1981

PPX-59 Survey

Your Answers Are Important!

Included with this newsletter is a copy of the 1981 PPX survey. We urge you to please fill this out at your earliest convenience. If you respond before June 30, 1981, you will receive a coupon good for 25% off your next purchase of a PPX Pakette and a chance to win one of five Texas Instruments 800 series digital watches to be given away.

This survey is very important to PPX because it lets us know how well we are doing in providing our members with the best possible service. It also helps us know what additional PPX services/products you may want. Your answers are important!

Thanks For Your Support!

The Exchange would like to take this opportunity to thank all those members who have contributed their articles and ideas to the Newsletter. We especially thank the following members who responded to our call for feature articles in the March/April 1980 issue:

Douglas P. Anderson	Jeremy Sprague
J.H. Barnhart	Gregory Stark
Otis F. Bryan, Jr.	Robert Sutliff
Forrest Chambers	George Vogel
Charles Cole	Robert Wyer
Jose Miguel G. Garcia	

Each of them have received their choice of either a complimentary one year PPX membership or Solid State SoftwareTM library module.

The module/membership offer is still open to those who would like to contribute feature articles. There is always room in the newsletter for more good articles.

When submitting an article, keep to a maximum of 4 typed double-spaced pages. Examples of keystroke sequences should be included to aid the reader where appropriate. Also send magnetic cards if there is a program with the article.

Send your feature article, routines, or other items of interest to:

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

Clearing Your Memory

By Robert Wyer

Not being the most experienced programmer in these parts, I have been known to program myself into a corner. It has been not unlike painting myself into a corner. Much of the time I have to wait until a stroke of genius saves me from a complete rewrite. The famous Minnesota Fats when asked how he wins so many games simply responded that he tries never to get into a predicament where it takes a great shot in order to win the game. Trick shots are neat but they do not always make a winner. The same holds true in programming.

Having just read the recent article on Optimization, I tried to think of a good practical example. A practical example is planning ahead for the clearing of data registers. Given the condition that a program can be run several times in succession, that certain values can be reused in subsequent program execution, and that some registers are used for such things as summation, there are several good methods for accomplishing such a task.

The first method is the most straight forward and probably the method most used. In this case the programmer simply stores "0" in the registers he wishes to clear. This method works very well except when there are many registers to be cleared.

When many registers have to be cleared, the command of choice is usually CMs. The only problem with the clear memories key is that it does exactly what it says. It clears all the memories. The solution to the problem is to place the reusable values where they cannot be destroyed and then execute the CMs. There are several good methods which do incorporate the use of the CMs key.

If only one piece of data need be retained, recall it into the display, execute clear memories, then store it back into its original data register. If you have no more than eight pieces of data, they can be stored in hierarchy registers 1-8. After the values are stored in hierarchy, clear memories, then recall the values and store them back into their regular data registers. This last method presupposes that no pending operations exist in the hierarchy registers. (For information as to how to store and recall hierarchy registers I refer you to the May/June 1980 Newsletter.)

The last two methods are primarily concerned with saving several pieces of data. Again, which one is the most optimal depends on your needs and requirements. The next solution still allows for the use of the CMs key. The idea here is to

Clearing Memory

place values that may be reused on subsequent runs in the higher data registers and place the raw data and work data into the lower data registers. For example, you wish to write a program in which certain fence construction parameters are recorded. The program then calculates the amount of wire needed for a fence given the type of wire. In this case, there is always a fixed amount of input and the same number of data registers would always be used. Using this method, you would instruct the calculator to repartition to incorporate the higher numbered data registers into program memory. The clear memories could then be executed without destroying needed data. Then, after clearing the memories, the calculator could then be instructed to repartition to the original partitioning, thus reexposing the data registers for program usage. Again, using the example above, you could construct the program as follows: Registers 0-19 are allowed for work, raw data, and summation registers. Registers 20-29 contain such information as number of fence posts, number of gates, number of right-angle turns, etc. Also assume the actual program for the calculation requires only 450 steps. The program could then run in power-up partitioning (479.59). At the end of the program's output sequence and if the user requests to run the test again with another type of wire, the calculator would then be instructed as follows:

```
450 02 2
451 69 DP
452 17 17
453 47 CMS
454 06 6
```

```
455 69 DP
456 17 17
457 61 GTO
458 00 00
459 10 10
```

The GTO command is purely fictitious and could be replaced with a return statement. Of course, there are certain drawbacks in this method. All permanently stored data must occupy registers along a partitioning boundary which may or may not make the most efficient use of data memory.

Therefore, it would be nice if there was a clear memories key that would clear only those registers requested. Although this function does not exist per se, it can be constructed. In this example, all memories past the required data must be cleared. You could simply store the last data register of use or have the calculator do it. This may be done by the following sequence.

```
000 53 (
001 69 DP
002 16 16
003 22 INV
004 59 INT
005 65 *
006 01 1
```

```
007 00 0
008 00 0
009 54 )
010 42 STD
011 10 10
012 92 RTN
```

This subroutine would store the last register of the current partitioning in register 10. All the data that was required to be saved for sequential runs is placed in registers 0-9. Registers 11-? are reserved for summations of raw data, or whatever you could imagine a fence maker would require. At the end

of the program and if the user requests another run with the same basic variables, the program could transfer to the following piece of code.

```
200 76 LBL
201 12 B
202 00 0
203 72 ST*
204 10 10
```

```
205 97 DSZ
206 10 10
207 12 B
208 92 RTN
```

This method clears inclusively between two registers by first storing a "0" in the last register used, decrement the indirect register call, store a "0" in the next lower register, decrement the indirect register call, ad nauseum. When register 10 contains the value "10", a "0" will be stored in register 10. When the DSZ counter tries to decrement the register, it discovers that the loop counter register already contains a "0" and then transfers around the automatic GTO statement and executes the next command. In this case, the program would return to the control routine from which it was called.

There are many techniques available for the clearing of memories. There are probably still a few that I am presently unaware of. The idea is to know what you want to accomplish prior to beginning. This is probably the best advice I was ever given when it comes to programming. I hope it is as useful to you.

TI-59 Programming Seminar

A Texas Instruments Programming Seminar may be coming to your area. These seminars will provide beginning and intermediate programming training on the TI-59. Tuition for the two day class is \$150.00 per person. This includes the instruction, workbook and luncheon for the two days. You should supply your own TI-59. To register send your check for \$150.00 payable to Texas Instruments to:

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

If you have further questions regarding the seminar call Sherry Schroeder at 806-741-3277. The schedule for 1981 is:

SEMINAR DATES

May	7-8
May	29-30
June	4-5
June	18-19
July	1-2
July	16-17
July	30-31
August	13-14
August	27-28
September	10-11
September	24-25
October	8-9
October	22-23
November	5-6

LOCATION

Chicago, IL
Las Vegas, NV
Philadelphia, PA
Atlanta, GA
Rochester, NY
Orlando, FL
Seattle, WA
Minneapolis, MN
San Francisco, CA
Houston, TX
Denver, CO
Washington, DC
Detroit, MI
Cincinnati, OH

Letters to the Editor

Do you have comments, compliments or (shudder) complaints about PPX-59? We have always welcomed letters from our membership, and, therefore, we are providing space in each newsletter to share your views on PPX with your fellow members. Approximately 3-5 letters dealing with issues of general interest will be featured in each issue. Letters will be edited to fit the space available.

Dear Editor:

I would like to know if any other PPX members have been experiencing similar trouble as I have been with my TI-59/PC-100A and my Radio Shack® ET-300 portable telephone. My trouble is that when my TI-59/PC-100A are turned on, the portable telephone's base unit has a tendency to click and chatter. The contacts make and break. I have tried reversing plugs in each outlet but without success. If the PC-100A is on but the calculator is off the ET-300 is not affected. It seems to me that the printer/calculator combination sends out a signal that makes the portable telephone's base chatter.

If anyone has a remedy, please publish it in the newsletter.
George R. Clark, Jr.

Dear Mr. Clark and Members,

I don't think there is a simple solution to reducing the noise interference between the TI-59/PC-100A and the portable telephone. But, knowing how clever PPX members are, I'll ask for a solution. The solution would depend on what is generating the "noise". PPX member Andrew Klausman has studied the interference by placing an AM-radio close to the PC100A/C (he tuned his radio between stations). He wrote that the most notable interference was caused by the PC100A/C motor and the TI-59 display.

I phoned a Texas Instruments' engineer and technician to see if something could be done. They said that they were currently looking into a solution and that until there is one, your only course of action would be to move the TI-59/PC-100A and portable telephone further away from each other. They did say the interference was probably from the PC-100A motor.

If anybody has a solution, please send it to me at the address below and I'll forward it to Mr. Clark.

Texas Instruments PPX
P.O. Box 53
Lubbock, Texas 79401
ATTN: Newsletter Editor

Tim Janes
Editor, PPX Exchange

Dear Editor,

I would like to offer two sets of thoughts.

First, I want to express my satisfaction with the TI-59 as a computer-equivalent, particularly with the assistance of Modules 1 and 2 and PPX-59. I teach statistics and over the years have grown heavily dependent on our university com-

puter for about 20 statistical programs written by others or myself. These are in BASIC to permit interactive use on a remote time-sharing terminal. But as time-sharing use of our computer increased, its availability via terminal decreased. Therefore, I sought to reduce dependency on it by writing several equivalent programs for the TI-59, or by obtaining such programs via PPX-59. Originally I had no expectation that **all** the programs, some quite complex, could be transferred to the TI-59. Fortunately, this expectation proved wrong. I now have TI-59 programs to match each of the BASIC programs that I need. Of course, the university computer has the advantage of tremendous speed. But the TI-59 has the advantage of complete and instant availability.

Second, I would like to offer some thoughts concerning statistical programs, particularly with respect to the "Methods, Equation . . ." section. It seems that the following merit attention in the documentation and in the program:

1. Equations employed.
2. Source of such equations, and other pertinent references.
3. Accuracy of the computations. If approximations are produced, their accuracy should be indicated in terms of number of significant digits or number of decimal places, etc. If the program has room and if an extra magnetic card is not required, it may be desirable to trim the display of results to the number of accurate significant digits or decimal places. If the program produces approximations, the sample problem(s) should compare approximate with exact results.
4. Execution time. If it varies with the number or magnitude of entries, the author should try to develop a formula that enables the user to approximate execution time. At the least, execution time should be given for the sample problem(s). I suspect that more than one PPX-59 user has found himself wondering whether a purchased program is looping due to error or just takes a long time.
5. Registers where inputs are stored, so that the user can check the accuracy of an input when in doubt.
6. Registers where outputs are stored, so that a user doesn't have to repeat a computation, particularly one that takes much time. Alternatively, or also, the program can be designed to re-display the output when the R/S key is pressed or when a subroutine is activated.
7. Operating information on whether or not it is necessary to re-read the magnetic card(s) and/or operating information on whether or not it is necessary to press CMs in order to do another problem.
8. Procedure for correcting erroneous inputs.
9. Tracking of inputs when these are numerous, as in a correlation, curve-fitting, ANOVA, etc. program. That is, the display would show either the number of inputs already entered, or the number of the next input to be made.
10. Echo of inputs to verify that the desired entry has been made; i.e. the display would show the most recent entry. While this may seem inconsistent with tracking, there are several ways to resolve the inconsistency. One way, in accordance with Point 5, is to inform the user as to the registers in which inputs are stored. Another way is to provide a subroutine that either displays the last entry or tracks the entries.

Letters

11. Range of problems covered. Does a program work only for small sample sizes, for example? For samples of any size (short of the astronomical)? For something between?
12. Unacceptable inputs. For example, a program may be limited to positive integers, so that fractions and all negative numbers are unacceptable. If execution time is long, if the program has room, and if an extra magnetic card is not required, it may be desirable to write the program so that an unacceptable input quickly activates the flasher and halts execution of the program.

No doubt, attention to these matters may require considerable effort by the author of a program. But it seems worth putting emphasis on the word "professional" in PPX-59.

P.S. Nothing against the number 13, but I forgot to mention clarity in the documentation, instructions, and examples. Better long and clear than snappy and murky.

Herman Burstein
Wuntagh, NY

Dear Members,

Mr. Burstein's letter could apply to all of the categories, not just statistics. We do have a lot of very professional programs in PPX, but there is always room for improvement. The improvement might not just be complete documentation, but also adding routines to facilitate clearing memories, initializing flags, error correcting, etc.

The Exchange can serve as a means to communicate programming ideas and methods between members. For example, a good method for clearing memories was sent to us by Robert Wyer. His article appears in this issue.

A good article on the subject of writing your program with the user in mind is "Anwenderfreundlichkeit" by Maurice Swinnen. It appeared in the September 1979 issue of the Exchange (Vol. 3, No. 5). I recommend that every programmer read it. By the way, the title of the article means "friendliness towards the user".

Editor

PROGRAMMING CORNER

NEED A PROGRAM YOU CAN'T FIND IN THE CATALOG?

Since PPX is not staffed to produce custom software, we offer special incentives to those PPX members who fill a program request. Here's how it works:

1. If there are programs that you would like to see made available through PPX, send us a description of your program needs, along with your name, address and member number on a postcard or letter (no phone requests, please). Program requests should be non-trivial and should be as explicit as possible. As many of these requests as space permits are featured in this column.

2. Programs submitted to PPX to fill a Programming Corner request should be accompanied by a note so stating.
3. Programs should be postmarked by the deadline set in each Programming Corner article.
4. All programs received for the same request will be reviewed by PPX Analysts, and the author of the program which we consider the best will receive an order form entitling him to a complimentary Solid State Software™ module of his choice.
5. Other members who submitted acceptable programs (according to standard PPX criteria, Member's Guide Pg. 3) to fill the request will receive an order form entitling them to a complimentary Specialty Pakette of their choice.

PPX member, Richard Stickle responded to the request in the November/December Newsletter for a critical path program by submitting "Program Evaluation and Review Technique" (PPX #688008, See Precis column this issue). An existing program that also deals with the subject is PPX #388003.

The request for a printing estimating program has been filled by David Vequist. His program "Paper Cost and Metric Weight" (PPX #798046) is also featured in this month's Precis Column.

The program requests for this month are listed below. All programs to fill these requests should be submitted by June 30, 1981.

- A program for surveying to calculate cut out and pre-determined area.
- A program to estimate heat transfer coefficients using conventional estimation techniques when given appropriate physical properties, operating temperatures, pressures, flow rates, system geometry, and mode of heat transfer.
- A program to determine the casing and tubing design for oil and gas wells.

MEMBERSHIP RENEWALS

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

A renewal card and reminder will be sent to each member before the time to renew. Return the card to PPX with your check or money order for \$20.00. Be sure to include your membership number on both your card and your check and mail to: Texas Instruments PPX Department, P.O. Box 109, Lubbock, TX 79408.

Membership Number	Renewal Due
903932-904842	March 15
904842-906014	April 15
906015-906918	May 15
914843-915787	March 15
915788-916789	April 15
916790-917908	May 15
924326-824863	March 15
924864-925578	April 15
925577-926125	May 15

from the Analyst's Desk

• **Thirty-One: A Card Game** - If you've tried this program, you've probably discovered that it is one of the better game programs from the PPX library (PPX #918051). But alas, after playing the version given in the January/February 1980 issue of the Exchange, you may have discovered the following typo: The constant stored in register nine should be 343610.26361 instead of 342610.26361. Our thanks go to Harry W. Stern for bringing this to our attention.

• Another improvement over the rounding routine given in the November/December newsletters was sent to PPX from Palmer O. Hanson, Jr. His method is shorter than the routine given in the January/February newsletter. Also, his method shows a good application of the Fix Ind keystroke. This may be a good program for beginning programmers to study.

Enter the number to be rounded, press A. Then enter the number of decimal places desired and press R/S.

```
000 76 LBL
001 11 A
002 99 PRT
003 32 XIT
004 98 ADV
005 91 R/S
006 42 STD
007 00 00
008 58 FIX
009 40 IND
010 00 00
```

```
011 32 XIT
012 52 EE
013 22 INV
014 52 EE
015 22 INV
016 58 FIX
017 99 PRT
018 98 ADV
019 98 ADV
020 98 ADV
021 91 R/S
```

potpourri

The New Submission Forms - The new one-sided submission forms are an aid in duplicating accepted programs. The copy center which does our xeroxing requires one sided original copies. The new forms help PPX spend less time at duplicating machines. If you have both the new and old forms, we prefer that you use the new ones.

Also, we prefer that you renumber the page numbers of the new forms according to the length of your program. Simply erase or use liquid paper to erase the existing page numbers and then renumber your submission with the sections in the following order:

1. Submission Abstract
2. Program Description
3. User Instructions
4. Sample Problem
5. Listing

Insert continuation sheets where necessary. If you need extra pages and are out of forms, either write us a letter asking for more forms, or use typewriter paper.

Archimedes Cattle Problem Solved Using A TI-58

D.H. Fowler of the University of Warwick Mathematics Institute solved Archimedes Cattle problem using a TI-58. This was no easy task since the answer yields 7.760271406... ¹⁰²⁰⁶⁵⁴⁴ cattle. To give you an idea of the magnitude of this number, Mr. Fowler said that it would fill 24 sheets of unspaced computer print-out, although the accuracy of his answer was limited by the 13 digit capacity of the calculator.

Four programs were used to assist in solving the problem: (1) Master Library Program ML-02 was used to solve simultaneous equations; (2) a number factoring program; (3) Program to evaluate continued fraction expansion of Quadratic surds, and, (4) a program to evaluate continued fractions of rational numbers.

Mr. Fowler said a TI-59 and printer could have lessened some of the bookkeeping and keypunching in the 10 days he spent computing the solution, but termed them "luxuries". He also cautions anyone thinking about solving the program to have some knowledge of continued fractions and a long weekend to spare.

Given below is only the **very beginning** of the problem. The interpretation of the problem yields seven equations with eight unknowns:

If thou art diligent and wise, O stranger, compute the number of cattle of the Sun, who once upon a time grazed on the fields of the Thrinacian isle of Sicily, divided into four herds of different colours, one milk white, another a glossy black, the third yellow and the last dappled. In each herd were bulls, mighty in number according to these proportions: Understand, stranger, that the white bulls were equal to a half and a third of the black together with the whole of the yellow, while the black were equal to the fourth part of the dappled and a fifth, together with, once more, the whole of the yellow

Sound interesting? If you're brave enough, drop the Editor a line for a copy of the complete statement of the problem.

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

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

Data List

This program is an improved INV List. It lists only those registers that contain data, i.e., that have a non-zero content. It was submitted to PPX in April 1978 by Maurice Swinnen, now the Editor of TI PPC NOTES (the newsletter of an independent TI Programmable Calculator Club). The program allows paper and time to be saved for those who paste data register contents in notebooks or PPX program submissions.

The documentation of DATA LIST points out that Mr. Swinnen wrote it with the following objectives in mind:

1. It should fit within 160 program steps, so that it could list as many of the 100 data registers as possible.
2. It could not use any user-defined keys.
3. It should be reasonably fast, so it uses direct addresses.
4. Register numbers should be printed in the right hand margin, in order to mimic INV List.
5. It should be able to detect the current partition (find the highest data register used) and then it should go in a one-by-one step, rejecting to print any registers that contain exactly zero.

6. The program should warn the user when the calculator is in the 959.00 partition, and so no registers are available.

The program followed these objectives except that one data register had to be used: Register 00. All the other temporary storage is provided by the t-register and by Hierarchy registers 2, 3, and 4 (for information on what these registers are and how to use them see the article "Hierarchy Strikes Again" in the May/June 1980 issue of the Exchange).

User Instructions:

1. Enter program. When keying program in, note that the keystrokes HIRxx are entered using a special technique (where xx is the two digit number following the HIR keystroke). To key in HIRxx, first enter STO 82 STOX, BST and Del the two STO's. Then, SST past the two locations containing the HIRxx instruction.

2. Program can be recorded in any partition. If you wish to read the card into the calculator while in another partition, enter a-1 into the display and then enter your card into the card reader. A negative one should be displayed if the card reads correctly.

3. Press SBR List to print the contents of the data registers. If the partition of the calculator happens to be 959, the output will be an alternating flashing display between 959 and .111111111. Stop the display flashing by pressing the R/S key and holding it down for at least a full second. Then repartition as needed to access the data registers.

4. Remember that the program uses register 00, therefore the contents of this register cannot be printed.

Example:

Store 1234567890 into registers 99, 89, 79, 69, 59, 49, 39, 29, 19, 09. Then list the contents of these same registers using the DATA LIST program. (Sample assumes card was recorded in a partition other than 159.99).

Enter	Press	Display	Comments
10	Op 17	159.99	Set Partition.
-1	Input Card	-1	Prog. entered.
1234567890	STO 99	1234567890	
	STO 89	1234567890	

Enter	Press	Display	Comments
	STO 79	1234567890	
	STO 69	1234567890	
	STO 59	1234567890	
	STO 49	1234567890	
	STO 39	1234567890	
	STO 29	1234567890	
	STO 19	1234567890	
	STO 09	1234567890	
	SBR List		

Listed Output:

```

1234567890      99
1234567890      89
1234567890      79
1234567890      69
1234567890      59
1234567890      49
1234567890      39
1234567890      29
1234567890      19
1234567890       9
END

```

Takes 90 sec.

000	76	LBL	052	44	44	104	77	GE
001	90	LST	053	08	8	105	01	01
002	98	ADV	054	77	GE	106	29	29
003	69	DP	055	00	00	107	61	GTO
004	00	00	056	59	59	108	00	00
005	09	9	057	02	2	109	11	11
006	09	9	058	85	+	110	82	HIR
007	42	STO	059	01	1	111	12	12
008	00	00	060	75	-	112	65	x
009	82	HIR	061	59	INT	113	01	1
010	02	02	062	82	HIR	114	00	0
011	73	RC*	063	34	34	115	61	GTO
012	00	00	064	95	=	116	00	00
013	69	DP	065	82	HIR	117	46	46
014	19	19	066	02	02	118	01	1
015	87	IFF	067	01	1	119	07	7
016	07	07	068	82	HIR	120	03	3
017	00	00	069	53	53	121	01	1
018	90	90	070	82	HIR	122	01	1
019	29	CP	071	13	13	123	06	6
020	73	RC*	072	29	CP	124	69	DP
021	00	00	073	22	INV	125	04	04
022	67	EQ	074	67	EQ	126	69	DP
023	00	00	075	01	01	127	05	05
024	85	85	076	10	10	128	91	R/S
025	43	RCL	077	82	HIR	129	03	3
026	00	00	078	14	14	130	03	3
027	22	INV	079	69	DP	131	06	6
028	77	GE	080	04	04	132	04	4
029	01	01	081	73	RC*	133	01	1
030	18	18	082	00	00	134	02	2
031	32	X:T	083	69	DP	135	00	0
032	01	1	084	06	06	136	06	6
033	82	HIR	085	69	DP	137	01	1
034	03	03	086	30	30	138	02	2
035	00	0	087	61	GTO	139	69	DP
036	82	HIR	088	00	00	140	04	04
037	04	04	089	19	19	141	69	DP
038	32	X:T	090	22	INV	142	05	05
039	55	+	091	86	STF	143	09	9
040	28	LOG	092	07	07	144	35	1/X
041	59	INT	093	25	CLR	145	66	PAU
042	82	HIR	094	01	1	146	25	CLR
043	33	33	095	00	0	147	09	9
044	22	INV	096	22	INV	148	05	5
045	28	LOG	097	44	SUM	149	09	9
046	85	+	098	00	00	150	66	PAU
047	32	X:T	099	82	HIR	151	25	CLR
048	01	1	100	52	52	152	61	GTO
049	00	0	101	82	HIR	153	01	01
050	00	0	102	12	12	154	43	43
051	82	HIR	103	22	INV			

HARD-WIRED FUNCTIONS

By Don O'Grady

Editor's Note: As TI-59 programmers become more sophisticated in the use of their calculator they tend to demand and obtain increased performance by taking advantage of "how" the 59 does things. The following article, describing how the hard-wired functions work, is based upon a submission by Palmer O. Hanson, Jr.

The TI-59 uses its own "keystroke" language when executing hard-wired functions (P/R, D.MS, and statistics operations). It is for this reason that these functions require the use of pending operations and the subroutine stack. A table presenting an overview of these requirements and a description of the HIR instruction used by these functions appeared in the May/June 1980 issue of PPX Exchange. This article, besides expanding that table, will describe the how and why behind it.

One instruction, that has yet to be described in the manual or any previous PPX Newsletter, is "HIR 20." As you will see, this instruction can act as Nop, GTO or INV SBR, depending upon a condition preset by the algorithm before the sequence is called. The only effect this sequence has in main memory is to kill a live entry state.

POLAR/RECTANGULAR CONVERSIONS

Steps 284-302 perform polar to rectangular conversions using the following equations:

$$x = (\cos \Theta) R \text{ and } y = (\sin \Theta) R$$

Upon completion of the function, R is left in H₇ and Θ in H₈. (H₈ denotes hierarchy register 8. This notation is used throughout the article.)

Rectangular to polar conversions are completed by

$$R = \sqrt{x^2 + y^2} \text{ and } \Theta = \cos^{-1} \left(\frac{x}{|x|} \right) + \tan^{-1} \left(\frac{y}{x} \right).$$

Note that if x equals zero, an error state is caused by division by zero. As the results of the trig functions are still correct, the error is simply cancelled by a CE instruction.

When using a TI-58 or TI-59, x is left in H₇ and y in H₈. The TI-58C sequence, however, was written using 2 less program steps and does not require the use of H₇. For the TI-58C, x is left in H₈ and y is discarded.

ANGLE CONVERSIONS

The D.MS to D.d conversion routine is found at steps 303-340. The actual equation used is:

$$D.d = (((DD \times 60 + MM) \times .6 + .SSs)/36).$$

This is the algebraic equivalent of:

$$\text{Degrees} = (((DD \times 60 \frac{\text{Min}}{\text{Deg}} + MM) \times 60 \frac{\text{Sec}}{\text{Min}} + SS.s)/3600 \frac{\text{Sec}}{\text{Deg}})$$

MM.SSs is left in H₈ by this operation.

Steps 341-379 represent the D.d to D.MS conversion. In this case, DDMM.m is computed and stored in H₈ by

$$H_8 = ((D + .dx.6) \times .01).$$

Then,

$$DD.MMSSs = ((DDMM + .m \times .6) \times .01).$$

Due to roundoff error, full degrees or minutes may sometimes be represented as 60 minutes or seconds. For example, 1.499999999 INV D.MS yields 1.296 instead of 1.36. Upon examination of the guard digits we find that this value is actually 1.295999999964. In order to minimize this problem the calculator's algorithm discards all digits not visible in the display (including guard digits and digits suppressed by fix mode) before calling the conversion routine. This way, a value such as 1.499999999999 (seen as 1.5 when rounded in the display) provides 1.3 when INV DMS is used.

STATISTICS DATA ENTRY

The $\Sigma +$ and INV $\Sigma +$ operations are handled by steps 192-249. $\Sigma +$ skips steps 213-235 and INV $\Sigma +$ begins at 213. These functions compute and store the sums shown on page V-33 of "Personal Programming" directly.

Note that due to the $+/-$ instruction at location 213, some difficulty may arise when using INV $\Sigma +$ with data entered using scientific notation. This instruction is used to change the sign of y_i so that it may be removed from the sums being generated for use with the statistics operations. However, if INV $\Sigma +$ is executed where y_i is in scientific notation and the calculator is in a live entry state, the $+/-$ instruction changes the sign of the exponent instead of the mantissa.

Both functions leave $x_i + 1$ in H₇. $\Sigma +$ leaves $x_i y_i$ and INV $\Sigma +$ leaves $-x_i y_i$ in H₈.

MEAN, VARIANCE, AND STANDARD DEVIATION

Means are calculated by the sequence at steps 067-083. The HIR 20 instruction at the end of the sequence acts as a RTN instruction. Calculations are performed directly ($\Sigma x/N$) and $\Sigma y/N$).

OP 11 (calculation of variances) requires the use of steps 067-106. The HIR 20 instruction is treated as a NOP instruction after the calculation of the means. The equations shown on page V-34 of "Personal Programming" are used in this calculation.

The INV \bar{x} (standard deviation) function begins at step 107 and ends at 148. Again, the equations found on page V-34 of "Personal Programming" are used. Upon completion of the operation, N-1 is left in H₈.

continued on page 8

ADDRESS CHANGES

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

PPX
P.O. Box 53
Lubbock, TX 79408

Functions

LINEAR REGRESSION

Op 12 (slope and intercept) execute steps 000-046. The HIR 20 instruction at the end of this sequence acts as a RTN in this case. As illustrated on page V-37 of "Personal Programming," the slope is calculated first and used in the calculation of the y-intercept. The value held by the display register at the time this function is called is left in H₈.

Op 13 (correlation coefficient) is found at steps 149-191. The equation used to determine R is:

$$R = \frac{(\Sigma xy - \frac{\Sigma x \Sigma y}{N})}{\sqrt{(\Sigma x^2 - \frac{(\Sigma x)^2}{N})(\Sigma y^2 - \frac{(\Sigma y)^2}{N})}}$$

Ops 14 and 15 both begin execution at step 000 when the slope and intercept of the regression line is calculated. Then at the HIR 20 instruction at steps 045/046, the Op 14 instruction transfers to step 058 and completes execution at step 066 while Op 15 falls through and terminates at step 057. Results are then obtained by:

$$y' = mx = b \quad \text{and} \quad x' = (y - b) / m$$

where m and b are the slope intercept.

Note that Op 14 and Op 15 both assume that x is the independent variable. That is, Op 14 provides the value of x needed to estimate the y input to the function using the same regression line as Op 14.

Both functions leave the input value in H₈. Also, Op 14 deposits mx in the t-register and Op 15 leaves y - b.

SUMMARY

Now that we know how these functions work we can determine how to get the most out of what is available. For example, the equation for calculating standard deviations is

typically written as a direct representation of its definition using

$$s = \sqrt{\frac{[\Sigma(x_i - \bar{x})]^2}{(N - 1)}}$$

INV \bar{x} , however, uses an algebraic equivalent to the above equation,

$$s = \sqrt{\frac{\Sigma y^2 - (\Sigma y)^2 / N}{(N - 1)}}$$

This equation allows the calculator to compute an answer without storing each individual data point, but can lead to poor results if roundoff errors become significant. This problem can occur when mean is large in comparison to the deviations from the mean. Attempting to find the standard deviation of 1.0000067, 1.0000069, 1.0000071, 1.0000073, and 1.0000075 for example causes an error because roundoff has caused the calculator to attempt the square root of a negative number. Knowing what caused the problem, we can still find the standard deviation of these data points by subtracting 1 from each value before entering it via $\Sigma +$. This transformation simply shifts the mean 1 unit and allows us to compute the correct standard deviation.

As another example, the equation actually used for Op 13 is an algebraic equivalent of the equation presented on V-37 of "Personal Programming." However, due to rounding during calculations these equations are not always computational equivalents. The difference is often negligible, as in computing the correlation coefficient for the example on page V-38 of "Personal Programming;" but knowing the exact means of computation can be beneficial in many questionable situations.

The following table details the requirements of these functions. This table and the other information provided in this article should prove to be beneficial to those wanting to get the most out of their calculator.

Function	Parenthesis	Pending Operations	t-Register	Subroutine Levels	H ₇	H ₈
P/R	1	1	R → x	1	R	θ
INV P/R	2	2	x → R	1	x*	y**
D.MS	4	2	—	1	—	MM.SSs
INV D.MS	2	2	—	1	—	DDMM.m
Σ +	0	0	x _i → x _i + 1	1	x _i + 1	x _i y _i
INV Σ +	0	0	x _i → x _i - 1	1	x _i + 1	x _i y _i
\bar{x}	1	1	\bar{x}	1	—	—
INV \bar{x}	2	2	σ_x	1	—	(N-1)
Op 11	1	2	σ_x^2	1	—	—
Op 12	2	3	m	1	—	Orig. Dsp.
Op 13	3	4	—	1	—	—
Op 14	2	3	mx	1	—	x
Op 15	2	3	y - b	1	—	y
*0 for TI-58C						
**x for TI-58C						

Code for TI-58/59 Hardware Functions*

000	82	HIR	072	03	03	144	82	HIR
001	08	08	073	54		145	18	18
002	53	(074	32	X:T	146	54)
003	53	(075	53	(147	34	FX
004	43	RCL	076	43	RCL	148	92	RTN
005	06	06	077	01	01	149	53	(
006	75	-	078	55	+	150	53	(
007	43	RCL	079	43	RCL	151	43	RCL
008	04	04	080	03	03	152	06	06
009	65	x	081	54)	153	75	-
010	43	RCL	082	82	HIR	154	43	RCL
011	01	01	083	20	20	155	04	04
012	55	+	084	33	X²	156	65	x
013	43	RCL	085	53	(157	43	RCL
014	03	03	086	94	+/-	158	01	01
015	54)	087	85	+	159	55	+
016	55	+	088	43	RCL	160	43	RCL
017	53	(089	02	02	161	03	03
018	43	RCL	090	55	+	162	54)
019	05	05	091	43	RCL	163	55	+
020	75	-	092	03	03	164	53	(
021	43	RCL	093	54)	165	53	(
022	04	04	094	32	X:T	166	43	RCL
023	33	X²	095	33	X²	167	05	05
024	55	+	096	53	(168	75	-
025	43	RCL	097	94	+/-	169	43	RCL
026	03	03	098	85	+	170	04	04
027	54)	099	43	RCL	171	33	X²
028	54)	100	05	05	172	55	+
029	53	(101	55	+	173	43	RCL
030	53	(102	43	RCL	174	03	03
031	24	CE	103	03	03	175	54)
032	65	x	104	54)	176	65	x
033	32	X:T	105	32	X:T	177	53	(
034	43	RCL	106	92	RTN	178	43	RCL
035	04	04	107	53	(179	02	02
036	94	+/-	108	53	(180	75	-
037	85	+	109	43	RCL	181	43	RCL
038	43	RCL	110	05	05	182	01	01
039	01	01	111	75	-	183	33	X²
040	54)	112	43	RCL	184	55	+
041	55	+	113	04	04	185	43	RCL
042	43	RCL	114	33	X²	186	03	03
043	03	03	115	55	+	187	54)
044	54)	116	43	RCL	188	54)
045	82	HIR	117	03	03	189	34	FX
046	20	20	118	54)	190	54)
047	53	(119	55	+	191	92	RTN
048	53	(120	53	(192	44	SUM
049	94	+/-	121	43	RCL	193	01	01
050	85	+	122	03	03	194	82	HIR
051	82	HIR	123	75	-	195	08	08
052	18	18	124	01	1	196	33	X²
053	54)	125	54)	197	44	SUM
054	55	+	126	82	HIR	198	02	02
055	32	X:T	127	08	08	199	32	X:T
056	54)	128	54)	200	44	SUM
057	92	RTN	129	34	FX	201	04	04
058	53	(130	32	X:T	202	82	HIR
059	32	X:T	131	53	(203	48	48
060	65	x	132	53	(204	82	HIR
061	82	HIR	133	43	RCL	205	07	07
062	18	18	134	02	02	206	33	X²
063	85	+	135	75	-	207	44	SUM
064	32	X:T	136	43	RCL	208	05	05
065	54)	137	01	01	209	01	1
066	92	RTN	138	33	X²	210	61	GTO
067	53	(139	55	+	211	02	02
068	43	RCL	140	43	RCL	212	36	36
069	04	04	141	03	03	213	94	+/-
070	55	+	142	54)	214	44	SUM
071	43	RCL	143	55	+	215	01	01

216	82	HIR	271	85	+	326	54)
217	08	08	272	53	(327	65	x
218	33	X²	273	82	HIR	328	93	.
219	94	+/-	274	18	18	329	06	6
220	44	SUM	275	55	+	330	85	+
221	02	02	276	82	HIR	331	82	HIR
222	32	X:T	277	17	17	332	18	18
223	22	INV	278	54)	333	22	INV
224	44	SUM	279	22	INV	334	59	INT
225	04	04	280	30	TAN	335	54)
226	82	HIR	281	54)	336	55	+
227	48	48	282	24	CE	337	03	3
228	82	HIR	283	92	RTN	338	06	6
229	07	07	284	53	(339	54)
230	33	X²	285	82	HIR	340	92	RTN
231	94	+/-	286	08	08	341	53	(
232	44	SUM	287	39	CDS	342	53	(
233	05	05	288	65	x	343	82	HIR
234	01	1	289	32	X:T	344	08	08
235	94	+/-	290	82	HIR	345	59	INT
236	44	SUM	291	07	07	346	85	+
237	03	03	292	54)	347	82	HIR
238	82	HIR	293	32	X:T	348	18	18
239	37	37	294	53	(349	22	INV
240	82	HIR	295	82	HIR	350	59	INT
241	18	18	296	18	18	351	65	x
242	44	SUM	297	38	SIN	352	93	.
243	06	06	298	65	x	353	06	6
244	82	HIR	299	82	HIR	354	54)
245	17	17	300	17	17	355	65	x
246	32	X:T	301	54)	356	01	1
247	43	RCL	302	92	RTN	357	00	0
248	03	03	303	53	(358	00	0
249	92	RTN	304	53	(359	54)
250	53	(305	53	(360	82	HIR
251	82	HIR	306	82	HIR	361	08	08
252	08	08	307	08	08	362	53	(
253	33	X²	308	59	INT	363	53	(
254	85	+	309	65	x	364	59	INT
255	32	X:T	310	06	6	365	85	+
256	82	HIR	311	00	0	366	82	HIR
257	07	07	312	85	+	367	18	18
258	33	X²	313	53	(368	22	INV
259	54)	314	82	HIR	369	59	INT
260	34	FX	315	18	18	370	65	x
261	32	X:T	316	22	INV	371	93	.
262	53	(317	59	INT	372	06	6
263	53	(318	65	x	373	54)
264	82	HIR	319	01	1	374	65	x
265	17	17	320	00	0	375	93	.
266	55	+	321	00	0	376	00	0
267	50	I×I	322	54)	377	01	1
268	54)	323	82	HIR	378	54)
269	22	INV	324	08	08	379	92	RTN
270	39	CDS	325	59	INT			

TI-58C Only:

000	82	HIR	021	43	RCL	042	43	RCL	266	54)
001	08	08	022	04	04	043	03	03	267	22	INV
002	53	(023	33	X²	044	54)	268	39	CDS
003	53	(024	55	+	045	82	HIR	269	54)
004	43	RCL	025	43	RCL	046	20	20	270	24	CE
005	06	06	026	03	03	250	53	(271	53	(
006	75	-	027	54)	251	53	(272	32	X:T
007	43	RCL	028	54)	252	24	CE	273	33	X²
008	04	04	029	53	(253	55	+	274	85	+
009	65	x	030	53	(254	32	X:T	275	82	HIR
010	43	RCL	031	29	CP	255	82	HIR	276	18	18
011	01	01	032	65	x	256	08	08	277	33	X²
012	55	+	033	32	X:T	257	54)	278	54)
013	43	RCL	034	43	RCL	258	22	INV	279	34	FX
014	03	03	035	04	04	259	30	TAN	280	32	X:T
015	54)	036	94	+/-	260	85	+	281	92	RTN
016	55	+	037	85	+	261	53	(282	00	0
017	53	(038	43	RCL	262	82	HIR	283	00	0
018	43	RCL	039	01	01	263	18	18			
019	05	05	040	54)	264	55	+			
020	75	-	041	55	+	265	50	I×I			

*The TI-58/59 code is the same as that for the TI-58C except for steps 000 through 046 (Op12) and steps 250 through 282 (INV P-R: R-P). The TI-58C code for these functions is given at the bottom of the righthand column.

Precis

This column presents some of the new PPX programs which have been recently accepted. The abstracts here are from programs that the analysts thought would be of special interest to members. You can purchase these programs at a cost of \$4.00 each. Send your order to: Texas Instruments: C/O PPX Department; P.O. Box 109; Lubbock, TX 79408. Include an additional \$2.00 to cover postage and handling.

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

098018G Payoff Matrix to Opportunity Loss Matrix

Converts a payoff matrix into an opportunity loss matrix for any payoff matrix with 45 or fewer rows (actions).

Barbara C. Hevener, Columbia, SC
149 Steps, Mod 1

098019G Payoff Matrix (Delete Dominated Actions)

This program finds and deletes inadmissible rows from a payoff matrix or from an opportunity loss matrix having up to 76 elements.

Barbara C. Hevener, Columbia, SC
223 Steps, Mod 1

228039G Binomial 2X2 Test

For two independent samples with independent proportions, the program computes the 1-Tail and 2-Tail significance levels, based on joint binomial probabilities. The pooled sample proportion provides the required estimate of the population proportion. The program can accommodate samples of any size with any proportions.

Herman Burstein, Wantagh, NY
473 Steps, Revision B

228044G Non Parametric Tests

Performs five of the most commonly used non-parametric tests: Tukey's W; Student-Newman, Keuls; Duncans Test; Dunnett's Test and least significant difference.

Guy H. Nelson, Richmond, VA
649 Steps, PC-100A, Mod 2

228052G Hypergeometric Probabilities/Fisher's Test

Produces very close approximations of hypergeometric probabilities for samples and populations of any size. Incorporates Fisher's Test, yielding 1-Tail or 2-Tail significance level. Essentially accurate to six significant digits.

Herman Burstein, Wantagh, NY
797 Steps

238010G Ratio Estimate

Given X and Y for each unit in a sample, the program produces an estimate of R as well as the lower and/or upper

confidence limit for R at the 90%, 95%, or 99% confidence level.

Herman Burstein, Wantagh, NY
630 Steps

338033G Eight Point Prony Interpolation

If 8 points are supplied, the X-values being in geometric progression, the program will determine if the data may be fitted with a formula based on the equation $Y = AX^C + BX^D + EX^F + GX^J$. This program develops an interpolation formula, and gives the coefficients, as well as interpolating.

Theodore M. Bones, Princeton, WV
637 Steps, Mod 1

338035G Difference Tables and Solutions

Interpolation between equally spaced data points along an interval by use of forward, backward and central difference tables and their associated formulas. Though more adaptable to computer technology, processing can be performed on the TI-59 with relative ease.

Philip Brassine, Seattle, WA
772 Steps, PC-100A

348027G 96 Point Gaussian Integration

This program will integrate a user specified function in an arbitrary finite interval by means of a 96 point Gaussian integration formula.

L.N. Tao, Lincolnwood, IL
290 + Steps

398205G Simultaneous Equations (6X6)

Allows inspection of columns during input of matrix and also permits convenient check of the validity of roots by plugging roots into each row and comparing the results. Program provides flexibility, features, safeguards and capabilities not available with ML-02 alone.

Raymond C. Ellis, Laurel Spring, NJ
400 Steps, Mod 1

398206G Integrals and Derivatives of Error Functions

The program computes the repeated integrals and the iterated derivatives of the error functions. It can be used for either positive or negative arguments. These integrals or derivatives of different orders can be computed either singly or collectively.

L.N. Tao, Lincolnwood, IL
358 Steps

398216G Associativity Test for Partial Groupoids

The Cayley table of a partial groupoid is tested for associativity. In the case of non-associativity, the first triple encountered for which the test fails is displayed. Also displays the consequences of the associativity law with respect to undefined products. Limitation: the order should not exceed 20.

H. Jurgensen, Reinheim 1, Germany
264 Steps, Mod 10

408050G Euler Half-Increment Method: Motion Studies

The Euler method for obtaining a numerical solution of second order differential equations is demonstrated using

Precis

Newton's second law of motion. Two different forcing functions are used in the program and provisions are made for the user to enter other forcing functions of interest. Position and velocity plots are also generated.

Paul C. Sharrah, Fayetteville, AR
790 Steps, PC-100A

408056G Bohn Hydrogenic Atom

The principal quantum numbers are entered and the energy levels and wave length of the photon are calculated using the Bohn formula. The atomic number other than 1 can be entered to calculate the equivalent quantities for hydrogenic atoms with only one electron remaining in the ionized atom. Also calculates the wave length of the K-alpha X-ray line.

Paul C. Sharrah, Fayetteville, AR
474 Steps, PC-100A

418102G Preparing a Standard Acid Solution

Unlike PPX #418036C, this program when given the specific gravity, assay composition, molarity of the acid to be prepared, volume in liters, and the mole weight will calculate the number of milliliters of a concentrated stock acid needed to prepare the acid solution. If the number of grams of 100 percent acid needed for a particular situation is known along with the first two inputs, the program will likewise determine the volume of concentrated acid needed.

Russell W. Rasmus, Manchester, IA
275 Steps, PC-100A

468013G Strata Depth and Thickness

Calculates the depth to a strata or the thickness of the strata bed. Five different cases for either depth or thickness can be calculated.

Barry J. Franz, Chevy Chase, MD
282 Steps

508061G Prediction of Myocardial Infarct

The probability of acute myocardial infarct is calculated from readily available clinical and electrocardiographic predictive variables.

Hershel Goren, Cleveland, OH
316 Steps, PC-100A

568009G Steady State Level And Multiple Elimination

This program determines the steady state level of drugs eliminated by one or more apparent first order processes plus one or two Michaelis-Menten processes in parallel.

Raymond E. Galinsky, Amherst, NY
431 Steps, PC-100A

628165G Steel Column Baseplates

This set of three programs computes the size and thickness of base plates for box and wideflange columns, and will size

square or rectangular baseplates for wideflange columns.

John Hiatt, Houston, TX
879 Steps, PC-100A

628171G Retaining Wall Design

Determines footing length; area of reinforcing required for stem, toe, and heels; overturning moment; and resisting moment against overturning and footing bearing pressure. Program accepts four loading types: lateral force at top of wall, triangular soil pressure diagram, rectangular surcharge pressure diagram, and vertical load centered on top of wall.

C. Michael Baker, Columbia, MO
941 Steps, PC-100A

628183G Rectangular Wood Column Design

This program gives substantial aid in the design of rectangular wood columns under an axial load. An iterative method is used to find the smallest allowable wood column based on a variety of governing conditions.

Timothy M. Rice, Pullman, WA
464 Steps

648106G Schmitt Trigger Design

Given the upper and lower threshold voltages, the supply voltage, E-B voltage, load impedance, and resistor tolerance, the program will compute the values of the four resistors used in the circuit. Upon re-entry of selected standard values for the resistors, the program computes the power dissipated by the load and the corrected upper and lower threshold voltages.

W.R. McCarty, Phoenix, AZ
321 Steps, PC-100A

648107G Bode and Closed Loop Calculations

Calculates gain and phase of a transfer function for any frequency. Any combination of differentiators, integrators, real zeros or poles, and quadratics (complex zero or pole pairs) can be used. Measured data can be added to define the complete open loop T.F. of a system, then closed loop gain and phase of the system are calculated.

Robert H. Plath, Linthicum, MD
590 Steps

658153G Two Tower Directional Array

This program calculates the horizontal R.M.S. field of a two tower directional AM broadcast array. Since it is limited to a two array, its run time is quite fast, and it is self prompting. In addition, any desired resolution can be specified.

Charles Kelly, Jr., Durango, CO
360 Steps, PC-100A

668133G Dynamic Balancing of Rotating Shafts

Computes the weight or location required to balance a rotating shaft after inputting the weight and location of all imbalances. It performs balancing of the configuration without introducing an additional bearing reaction. Balance weights are located in two planes, unless all unbalances occur in the same plane.

H. Greg Porter, Bellaire, OH
192 Steps

Precis

668141G Pipe Schedule and Expansion Tank Size

This program sums pipe lengths and computes system volume in a heating system given pipe sizes and lengths. Upon entering auxiliary volumes and operating conditions, an expansion tank is sized for the system.

Morris Berengut, Calgary, Canada
632 Steps, PC-100A

688008G Program Evaluation and Review Technique (PERT)

This program allows for the solving of the critical path in PERT calculations. Also allows the user to obtain the time required to complete a given project and the probability of timely completion.

Richard W. Stickle, Washington, DC
222 Steps, PC-100A

748037G Spherical Aberration for Mirrors

Given the distance of the incident ray from the optical axis and the focal length of the spherical mirror, the program calculates spherical aberration, the focal length for the minimum confusion circle, the size of the confusion circle in millimeters and seconds of arc. The incident ray is assumed to be marginal.

Rudy E. Kokich, Flushing, NY
476 Steps, PC-100A

788053G Occultation of Planets or Stars (Eclipse)

The observer's geographic coordinates and data from the American Ephemeris and Nautical Almanac are used to calculate the time of the four contacts of an occultation of a planet (or star) by the moon. The position angle of the contact and the altitude and azimuth of the moon are also calculated.

Robert B. Caldwell, Sunnyvale, CA
639 Steps

798046G Paper Cost and Metric Weight

Calculates the cost of printing papers of any size and substance, using either CWT price or M/Sheet price. Calculates for bond, book, index bristol and newsprint. Figures M weight, GM^2 , and KGM.

David V. Vequist, Pittsburg, KS
385 Steps, PC-100A

908182G Advanced Register Manipulations

This program will (1) copy a continuous block of data registers into another block of equal number, (2) exchange one block with another, (3) clear a selected block, (4) find, indicate, and clear all registers in an error condition, and, (5) pack all data within a given partition in continuous manner, in the same original order, from the highest register allowed by the partition on down.

William H. Beebe, Lilburn, GA
193 Steps

918221G Chase

You are in a high-voltage prison maze. There are five security robots trying to intercept and destroy you. Your only chance for survival is to make them hit the maze or each other. If you cannot destroy them all, one of them will get you.

David Kantrowitz, Brookline, MA
560 Steps, PC-100A

918236G Cluo Too

Finally, a solitaire game like Clue. This one is strictly for Clue lovers - or potential lovers. There are eight rooms, six suspects and six weapons. You are one of four players. Each make suggestions, and get responses from the other players, but only you can make an accusation.

David S. Lane, Clearwater, FL
479 Steps, PC-100A

918245G Misadventure

No dice rolling or card dealing or other randomness in this game. It's strictly deterministic - solved solely by your brilliant, logical mind. You start at the mouth of a tunnel, flash light in hand. There is gold at the end of the tunnel, but before you get there, there are many impediments with their associated mollifications.

David S. Lane, Clearwater, FL
544 Steps, PC-100A

928036G Line Equation Problem Generator

Prints an easy-to-read list of pairs of random points in a coordinate plane. For each pair it prints two forms of the equation for the line containing those points: the slope-intercept form and the $Ax + By = C$ form. All fractions and equations are in reduced form.

David Kantrowitz, Brookline, MA
409 Steps, PC-100A, Mod 1

928037G Quadratic Factoring Problem Generator

Generates an easy-to-read printed list of random quadratic polynomials that can be factored over the integers and their factorizations.

David Kantrowitz, Brookline, MA
322 Steps, PC-100A, Mod 1

958019G Multiple Flash Exposure

Will determine the exposure for multiple flash set-ups of any number of bulbs, of any size, at any distance and in any position.

Edwin L. Bond, Baltimore, MD
854 Steps, PC-100A

968012G Music Key Signature

If user inputs a major (minor) key name, the following outputs may be obtained: number of sharps or flats, name of relative minor (major) key. If user inputs number of sharps or flats, outputs the name of the corresponding major or minor key. An input key name is initially converted to the appropriate enharmonic equivalent, and the equivalent is displayed.

Barbara C. Hevener, Columbia, SC
615 Steps