PPX EXCHANGE

Vol. 1 Number 6 Copyright 1977 November 1977

WELCOME PPX-59! As mentioned in previous newsletters, PPX-52 and PPX-59 will share this newsletter to the advantage of both of these software exchanges. It is hoped that the infusion of new people and ideas into the PPX Exc change will stimulate exploration of different and unique programming applications. The PPX Exc hange will continue featuring articles of specific application to the SR-52, with the addition of articles concerning the TI-59. These articles are bound to be of interest to the owners of both machines and in many cases they will be applicable to both. So, with these thoughts in mind, we again welcome PPX-59 and pledge our continued support to PPX-52 and to the goal of an efficient idea Exc hange.

PPX POTPOURRI

1. Due to PPX's recent change of address, there has been confusion as to where to send preprinted order forms bearing the old Dallas address. (Hopefully, these order forms are no longer in the system.) If an order form has the Dallas address printed on it, it will be forwarded to PPX. If there is no preprinted address to contend with, whether it be money or correspondence, send it to:

TEXAS INSTRUMENTS PPX
P. O. Box 53
Lubbock, Texas 79408

2. On the subject of addresses — If you move, please let PPX know your new address. A lot of mail is being returned to PPX as undeliverable due to lack of present addresses.
3. Due to the administrative difficulties related to handling memberships, orders, and program contributions, it is PPX's policy not to intermingle the order activities of PPX-52 and PPX-59. For this reason, PPX is unable to transfer PPX-52 magnetic cards/program credits to PPX-59. The number of PPX-52 members involved in such transactions makes intermingling prohibitive.
4. How time flies! Membership renewal time is just around the corner for many PPX-52 members. To make certain that these members have a full year of PPX benefits, memberships originated prior to January 1, 1977 will not expire until January 1, 1978. PPX will be informing everyone of their membership anniversary dates in ample time for them to rejoin PPX-52 for their second year.

LOTS OF USER DEFINED KEYS

David M. Brender

Editors Note: The following article discusses User Defined Keys on the SR-52; but it also applies to the TI-59.

On the SR-52, instead of using the keys A through E individually to directly access only five subroutines, they can be used in pairs (such as AA, AB, BA, . . ., EE) with each pair providing direct access to a distinct subroutine. These pairs perform as the equivalent of 25 user defined keys (each labeling a distinct subroutine). Within each user defined label, flags are tested/set and branching is performed to addressable locations (where specific routines are executed). For rapid execution, all branching logic should be placed at the beginning of the program and all "perform" logic placed towards the end.

One flag is assigned to each key. The subroutine for each individual key first tests each of the five flags in turn, and if none are on, turns on its own assigned flag and returns (but can, if desired, execute a preliminary procedure before returning). Pressing the next individual key will result in branching to the desired subroutine on one of the five tests. Each of the twenty-five possible subroutines must have its own distinct LBL or absolute address, and terminate with *rset (000 HLT). For example, key B executes branching to the programs associated with pairs AB, BB, CB, DB, and EB.

Using 4 flags and 4 keys, there are 16 possible pairs (4x4). Using 4 flags and 5 keys allows 20 possible pairs (4x5). If a key doesn't have its own assigned flag, it cannot be the first key (on the left side) of a pair, since it cannot be recorded for later testing. An unflagged key can also be used individually for normal program access.

The A' through E' keys can also be utilized, but since the SR-52 has only five flags, it can only be used as the second key (on the right hand side) in pairs with A through E. (If the prime keys were used they would add 25 more pairs; however, 50 pairs are more pairs than can be used on the SR-52). On the TI-59 with its 10 flags, 10 keys (including prime keys) yield 100 pairs.

To calculate program overhead, note that each flag requires two steps (to set it), each pair requires three steps (to test and branch), each individual key, A through E, requires three steps (to label and return), and there is one step needed at location 000 (to halt upon resetting). The average overhead required to use these pairs varies from four program steps per pair (101 steps for 25 pairs on 5 keys) to six steps per pair (23 steps for 4 pairs on 2 keys). (Using absolute addresses is the best way to reduce overhead.)

The combinations A RUN through E RUN can be used to add five more subroutines without additional overhead involved. Replace the *rtns of the A through E programs with HLT, followed by the new programs (which end with *rset). Notice that the use of these RUN pairs reduces **average** overhead because they don't require a label, testing/setting of flags, or branching.

By further stretching your ingenuity, more subroutines can be accessed. However, you reach a point of no return, as you only have room for branching logic and none for "perform" logic.

CALCULATOR DOCTOR

This column is intended to answer frequently occurring questions relating to either SR-52 or TI-59 operation and programming. These questions are obtained from TI's Consumer Relations Department. If you are having difficulty with your calculator, contact TI's Consumer Relations Department for assistance. (This month, there are no problems concerning the SR-52, therefore this column will be devoted to TI-59 questions.)

QUESTION: I am confused about how to read the magnetic cards with my TI-59. The Personal Programming manual seems to imply that the sequence INV *Write is required, but the salesman who demonstrated the calculator to me just pushed in the card. What is the correct procedure?

ANSWER: TI has received many inquiries concerning reading cards with the TI-59; hopefully this one answer will cover most the questions. First of all, the sequence INV *Write is required only when setting up a card-read operation **within a program.** That is, the program, itself, sets up the calculator to read a card.

When loading a card manually, three options (none of which involve INV *Write) are available. First, if a zero is in the display when the card is inserted, any bank may be read (provided that the partition of the calculator and recorded program are the same). The calculator will put the information where it belongs. (The number in the display represents the bank that the information is loaded on.)

Second, if an integer N ($1 \leq N \leq 4$) is in the display at the time of card reading, the calculator will read the card only if the bank number recorded on the card side being read agrees with the number in the display. If it does not, the calculator will flash the bank number that was on the card. If partitioning is incorrect, the card is not read and the number of the entered bank is flashed in the display.

Third, if a negative integer N ($-4 \leq N \leq -1$) is in the display when the card is inserted, the calculator will load the contents of the read card side into bank N regardless of the bank number recorded on the card and regardless of whether the current partitioning of the calculator agrees with the partitioning information recorded on the card. This disregarding of partitioning can result in some rather unexpected results (what was supposed to be data ends up as program steps).

I have found the first method (pressing CLR and inserting the card) the easiest to use, but the other two methods are available and useful under certain circumstances as described in your Personal Programming manual.

QUESTION: How can I get my TI-59 to "fracture" its display?

ANSWER: As far as I know, it can't be done. "Fracturing" the display on the SR-52 involves overwriting or masking data in pending operation registers. Those registers aren't accessible on the TI-59 in the same way.

COMMENT: Some complaints have been received concerning both the failure of some TI-59s to write out onto a card, and the inability of some machines to read a card written by another TI-59.

• Many of the TI-59 write failures reported are caused by operator error. Often the dummy label cards (diecut on the plastic perforated sheets) packed with the Master Library are mistaken for magnetic cards. Note that **only** those cards supplied in the ziplock bag (received with your calculator) are programmable.

• Cards written on one TI-59 may not read on another machine. This incompatibility is attributed to the difference in machine signatures, which is a result of increased magnetic card density. (The density of the TI-59 card is twice that of the SR-52.)

THE CHALLENGE AND EXPLOITATION OF OPTIMIZING

J. E. Eller

Editors Note: The following article deals with optimizing on the SR-52; but it also applies to the TI-59.

Many words have been written in description and praise of programmable calculators in general, and the SR-52 in particular. However, all too often, there appears a tendency to ignore, and sometimes even to downgrade, the concept of optimizing programs in such calculators. T.I., in the SR-52 owner's manual, notes: "Condensing . . . is a time consuming exercise. If a program is less than 224 steps and operates properly, any time spent to condense the program is virtually wasted . . .". My experience has proven otherwise. Perhaps the key lies in the degree of utility and life of a program, and in the semantics of the phrase "operates properly". If the life of a program is brief, or its use infrequent, excessive time spent in optimizing a "working" program may be unwise. However, for most programs, it is both possible and useful to add one or more features to those originally conceived and initially incorporated. In my own experience, at least 60-70% of all programs I have obtained, (from TI Libraries, PPX, or other sources) have proven candidates for extended usefulness.

Before proceeding further, it may be appropriate to categorize the word "optimizing". Properly, it has several meanings, e.g. decreasing program running time. This article addresses itself only to the narrow definition of gaining program steps. Another point, albeit somewhat obvious, should be noted — in all cases, the required effort and time needed to optimize increases exponentially as optimizing steps are applied. However, even when the program appears to be "bled dry" (and the programmer pleasantly exhausted), there is usually at least one more possibility!

Now, let us move on to the procedures of optimizing. It is often said that programming is a very personal operation — that each programmer will approach a problem differently. This is indisputably true of programming in general, and is eminently more accurate in describing optimizing. Thus, it would be impractical to provide detailed procedures to apply to such an individualized operation. We must relate to ideas rather than to steps. A number of excellent ideas are offered in the SR-52 "Programming Workbook" published by Texas Instruments. This publication is highly recommended, so no attempt will be made to duplicate it here. One of the best ways to gain ideas for possible future use in your programming or optimizing is to carefully consider routines printed in the PPX Exchange, as well as routines used in programs which you obtain from other authors.

To evaluate if optimization is needed, I begin by considering the program listing of each new acquisition for the more obvious possibilities. Examples of these are: duplicated routines as candidates for subroutines, unnecessary steps, the use of vacant user defined labels for sub-

routines in lieu of other labels, etc. This step, with a little practice, is usually the simplest. Note that "duplicated routines" may be an entire procedure or only a memory operation on a single register which occurs a number of times. Next, consider whether any required operations can be performed with fewer steps. As an example, consider the requirement for a number which continually varies "randomly" between 0.5 and 1.2. There are many routines for developing "random" numbers, but, with some thought, even a requirement as complex as this one can be accomplished in only 26 steps (perhaps you can do it in less?) as follows:

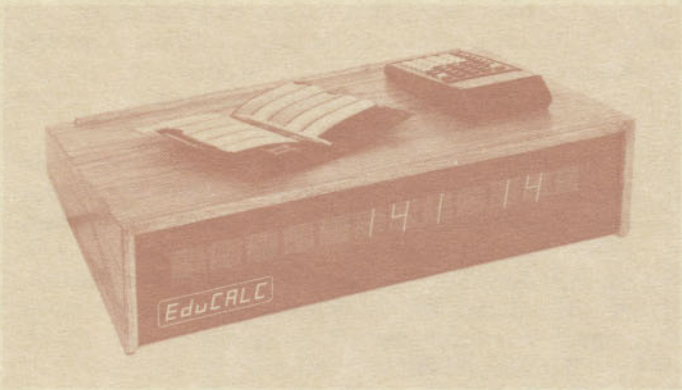
(*π x RCL 1 9 - *fix 0 EE INV EE + *if pos
*9 . 7 + *LBL *9 . 5) STO 1 9

(This sequence can not stand by itself, you need to use this in your program, as a multiplier or otherwise.) Next, consider whether use can be made of one of the operating characteristics of the calculator. For example, suppose you have reason to count the number of times the random factor is applied. In the above sequence, you may insert between *π and x the following: 1 SUM 6 7. These steps will not effect the multiplication factor still in register 19.

By this point, you should be fairly well aware of the calculator operating features you can use to your advantage. It may be appropriate to begin writing possible routines using them, and see where they can be inserted. When all other steps have been applied, you may wish to prepare a flow diagram for the program as it now stands. This could point out possible program segments which are candidates for rearrangement. At this time, it is usually best to set aside this particular effort, to be returned to in a few hours or days. For then, to your surprise — Behold, there is one more possibility!

By now, you may be wondering why you should expend this effort on a program you have already written, bought, or traded. There are at least three reasons, applying to individual readers in varying orders of importance. Consider these:

1. The extended usefulness of the program, through added features,
2. The personal satisfaction gained from meeting the challenge, and
3. Perhaps most lasting of all, the increased understanding of, and facility with, your prized possession, the SR-52. For I know of no better way to gain proficiency in programming than by optimizing!



PROGRAMMABLES FOR TEACHERS

Do you teach or lecture using your TI Programmable? If so, the EduCALC™ may be just the tool for you. EduCALC™ is manufactured by Educational Calculator Devices, Inc. using a Texas Instruments calculator. It is a new Audio-Visual tool: a teacher's calculator with two displays.

Each unit is an oaken lectern a little larger than a portable typewriter. It has a "master" TI-calculator permanently mounted on top. The calculator is operated in the usual manner, reading its red numerical display for results. But, there is also a much larger display panel facing toward the audience. The display panel, consisting of large digits, displays the same numbers as the TI calculator. This "slave" display has a viewing angle of 130° and is legible at 60 feet (18 meters).

The EduCALC™ is designed for classroom instruction. It enables presentations without the instructor pausing to read off displayed numbers or write numbers on the blackboard. The class sees all the entries and results as they happen allowing the class to share in the excitement and power of TI Programmables. The instructor can demonstrate either a program previously written or originate a program in front of the class.

EduCALC™ models incorporating the TI Programmable 58 or 59 as "master" calculators are available as well as compact units which utilize the SR-40 or the TI Business Analyst. The Programmable models are supplied with a rigid carrying case. All models carry a one-year warranty.

Texas Instruments does not manufacture or sell EduCALC™s, nor does TI warrant or guarantee them in any way. This article is meant only to inform PPX members of the availability of these machines. For further information, please direct all inquiries to the manufacturer:

Educational Calculator Devices, Inc.
P. O. Box 974, Dept. PPX
Laguna Beach, California 92652

Editors Note: This article was presented to acquaint you with one of the newest tools used in teaching programming. In January's issue, PPX Exchange will describe Texas Instruments newly developed programming course. Using the TI-57, this course provides basic computer literacy to students, starting at the Junior High School level.

NAVIGATION IS A "CINCH" — WITH A PROGRAMMABLE

E. S. Maloney

Navigation is a "natural" application for a programmable calculator as essentially all processes are mathematical. The TI-produced libraries and software modules, and the programs written by users are only the beginning. Boating magazines have carried many articles on the use of calculators for navigation and related fields, and an increasing number of books are appearing on this subject.

The owner of a programmable should not, however, feel dependent upon published programs. Navigational equations are not overly complex and a satisfactory program can be developed by anyone to meet the needs of a particular situation. Calculations can be made for fuel consumption, propeller slip, racing handicaps, and many other boating functions, as well as those directly related to piloting and celestial navigation.

There are three steps to be taken towards finding the solution to a navigational problem using a programmable. First, check the Software Catalog for an already-existing program that can be used directly or modified to meet the

SR-52/TI-59 CONVERSION

Technology always seems to be one step ahead of us. While any new advance in industry is exciting, it can also be very perplexing to the consumer. The advent of the TI-59, Texas Instrument's most recent programmable calculator, has caused quite a stir among present owners of the SR-52. After all, the SR-52 is quite a machine and many a user has spent hour upon hour writing SR-52 programs that best suit his needs. The most prevalent question that arises is, "If I buy a TI-59, what will I do with all the SR-52 software into which I have invested my time and money?" Well, do not despair! The TI-59 uses the same AOS logic as the SR-52 and therefore the SR-52/TI-59 user is capable of upgrading his present library of SR-52 programs to run on the TI-59. The process of converting an SR-52 program is relatively easy (even though it is not always a one-to-one relationship). The following chart contains SR-52 keystrokes and their TI-59 equivalents (registers available with the instruction are shown on the outer margins).

register(s)	SR-52 keystrokes	TI-59 keystrokes	register(s)
	$^{x}\sqrt{y}$	INV y^{x}	
	*rtn	INV SBR	
	HLT; RUN	R/S	
00	*dsz	① *Dsz	0-9
	*x!	② *Pgm 16	
00	*P/R	③ *P→R	t
	*if pos	④ x≥t	t
	INV *if pos	④ INV x≥t	t
	*if zro	④ x=t	t
	INV *if zro	④ INV x=t	t
any accessible	1 SUM *(2 digit register)*	⑤ *Op 2 *(1 digit register)*	0-9
any accessible	1 +/− SUM *(2 digit register)*	⑤ *Op 3 *(1 digit register)*	0-9
	*if err	⑥ *Op 19	
	INV *if err	⑥ *Op 18	
	*IND *RCL*	⑦ *RCL* *Ind	

NOTES:

① There are up to ninety-nine nested loops possible using the *Dsz key on the TI-59. The TI-59 manual discusses ten registers, 0-9. The format of the *Dsz instruction is *Dsz nn followed by a transfer address where nn is the register to be decremented and tested. Actually with a little more effort, the other 89 registers (excluding register 40 which implies indirect) can be used with the *Dsz key. To take advantage of the additional 89 registers, the nn can be generated by the following sequence: LRN, *Dsz, STO, n, n, BST, BST, *Del, SST, --- LRN. Only two BST are needed because the 2 digit register number is merged and occupies one location on the TI-59.

② *Pgm 16 is one of the many TI-59 subroutines within the Master Library Module. Module subroutines can replace portions of an SR-52 program.

③ On the SR-52, register 00 is used when making polar/rectangular conversions. The TI-59 uses the t register, a data memory register where values can be stored, recalled, and compared against the display register.

④ When performing comparisons on the SR-52, the display register is compared to zero, whereas, the t register is used when performing comparisons on the TI-59. When *x ≥ t, INV *x ≥ t, *x=t, or INV *x=t is pressed, the TI-59 compares the contents of the display register to the value in the t register.

⑤ The TI-59 has a series of operations that are accessed by the *Op key. It is comparable to the 2nd key in that it provides more functions to the user. Only those *Op codes which replace an SR-52 keystroke sequence are shown here. (See page V-27 in the Personal Programming manual for additional *Op codes.)

⑥ Same as 5 above. *Op 18 and 19 use flag 7 for error checking. (See page V-67 in the Personal Programming manual for further explanation.)

⑦ Leading to more efficient programming, the indirect address capabilities have been increased on the TI-59. The TI-59 has twenty-seven indirect instructions compared to the SR-52's seven instructions. (See page V-68 in the Personal Programming manual for a list of indirect instructions.)

Once the art of the SR-52/TI-59 conversion is mastered, it becomes apparent that the programs written by SR-52 users are not to be discarded. Rather, excellent programming techniques are uncovered while converting a fellow SR-52 user's program. By combining these techniques with the added capacity of the TI-59, larger, more complex problems can be solved and your programming job will be made easier.

PPX PRESENTS YOU WITH A SPECIAL THANKSGIVING TREAT... AND THAT'S NO JIVE!

Jive Turkey

This game resembles the classic game of Hi-Lo; but only up to a point. This program sometimes "jives" you.

To play the game, you try to guess a number that the program has generated. It then tells you whether your guess is high, low, or correct. However, at the beginning of the game you have to enter a "probability of truth", a number between 0 to 100. This entry determines how often the program "jives" you. For example, if you enter 75, the program gives you correct answers 75% of the time and incorrect answers 25% of the time.

PPX wishes to thank the author of the original SR-52 "Jive Turkey", Maurice E. T. Swinnen, for his excellent program.

INSTRUCTIONS (Display and Printer)

1. For TI-59 ONLY press [SBR] [CLR] to initialize program; 0. is displayed. For SR-52 ONLY press [GTO] [CLR] [RUN] to initialize program; 0. is displayed.
2. Enter the "probability of truth" as a number between 0 and 100 and press [A].
3. Enter a seed number, 0 to 199017, and press [B]. The program then generates an integer between 0 and 100.
4. Enter your guess (0 to 100) and press [E]. If your guess is too high, 1 is displayed; too low, -1 is displayed. If your guess is correct, the correct number is flashed in the display. (Your guess and the program's answer are printed.)
5. Press [D] to determine how many guesses you have taken. (First press [CE] if display is flashing.)
6. To play another game, go to Step 1.

EXAMPLE: Play Jive Turkey using 956 as the seed number and the probability of truth equal to 80.

Enter	Press	Display	Comments
(SR-52 ONLY)	[GTO](CLR](RUN]	0.	Initialize
(TI-59 ONLY)	[SBR] [CLR]	0.	Initialize
80†	[A]	80.	Probability of truth
956†	[B]	0.	Seed
50†	[E]	1.†	High
25†	[E]	1.†	High
15†	[E]	-1.†	Low
20†	[E]	1.(-1)†	High (Low, SR-52 ONLY)
17†	[E]	-1.†	Low
18†	[E]	-1.†	Low
19†	[E]	1.(-1)†	High ? (Low, SR-52 ONLY)
19†	[E]	-1.†	Low
19†	[E]	-1.(1)†	Low (High, SR-52 ONLY)
22†	[E]	"22."†	Correct (flashing display)
	[CE] [D]	10.†	Guesses

†These values are printed if the PC-100A is connected.
(Due to hardware differences in the calculators, the SR-52 and TI-59 "jive" at different times but have the same probability of truth.)

SR-52 LISTING*

000	46	LBL	041	56	RTN	082	95	=	123	33	COS	164	09	9
001	48	EXC	042	46	LBL	083	80	IF+	124	43	RCL	165	09	9
002	75	-	043	25	CLR	084	42	STO	125	00	0	166	01	1
003	93	.	044	57	FIX	085	50	STF	126	02	2	167	54	)
004	05	5	045	09	9	086	01	1	127	94	+/-	168	55	÷
005	54	)	046	00	0	087	46	LBL	128	30	ΓX	169	01	1
006	57	FIX	047	42	STO	088	42	STO	129	40	X²	170	09	9
007	00	0	048	00	0	089	43	RCL	130	98	PRT	171	09	9
008	52	EE	049	05	5	090	00	0	131	99	PAP	172	00	0
009	22	INV	050	56	RTN	091	02	2	132	56	RTN	173	01	1
010	52	EE	051	46	LBL	092	75	-	133	46	LBL	174	07	7
011	22	INV	052	15	E	093	43	RCL	134	13	C	175	42	STO
012	57	FIX	053	22	INV	094	00	0	135	51	SBR	176	01	1
013	56	RTN	054	50	STF	095	01	1	136	37	DMS	177	00	0
014	46	LBL	055	01	1	096	95	=	137	98	PRT	178	54	)
015	11	A	056	42	STO	097	90	IF2	138	56	RTN	179	42	STO
016	42	STO	057	00	0	098	33	COS	139	46	LBL	180	01	1
017	00	0	058	02	2	099	80	IF+	140	14	D	181	01	1
018	04	4	059	98	PRT	100	34	TAN	141	43	RCL	182	51	SBR
019	98	PRT	060	01	1	101	60	IFF	142	00	0	183	48	EXC
020	56	RTN	061	44	SUM	102	01	1	143	05	5	184	22	INV
021	46	LBL	062	00	0	103	30	ΓX	144	98	PRT	185	44	SUM
022	12	B	063	05	5	104	46	LBL	145	56	RTN	186	01	1
023	42	STO	064	51	SBR	105	40	X²	146	46	LBL	187	01	1
024	00	0	065	37	DMS	106	01	1	147	37	DMS	188	43	RCL
025	09	9	066	53	(	107	98	PRT	148	57	FIX	189	01	1
026	98	PRT	067	24	CE	108	99	PAP	149	09	9	190	01	1
027	53	(	068	65	×	109	56	RTN	150	53	(	191	65	×
028	51	SBR	069	01	1	110	46	LBL	151	53	(	192	43	RCL
029	37	DMS	070	00	0	111	34	TAN	152	02	2	193	01	1
030	65	×	071	00	0	112	60	IFF	153	04	4	194	00	0
031	01	1	072	54	)	113	01	1	154	02	2	195	54	)
032	00	0	073	51	SBR	114	40	X²	155	09	9	196	42	STO
033	00	0	074	48	EXC	115	46	LBL	156	08	8	197	00	0
034	54	)	075	42	STO	116	30	ΓX	157	65	×	198	09	9
035	51	SBR	076	00	0	117	01	1	158	43	RCL	199	55	÷
036	48	EXC	077	03	3	118	94	+/-	159	00	0	200	43	RCL
037	42	STO	078	75	-	119	98	PRT	160	09	9	201	01	1
038	00	0	079	43	RCL	120	99	PAP	161	85	+	202	00	0
039	01	1	080	00	0	121	56	RTN	162	09	9	203	54	)
040	00	0	081	04	4	122	46	LBL	163	09	9	204	56	RTN

TI-59 LISTING

000	76	LBL	041	25	CLR	082	92	RTN	123	02	2			
001	11	A	042	71	SBR	083	76	LBL	124	04	4			
002	42	STO	043	88	DMS	084	30	TAN	125	02	2			
003	04	04	044	53	(	085	87	IFF	126	09	9			
004	99	PRT	045	24	CE	086	01	01	127	08	8			
005	92	RTN	046	65	×	087	33	X²	128	65	×			
006	76	LBL	047	01	1	088	76	LBL	129	43	RCL			
007	12	B	048	00	0	089	34	ΓX,	130	09	09			
008	42	STO	049	00	0	090	01	1	131	85	+			
009	09	09	050	54	)	091	94	+/-	132	09	9			
010	99	PRT	051	59	INT	092	99	PRT	133	09	9			
011	53	(	052	42	STO	093	98	ADV	134	09	9			
012	71	SBR	053	03	03	094	92	RTN	135	09	9			
013	88	DMS	054	32	X:T	095	76	LBL	136	01	1			
014	65	×	055	43	RCL	096	39	COS	137	54	)			
015	01	1	056	04	04	097	43	RCL	138	55	÷			
016	00	0	057	22	INV	098	02	02	139	01	1			
017	00	0	058	77	GE	099	94	+/-	140	09	9			
018	54	)	059	42	STO	100	34	ΓX	141	09	9			
019	59	INT	060	86	STF	101	33	X²	142	00	0			
020	42	STO	061	01	01	102	99	PRT	143	01	1			
021	01	01	062	76	LBL	103	98	ADV	144	07	7			
022	00	0	063	42	STO	104	92	RTN	145	42	STO			
023	92	RTN	064	43	RCL	105	76	LBL	146	10	10			
024	76	LBL	065	02	02	106	13	C	147	54	)			
025	25	CLR	066	32	X:T	107	71	SBR	148	53	(			
026	58	FIX	067	43	RCL	108	88	DMS	149	53	(			
027	09	09	068	01	01	109	99	PRT	150	22	INV			
028	00	0	069	67	EQ	110	92	RTN	151	59	INT			
029	42	STO	070	39	COS	111	76	LBL	152	65	×			
030	05	05	071	22	INV	112	14	D	153	43	RCL			
031	92	RTN	072	77	GE	113	43	RCL	154	10	10			
032	76	LBL	073	30	TAN	114	05	05	155	54	)			
033	15	E	074	87	IFF	115	99	PRT	156	42	STO			
034	22	INV	075	01	01	116	92	RTN	157	09	09			
035	86	STF	076	34	ΓX	117	76	LBL	158	55	÷			
036	01	01	077	76	LBL	118	88	DMS	159	43	RCL			
037	42	STO	078	33	X²	119	58	FIX	160	10	10			
038	02	02	079	01	1	120	09	09	161	54	)			
039	99	PRT	080	99	PRT	121	53	(	162	92	RTN			
040	69	OP	081	98	ADV	122	53	(						

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

problem at hand. Use the Keyword Index — it's a great help — and check each title in categories 93 and 94. There is also the possibility of a program in the Mathematics section that could be adapted. If a program can not be found, the second step is to search navigational literature for an equation (or equations) that expresses the relationship between the known and unknown factors of your problem. Of the many books on navigation, it is most likely that you will find what you are seeking in "Bowditch" (Defense Mapping Agency Hydrographic Center Publication No. 9) or "Dutton's Navigation and Piloting" (published by the U. S. Naval Institute).

If an equation can not be found, the third step is to study the literature for any discussions of the relationships between the variables involved, and then develop your own equation(s). Look for clues in published books and articles.

For example, let us follow my development of a program that enables a navigator to directly calculate the height of tide at any given time (taken, that not a program was listed or known to be available, and that an equation could not be found in "Bowditch" or "Dutton's"). I searched for clues in other books and articles. Finally, in an explanation of Table 3 in the "Tide Tables" (published by the National Ocean Survey), a clue was found. This clue revealed that a cosine relationship had been assumed by NOS in the preparation of Table 3.

The predicted height of tide at any time is the predicted height at low water plus a **correction factor** that varies with time. Upon study, this factor was actually found to vary in a **haversine** relationship [hav $\theta = \frac{1}{2}(1-\cos\theta)$]. The time of low water preceding the desired time is taken as 0° and the subsequent high water as 180°. The interval between low water and the desired time is divided by the interval between low water and high water; the resulting fraction is multipled by 180° to determine θ. The correction factor obtained from the haversine equation is multiplied by the range of the tide (high water height - low water height) to get the correction amount, which is added to the low water height to get the height at the desired time.

These relationships were programmed; "Height of Tide at Given Time" (PPX-52 #940001). Now, the predicted height of tide at either a **primary reference station** or a **subordinate station** is more readily available. When you can get away from tables, Navigation is a cinch!

FROM THE ANALYST'S DESK

• PPX-52 would like to apologize to Dr. Pierre Brind 'Amour for the misspelling of "calendar" in his article "A Look Into Yesterday" (September, PPX Exc hange). We should have known better than to trust our Texas Dictionary!

• PPX-52 would like to extend an apology to Mr. Willard B. Johnson for not including his program, "Two Track Cribbage Board" (PPX-52 #910108) in the C Addendum to the PPX-52 Catalog. This program serves as a two track 121 point cribbage board. It keeps track of each team's highest hand scored, number of points scored in the game, and number of corners, games, and skunks won. This program is **currently** available.

• For those PPX-59 members using PC-100As, we would like to offer the following suggestion: We have found that Scotch® Spra Mount is an excellent adhesive for pasting up program listings submitted to PPX-59. (If you choose to use tape instead, do not tape over any printing as that printing will quickly fade.)

• For those members who have purchased "Prime Number Generator" (PPX-52 #390048) by Thomas N. Ferguson, the program steps 040-049 should read as follows:

*if zro *2 RCL 0 2 *x² - RCL 0 1

• PPX member, Mr. Serge Borodin has found his SR-52 to be quite useful in his professional field. Mr. Borodin works for Consolidated Edison Company of New York as a systems equipment studies electrical engineer. Upon recently experiencing the blackout in New York City, Mr. Borodin has found that a calculator in hand is more useful than a room full of computers with their life line cut off — electricity. The complexities involved in analyzing and diagnosing electrical power failures are clearly shown in Mr. Borodin's programs: "H.V.A.C. Circuit Breaker Short Time Ratings" (PPX-52 #640047), "Minimum Total Emergency Capability of Circuits" (PPX-52 #640049) and "Power Transformer Emergency Overloads" (PPX-52 #640055). The SR-52 adds an important means of immediate investigation into the effect of varying the input variables associated with an electric power failure.

• The publication of the PPX-59 November Software Catalog brought many useful and interesting programs across the analysts' desk. The "TI-59 Banner Program" (PPX-59 #908015) by Mark K. House of Dallas, Texas was among these. This program will print, using a matrix block form, the letters of the alphabet as well as many special characters. PPX has used this program numerous times to make functional, as well as humorous, banners. Below is a reduced example of the "TI-59 Banner Program" output.

• Mr. George A. Bowman uses his SR-52 to aid him in his job as a microwave engineer. Instead of retreating to expensive large computers, Mr. Bowman has tackled the problem of microwave filter design with his programmable. His program, "Microwave Interdigital Filter" (PPX-52 #650024), incorporates 9 important equations (into a 2 card, 415 step program), dealing with the design of a Tchebyshev interdigital filter. By using this program, one can learn a great deal about the physical construction of this filter as it relates to electrical tolerances, physical packaging and implementary adjustments.

The PPX Exc hange 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