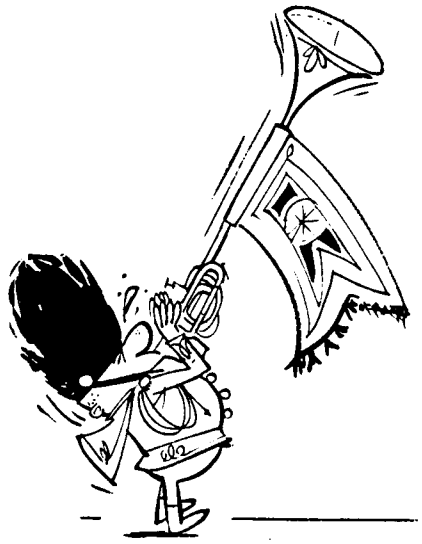
* TI PPC NOTES *
* *
****

NEWSLETTER OF THE
TI PERSONAL PROGRAMMABLE CALCULATOR
CLUB

9213 Lanham Severn Road
Lanham MD 20706 USA.



EXTRA– Read all About it!

The big news, of course, is the new TI-88. On page 13 you will find the very first program I wrote on the new machine: JIVE TURKEY !

It had to be Jive Turkey, of course. No other program would do. Even that program is hated in some circles, it still occupies a very dear spot in my heart.

After waiting about four weeks for my first 88 to arrive (they are rarer than hen's teeth, you know) I finally was able to play with it for about one week before starting to write the final pages of this issue. It is everything I expected, and more.

You will also find some samples of the Master Library module. The Random Number Generator works fine this time and so does the last program in the module, called Function Evaluator.

Besides the 12 programs of which you will find a list on page 25, there are three more programs, which are normally only called as subroutines by one of the main Master Library programs.

Plotting capability is minimal. The Function Evaluator program has some capability which reminds one vaguely of the OP 07 in the 59. But this one has even less width: only 12 asterisks. We will have to remedy that situation soon by writing a good plotting program. Maybe one that uses individual dots and so will have a resulution of 5 X 16 = 80 dots across the width of the paper.

And if you think the 59 was flexible with respect to comparisons, this one is even more so: after a comparison (which can be done with ANY register) you don't have to branch if you don't want to. The calculator will either execute the command following the comparison (if that one is true) or jump the first command. (if the comparison is false) Very handy!

We have our very first CHESS program for the 59. Michael Sperber (Fast Mode inventor) did it!

I also recommend highly Patrick Acosta's HEX-KEYCODE CREATOR on page 18. The 59 is not dead, not by a long shot! Maurice.

IN THIS ISSUE:

Dear Maurice,

The latest PPX newsletter contains an interesting TI-59 users test attributed to you. I am sorry to say, TI forgot to give us question 42. Even an expert will tell you that I have only a 20% change of getting that one right. That "indirect SBR" was a Jim Dandy! No wonder they said "please don't try to decipher the program. Just punch it in." Question # 18: You're tricky one, Swinnen! On question #82: you are wrong and on question # 83: I don't think any of the answers are right. On question #61: DEL also works, etc. etc. (this is a composite of many letters received on this subject.)

Greetings,
A member.

Dear Member,

The original TI-59 test was written by me. It was usually administered to students who registered for a five-day programming course. It was given "before" and "after" to show the student "how little he knew" and "how much he had learned in those five days!Psychology, you see! But then the brain trust Worthington and Regelman got hold of it one day and rewrote it to the bone. They even included the program with it. I plan to ask them to justify their actions in a long article in the NOTES, if I ever get hold of them. One moved out of my region, to a place called Burks Eagle Nest or something that sounds like it, way out in the state of Virginia and the other one, even though he still lives almost next door in Bowie, Maryland, has been busy taking a wordprocessor program (in machine language, mind you) apart and put it back together again. Compared to that, I am afraid the 59 doesn't offer much of a challlenge.

Dear Maurice,

I enjoy LRN and hope you're sufficiently encouraged to keep on editing it.

My TI-59 recently failed to recall stored data; yet INV LIST would print it. I have been reading all your "take-it-apart" notes and was about to when something told me to scrub all contacts with a pencil eraser. This fixed it.

Very Truly Yours,
R. H. M. South Bend IN.

Dear Rawson,

Thanks for the tip. I suppose with "all the contacts" you mean the contacts between the calculator and the printer and not the key contacts themselves. Yes, a pencil eraser is well known by technicians and engineers as the best "de-oxidizer around.

Dear Maurice,

You may be interested in the article "The ABC's of Simplifying Logic Equations Simply" in the July 1982 issue of Computer Design, pp 99-102. It lists a TI-59 program which solves complex logic equations arithmetically. To get a copy, which costs $ 1.00, write to Computer Design, 11 Goldsmith Street, Littleton, MA, 01460.

M.G.Jamaica, NY

Thanks, Morris.

The TI-88 versus the TI-59. The difference between those two calculators is what most of us, TI-59 users, are concerned with: "What new tricks will we have to learn to program this new one with any degree of success?" There are quite a few differences, as I found out. Some of them will come as a pleasant surprise and will make you complain "why couldn't they make THAT one available for the TI-59?" Well, technology advances all the time, friends. Maybe that particular feature, even if it was known at the time the 59 was designed, could not be incorporated because it was not economically feasible.

The first thing that strikes you when you finally get your hands on a TI-88 is the extreme ease with which to enter alpha. You just press a key marked ALPHA and sito presto you transform the whole machine into a (maybe primitive) typewriter. No more look up tables to enter print code. If you now press A, you will get an A, in the display and printed if you so desire. Press ALPHA again and back comes the old calculator with its multiple functions. It is that simple. And the display will show alpha, digits or a mixture of both. And your print commands will print whatever the display shows.

Once you start programming you will notice a few more differences, such as the enhanced AOS system. Priorities of execution are based on rules of mathematics, as we all know already. But those priorities have been changed. Here is a table that shows the priorities in the TI-88 on the left and those we are familiar with in the TI-59 on the right.

TI-88

PRIORITY	FUNCTION NAME
	Reciprocal
	Factorial
First	Integer Powers
	Absolute Value
	Signum Function
	Square Root
	Common Log
	Common Antilog
	Natural Log
	Natural Antilog
	Sine
Second	Arcsine
	Cosine
	Arccosine
	Tangent
	Arctangent
	Integer
	Fraction
Third	Multiply
	Divide
Fourth	Addition
	Subtraction
Fifth	Equals

TI-58/58C/59

PRIORITY	FUNCTION NAME
	Reciprocal
	Square Root
	Square
	Absolute Value
	Signum Function
	Common Log
	Common Antilog
	Natural Log
First	Natural Antilog
	Sine
	Arcsine
	Cosine
	Arccosine
	Tangent
	Arctangent
	Integer
	Fraction
Second	Powers and Roots
Third	Multiply
	Divide
Fourth	Addition
	Subtraction
Fifth	Equals

ti88 cont1

But the way certain functions have to be entered now differs drastically with what we are accostumed to. In the 59 we enter the sine of 30 degrees as 30 SIN and the log of 2.34 as 2.34 LOG. In the 88 we have to enter them as SIN 30 = and LOG 2.24 = respectively. If you think about this critically you will have to agree that entering 30 SIN in the 59 was in reality a form of RPN, although TI will never admit to that. With this enhancement we can truely say we use an algebraic system.

Powers and roots also require a different manipulation, this time involving arrows pointing up and down. (for ease of typing I will use "up" and "dn") So, what for the 59 was y Y^x x = is entered into the 88 as y up x dn, the last arrow down telling the machine that the operation is finished. This form of exponent and root entry allows some neat tricks, such as nested powers $3^{4^{5+2}}$ which is entered as 3 up 4 up 5 + 2 dn. Roots have to be entered as exponents of less than one. For example $\sqrt[4]{5}$ has to be rewritten as $5^{\frac{1}{4}}$ and is then entered as 5 up .25 dn. Of course, you may also make good use of a special key marked $()^N$ and thus the computation might also be entered as $5\uparrow 4\downarrow ()^{-1}$. The same key is also used to enter the square of a number. Thus, 5^2 would be entered as 5 $()^N$ 2 with the possibility of entering negative bases such as -5^9 entered as

5 +/- $()^N$ 9 .

The Rectangular-to-Polar conversion in the 59 is done on angles ranging from -90 to 270 degress while in the 88 it done on angles in a range from -180 to 180 degrees. Statistical values are now accumulated in HIRs as opposed to data registers in the 59. This frees more registers for actual data accumulation or for working registers.

Calling solid-state libraries requires a little more button pushing this time. The 59 has only one I/O port and a library call is simply PGM NN. Now, with two I/O ports you will have to call PGM MMNN, in which MM is the module number and NN the program you want.

The clear function has also been changed. In the 59 CE clears the display and CLR does a little more. Now, there is only one key marked CE/C which, when pressed once performs the function of CE and pressed twice means CLR. And the CMs function can only be performed under program control. Much safer that way.

Memory addressing differs too. First of all, the numbers of registers available is now more than 99, so we have to use three instead of two digits. So, we say something like RCL 126 or STO 007. But then, to make things easy, you may address the first 26 registers as A through Z and say, for example, RCL H or EXC C.

Decision making is also slightly different: in the 59 each test has to be followed by a branch address (label or direct address). If the result of the test is true the program will branch to that address. Otherwise execution will simply continue. In the 88, on the other hand, those tests are a little more flexible. If the test is true, the calculator executes the first instruction block following the test (no branching here). If the test is false the program skips that first instruction block and continues execution. You may further chain (concatenate in computer parlance) decision making tests, each of them acting like branches of an OR-gate. This means that if ANY of the tests in the chain is true the first instruction block following the tests will be executed, if false skipped.

OP codes are different and more plentiful in the 88. The familiar OP 20 through 39 increment and decrement registers 0 through 9 are gone. Instead you will have to use STO+ and STO- which are in realty the SUM and INV SUM instructions from the 59. But other OP codes, 88 in all (many people pointed out to me that maybe the 88 was named after the number of OP codes available. I doubt it, as the prototype I saw some time ago had 89 OP codes. At that time the name TI-88 was firmly established at TI after the initial name of "Product X". It was later discovered that in reality one of the OP codes was redundant, so one was dropped.) are there to be used. Here follows a list of all OP codes and what they do. In future issues we will discuss the OP codes in more detail. By the way, if one is not sure about what a particular OP code does it is not necessary to consult the book right away. Just press INV OP NN and the definition of that OP code is displayed.

CODE	FUNCTION NAME	INV OP MESSAGE
OP 00	Define OP Codes	OP DEFINITIONS
OP 01	Set Defaults	SET DEFAULTS
OP 02	Display Calculator Settings	SHOW STATUS
OP 03	Recall Error Message	ERROR MESSAGE/a
OP 04	All Response Cue	ALL CUE
OP 05	Yes NO Cue	YES/NO CUE
OP 06	Enter/Continue Cue	ENT/CONT CUE
OP 07	Continue Cue	CONT CUE
OP 08	List Alpha Second Functions	AL ENTRY TABLE
OP 09	Recall Alpha Display Registers	RECALL ALPHA
OP 10	Right Circular Shift	→SHIFT→
OP 11	Left Circular Shift	←SHIFT←
OP 12	Show Internal Digits	SHOW 13 DIGITS
OP 13	Round Display Register	ROUND DISPLAY
OP 14	Unformatted Display Mode	UNFORMATTED MODE
OP 15	Cancel Unformatted Mode	FORMATTED MODE
OP 16	Hexadecimal Mode	HEX MODE
OP 17	Decimal Mode	DECIMAL MODE
OP 18	Display System Flag Definitions	FLAG DEFINITIONS
OP 19	Display Flags Set	SHOW FLAGS SET
OP 20	Save User Flag	SAVE FLAGS
OP 21	Exchange User Flags	EXCHANGE FLAGS
OP 22	Default Pause Timing	SET PAU TO 1.5
OP 23	Set Pause Timing	SET PAU TIMING
OP 24	Turn On Implied Multiplication	IMPLIED MULTIPLY
OP 25	Cancel Implied Multiplication	NO IMPLIED MULT
OP 26	Absolute Value	ABSOLUTE VALUE
OP 27	Signum Function	SIGNUM FUNCTION
OP 28	Convert D.MS to Decimal Degrees	D.MMSS→D.d
OP 29	Convert Decimal Degrees to D.MS	D.d→D.MMSS
OP 30	Display Angle Mode	ANGLE MODE
OP 31	Convert Degrees to Radians	D→R CONVERSION
OP 32	Convert Radians to Degrees	R→D CONVERSION
OP 33	Convert Radians to Grads	R→G CONVERSION
OP 34	Convert Grads to Radians	G→R CONVERSION
OP 35	Convert Grads to Degrees	G→D CONVERSION
OP 36	Convert Degrees to Grads	D→G CONVERSION
OP 37	Clear Statistical Registers	CLEAR STATISTICS
OP 38	Compute Y-Intercept and Slope	INTERCEPT-SLOPE
OP 39	Compute Correlation Coefficient	CORRELATION COEF
OP 40	Compute Y-Estimate	Y=Mx+b
OP 41	Compute X-Estimate	X=(Y-b):m
OP 42	Compute Statistical Means	MEANS (Y-X)
OP 43	Display Number of Data Entries	NUMBER OF POINTS
OP 44	Compute Standard Deviation (N)	N STD DEV (Y-X)
OP 45	Compute Standard Deviation (N-1)	N-1 STD DEV (Y-X)
OP 46	Set Program Counter	DISP→PGM COUNTER
OP 47	Copy Program Code to Display	PGM STEP→DISP
OP 48	Copy Program Code to Program	DISP→PGM STEP
OP 49	Set Default Partitioning	480 PGM STEPS
OP 50	Set Partitioning	SET PARTITION
OP 51	Soft Partitioning	SOFT PARTITION
OP 52	Hard Partitioning	HARD PARTITION
OP 53	List Program Labels	LIST PGM LABELS
OP 54	Calculator System Test 1	TEST 1
OP 55	Calculator System Test 2	TEST 2
OP 56	Read main memory File	TAPE→MAIN MEMORY
OP 57	Record Main Memory File	MAIN MEMORY→TAPE

OP 58	Read Program File	TAPE PGM MEMORY
OP 59	Record Program File	PGM MEMORY TAPE
OP 60	Read Data File	TAPE DATA MEMORY
OP 61	Record Data File	DATA MEMORY TAPE
OP 62	Read Numbered Module File	TAPE MODULE
OP 63	Record Numbered Module File	MODULE TAPE
OP 64	Convert Decimal to Hexadecimal	CONVERT DEC HEX
OP 65	Convert Hexadecimal to Decimal	CONVERT HEX DEC
OP 66	Test Module Number	SHOW MODULE a
OP 67	Check Module Status	MODULE STATUS
OP 68	Number Primary Port Module	NUMBER MODULE
OP 69	Erase Primary Port Module	ERASE MODULE
OP 70	Read Program from Module	MODULE→PGM MAIN
OP 71	Write Program to Module	MAIN→PGM MODULE
OP 72	Protect Module	PROTECT MODULE
OP 73	Copy Module	COPY MODULE
OP 74	24-Hour Mode	24 HOUR CLOCK
OP 75	12-Hour Mode	12 HOUR CLOCK
OP 76	Add Time Values	HH.MMSS ADD
OP 77	Subtract Time Value	HH,MMSS SUBTRACT
OP 78	Set Alarm Time	SET ALARM TIME
OP 79	Turn Alarm On	CLOCK ALARM ON
OP 80	Turn Alarm Off	CLOCK ALARM OFF
OP 81	Sound Tone	TONE
OP 82	Turn On Error Tone	TONE ON ERROR
OP 83	Turn Off Error Tone	NO TONE ON ERROR
OP 84	Turn On Cue Tone	TONE ON CUE
OP 85	Turn Off Cue Tone	NO TONE ON CUE
OP 86	Keyboard Tones	KEYBOARD TONES
OP 87	Output Data	DISPLAY→I/O
OP 88	Input Data	I/O→DISPLAY

Labels are another matter again: in the 59 we use the user-defined keys A through E and A' through E'. In the 88 we use A through J. In the 59 we can use almost any key as a common label for branching and such. In the TI-88 we are not allowed to do that. We may use instead all the labels A through Z and numeric labels 00 through 99. Two special functions are included when transfer is made to a numeric label: GTO Lbl and SBR Lbl, which both merge to Gtl and Sbl respectively.

Direct addressing is also possible in the 88, however with a 4-digit address this time. Besides that, you can also do a RELATIVE GO FORWARD and a RELATIVE GO BACKWARD with a 2-digit address. This means that you can jump forward or backward RELATIVE to your present position (program step) 1 through 99 steps. Very handy indeed when you want a quick transfer to another step and you plan to do a lot of inserting and deleting in the rest of the program. Of course, a restriction exists in that you may not do any editing between your "present position" and the relative jump address. But you are free to do any other editing somewhere else in the program without disturbing that direct jump. The HP calculators have had this feature for years.

Partitioning is designated in terms of program steps, eight at a time, rather than in terms of data registers, as done in the 59. This allows for much finer tuning.

Plotting is absent on the 88, but program 12 of the ML module has the functional equivalent of plotting. But if you don't like that routine you may write your own in one of your permanent RAM modules and access it when needed in your main program.

As I said earlier, The TI-88 will print everything it finds in the display. This works the other way too: it will display everything that you determine to be printed, even when the printer is not attached. And it will maintain it in the display while the program continues executing until overwritten by new output data. This feature will undoubtedly save many a roll of printing paper.

EE INV EE will not round anymore, but TI has provided OP 13 to do the same. The t-register is gone (requiescant in pace) and so decision-making tests do not depend anymore upon the contents of a specialized register, but can be performed on the contents of ANY register in the 88. There is a utility register, however, but it is NOT used for comparisons. Instead it is used for statistics, polar-to-rectangular, for casette operations and for input/output.

And finally, the trace mode key is gone from the printer but is present on the calculator itself instead. This allows to trace in the display, another paper saver.

Besides all these differences with respect to good, old 59, the TI-88 has some unique features, which will be the subject for many future discussions in the NOTES, I suspect. I will not name them all here (it would take me another three pages to do it) but some of the most attractive ones are the calendar-time-alarm clock, the unformatted mode, the binary/hexadecimal mode and the ability to access all 63 HIR-registers. With respect to the latter you should know what you are doing when you try to "mess" with those. Otherwise, fatal crashes occur from which the only recovery possible sometimes is to remove the battery and let all the charges leak off for a few minutes. HP41C users can tell you horror tales about such happennings! You are dealing here with a constant-memory CMOS calculator and some of the crashes are "doozies".

People ask my opinion about the TI-88 and they expect clear answers, of course:"Is this one going to be a winner or just another Edsel in the market place?" I really don't know and I think it is still much too early to make a definite pronouncement. I remember, a couple of years ago, when the HP41C was first introduced to the members of the local HP club. I was invited, partly because I am on very friendly terms with Bill Kolb, the local organizer of the club and crack programmer, partly because I am nosy and wanted to be invited. The presentation was at the HP headquarters in Rockville, MD. After the first enthusiam had slightly subsided, some of the members confided to me that they had expected much more. They were disappointed, for example, that they could not call individual segments in a CROM as subroutines in their main program, something the TI-59 could do easily. But I pointed out to them that HP probably had hurried this one to the market place and that future versions would be able to do so. (it turned out I was right) I also argued that the hardware alone doesn't make a great calculator. It is the software, and especially the one written by real users in the field, and the availability thereof which will induce a lot of people to part with their money and purchase one of the machines. The superb software plus all the neat tricks discovered by HP41C fans in the HP PPC Club (especially the synthetic programming) made for good sales, which in turn encouraged HP to to bring some fantastic peripherals to the market place. (HP-IL loop, which allows to connect almost any HP instrument to be connected to the HP41C calculator used either as input or as output device. By the way, the two-wire interface loop TI designed into the TI-88 will be even handier for interfacing external devices. And they don't have to carry the TI brand per se.) So, let's wait before we start to laud or condemn and let's start writing software. The software contained in the CROMs is not bad at all (if you knew the people who wrote it you would agree with me that TI couldn't have found better talent) but we, the amateurs, have shown in the past that we can do just as well or better. So, avanti populo, let's show them!

Maurice E.T. Swinnen.

FAST MODE AND GRAPHICS MODE COMBINED: In last issue I told you about this feat having been accomplished by Peter Poloczek in Frankfurt, West Germany. His program is called FAST-GRAFIK-3-D-PLOT. The original one I received from him was written in German and, because I am rather busy these times, I hesitated to translate it in English right away. One of the maxims of a "good" electronics engineer is : Procrastinate until a solution pops up." It worked again this time. Peter wrote me letter this week saying HE had translated the program into English himself. He will send it soon and anybody wanting a copy can now get one from me. (Peter gave me permission to copy) The thing is printed on 25 pages, so it will cost about $ 2.00 copying and another $ 2.00 mailing, including the envelope.That is for US First Class. Overseas members $ 5.50 total.

SUPERCHECKSUM. Björn Gustavsson is the author of this enhanced version. All of the other versions I have seen so far were based on algorithms that had one or another shortcoming in it: some even ignored completely three digits, the exponent (two digits) or the signs (one digit). The several versions in Fast Mode did not show a different algorithm. They were just faster than the plain vanilla variety. Until Björn sent me this one. The contents of each register is regarded as a 16-digit integer which is split into two 8-digit integers and added together. If the sign-digit in a register is 8 or 9, the register will, as usual, overflow when recalled. In such a case it is impossible to get any information about the contents. Nobody has found a satisfactory solution to that problem and I don't expect anybody to find one either. This program ignores such a register, although it counts the number of such registers. This "number" is appended to the 8-digit checksum: xxxxxxxxyy; the resulting checksum is thus a 10-digit integer. As an option, the program can also print a list of all "unsafe" steps. (the steps contained in overflow registers)

Björn emphasizes, however, that:

1. An incorrect checksum tells you with absolute certainty that you have made a mistake in your keying in the program.
2. A correct checksum only tells you that your program PROBABLY is correctly entered. Nothing more, nothing less.

USER INSTRUCTIONS:

1. Load bank 1, press A. Reload bank 1. The display shows a 2 now.
2. Load bank 2. The display shows a zero and the printer prints CHECKSUM.
3. Enter the number of the bank of which you want a checksum. (your own program)
4a. If you just want the checksum, press R/S. Slide in the slot the magnetic card containing the bank to be checksummed. The checksum of it will be printed.
4b. If you also want the list of all "unsafe" steps, as explained above, press +/- R/S. Slide the mag card with your own program into the slot. All "unsafe" steps will be printed in the form xxx.yyy, labeled with a "?". Note now that xxx.yyy may be incorrect and that the checksum might not be affected. Finally, the checksum itself will be printed.
5. If there are more banks on your program to be checksummed, repeat steps 3 and 4a or 4b. Otherwise, press R/S to obtain the checksum for the complete program.
6. If you want to checksum more programs, press R/S and go back to step 3, above.

What I would like to ask the collective membership is the following: Check out this program thouroughly. We, that is two reviewers and I, the editor, did and we cannot find any fault with this one. If I don't get any unfavorable comments on it, I would like to adopt this one as the official TI PPC NOTES checksum program, to be used in future on all published programs.

So, please, if you cannot materially improve upon this one, and show proof how you did it, don't send me any new versions. I have lots of them and none of them bring, as the Dutch say it so eloquently "new earth to the dikes."

TEXAS ON THE MOVE.- Dick Pountain, Calculator Corner editor of Personal Computing World, (Britain) writes in the July 1982 issue: "In order to bring you this information I, and a handful of European journalists, undertook a perilous adventure, risking our lives by flying to Nice, enduring the dangerous rays of the Mediterranean sun and inviting early heart attack by consuming the livers of unwell geese and Roquefort cheese steeped in Armagnac. I only hope you are grateful.....The TI-88 has been developed at Lubbock, Texas and at TI France in Nice....The reason for the delayed appearance (of the 88) ...is that TI developed a new CMOS technology for the 88 and ran into the dreaded substrate problem (the BBC computer was delayed for similar trouble). Twin 4-bit processors of new design are used; they are claimed to run two to three times faster than the 59.....Further increases in speed are claimed from the use of a new fast ROM (three times the speed of the 59).....Current peripherals.. Future plans include a bar code reader.... All in all, the TI-88 is a great improvement on the 59 and represents a catching up with current technology....it is certainly the equal but not superior to the HP-41CV and the Sharp PC-1500. There is no doubt that in the hands of an experienced programmer the TI language is both more economical of memory and more flexible than Basic...(TI) is pushing ahead with a high-level language machine as well; this is due in late '82 and will have a choice of languages, including Basic and Pascal in ROM packs."

Superchecksum, Björn Gustavsson, Listing.

```
000  00  0
001  00  0
002  00  0
003  76  LBL
004  11  A
005  36  PGM
006  02  02
007  71  SBR
008  02  02
009  39  39
010  09  9
011  00  0
012  22  INV
013  58  FIX
014  22  INV
015  57  ENG
016  02  2
017  91  R/S
018  98  ADV
019  47  CMS
020  01  1
021  05  5
022  02  2
023  03  3
024  01  1
025  07  7
026  01  1
027  05  5
028  02  2
029  06  6
030  69  DP
031  00  00
032  69  DP
033  01  01
034  03  3
035  06  6
036  04  4
037  01  1
038  03  3
039  00  0
040  06  6
041  02  2
042  58  FIX
043  02  02
044  69  DP
045  02  02
046  69  DP
047  05  05
048  01  1
049  04  4
050  01  1
051  03  3
052  03  3
053  01  1
054  02  2
055  06  6
056  69  DP
057  00  00
058  69  DP
059  01  01
060  98  ADV
061  58  FIX
062  09  09
063  25  CLR
064  91  R/S
065  29  CP
066  67  EQ
067  02  02
068  57  57
069  77  GE
070  00  00
071  74  74
072  94  +/-
073  22  INV
074  86  STF
075  00  00
076  42  STO
077  34  34
078  85  +
079  01  1
080  95  =
081  58  FIX
082  08  08
083  69  DP
084  02  02
085  69  DP
086  05  05
087  58  FIX
088  09  09
089  02  2
090  04  4
091  00  0
092  49  PRD
093  34  34
094  22  INV
095  44  SUM
096  34  34
097  04  4
098  94  +/-
099  22  INV
100  96  WRT
101  02  2
102  09  9
103  93  .
104  01  1
105  42  STO
106  30  30
107  25  CLR
108  42  STO
109  31  31
110  42  STO
111  32  32
112  35  1/X
113  32  X:T
114  73  RC*
115  30  30
116  67  EQ
117  02  02
118  31  31
119  29  CP
120  67  EQ
121  01  01
122  80  80
123  55  ÷
124  77  GE
125  01  01
126  28  28
127  02  2
128  93  .
129  44  SUM
130  31  31
131  53  (
132  82  HIR
133  11  11
134  52  EE
135  55  ÷
136  52  EE
137  00  0
138  00  0
139  50  I×I
140  54  )
141  42  STO
142  35  35
143  65  ×
144  01  1
145  52  EE
146  07  7
147  75  -
148  59  INT
149  44  SUM
150  31  31
151  95  =
152  65  ×
153  01  1
154  52  EE
155  08  8
156  95  =
157  44  SUM
158  31  31
159  43  RCL
160  35  35
161  50  I×I
162  28  LOG
163  58  FIX
164  00  00
165  52  EE
166  65  ×
167  77  GE
168  01  01
169  71  71
170  04  4
171  93  .
172  44  SUM
173  31  31
174  01  1
175  00  0
176  95  =
177  50  I×I
178  44  SUM
179  31  31
180  08  8
181  44  SUM
182  34  34
183  25  CLR
184  58  FIX
185  09  09
186  97  DSZ
187  30  30
188  01  01
189  12  12
190  43  RCL
191  31  31
192  65  ×
193  01  1
194  00  0
195  00  0
196  85  +
197  43  RCL
198  32  32
199  22  INV
200  86  STF
201  01  01
202  75  -
203  53  (
204  24  CE
205  55  ÷
206  01  1
207  00  0
208  22  INV
209  28  LOG
210  54  )
211  59  INT
212  65  ×
213  01  1
214  00  0
215  22  INV
216  28  LOG
217  95  =
218  44  SUM
219  33  33
220  99  PRT
221  22  INV
222  87  IFF
223  01  01
224  00  00
225  60  60
226  25  CLR
227  91  R/S
228  61  GTO
229  00  00
230  18  18
231  01  1
232  44  SUM
233  32  32
234  87  IFF
235  00  00
236  01  01
237  80  80
238  43  RCL
239  34  34
240  85  +
241  53  (
242  40  IND
243  85  +
244  07  7
245  54  )
246  55  ÷
247  03  3
248  22  INV
249  28  LOG
250  95  =
251  58  FIX
252  03  03
253  99  PRT
254  61  GTO
255  01  01
256  80  80
257  03  3
258  03  3
259  03  3
260  05  5
261  03  3
262  02  2
263  02  2
264  02  2
265  00  0
266  00  0
267  69  DP
268  00  00
269  69  DP
270  01  01
271  69  DP
272  05  05
273  43  RCL
274  33  33
275  61  GTO
276  02  02
277  00  00
```

```
CHECKSUM:

BANK 1
       32.039?
       56.063?
       88.095?
      184.191?
      216.223?
 6543867905.

BANK 2
 4097848000.

PROG
  641715905.

CHECKSUM:

BANK 1
 6543867905.

BANK 2
 4097848000.

PROG
  641715905.
```

HEX KEY CODES,- Patrick Acosta says in the letter accompnaying the hex key code creator program, somewhere else in this issue:
" In section 27, lines 40 through 49 of patent # 3,900,722, it explains that the adder can operate in hexadecimal at state-time So. That explains why only the last digit of a key code at a step divisible by eight can be made hexadecimal. This is apperently the digit added at state-time So as the calculator implements the INS function."

EXECUTION TIMES ON THE TI-59.- In one of the past issues (I don't remember exactly in which one) I suggested as a membership exercise to research execution times on the TI-59. Many of you pointed out it had already been done, notably in the 52-NOTES. So, I looked up all the references given, including Harald M.Otto's Program Tricks #4, better known as BESSERE PROGRAMME, July 1980. Here is a more or less comprehensive list, which might be useful to you when you have to decide which function is faster.

CODE	KEY	TIME	CODE	KEY	TIME
00	0	16	50	IxI	38
01	1	16	52	EE	38
02	2	16	53	(	30
03	3	16	54	)	30
04	4	16	55	÷	185
05	5	16	57	ENG	65
06	6	16	58	FIX	100
07	7	16	59	INT	40
08	8	16	60	DEG	16
09	9	16	61	GTO	90
10	E'	40	62	PG*	85
11	A	40	63	EX*	140
12	B	40	64	PD*	180
13	C	40	65	×	140
14	D	40	66	PAU	500
15	E	40	67	EQ	140
16	A'	40	68	NOP	15
17	B'	40	69	OP	90
18	C'	40	70	RAD	16
19	D'	40	71	SBR	90
20	CLR	17	72	ST*	130
22	INV	16	73	RC*	130
23	LNX	140	74	SM*	160
24	CE	15	75	-	75
25	CLR	17	76	LBL	25
27	INV	16	77	GE	140
28	LOG	220	78	Σ+	900
29	CP	35	79	$\bar{x}$	470
30	TAN	310	80	GRD	16
32	X:T	50	81	RST	20
33	X^2	85	82	HIR	80
34	$\sqrt{X}$	95	83	GO*	90
35	1/X	95	84	OP*	90
37	P→R	1200	85	+	75
38	SIN	420	86	STF	80
39	COS	420	87	IFF	140
40	IND	30	88	DMS	820
42	STO	100	89	π	36
43	RCL	100	92	RTN	60
44	SUM	130	93	.	16
45	Y^X	330	94	+/-	20
47	CMS	200	95	=	30
48	EXC	110	97	DSZ	150
49	PRD	150			

NOTE: Execution times are given in milliseconds.
Execution times are about 3% longer with the PC100 attached.

As a last minute note: Sterling Hartman would like to run benchmarks on the several calculators on the market. Richard Nelson, editor of the HP PPC Journal, ran a quicky on the 59, the41 and the 88. He just let each calculator count up to 200 as in 1+1+1+1 etc ...+ 1 = R/S . The 59 took 7.4 sec to do it, the 41 took 9.2 sec and the 88 only 4.7 sec!!! Don't hang too much importance on this test, though.

RANDOM NUMBER GENERATOR.- To give you a taste of how programs are constructed on this new TI-88, I have included two programs in this issue. One written by Don O'Grady of TI and incorporated as the Random Number Generator in the Master Library module, the other one a concoction of my own called Jive Turkey.

On the next page you will see a downloading of the randon number generator. As you are able to observe, the first 256 steps consist entirely of alpha. Everything flanked front and back by those two silly little signs AL-PH is printed as alpha. Note also that each of these "modules"(logical segments of a computer program) is defined by means of a numerical label and either called by Gtl nn (GTO LBL nn) or Sbl nn (SBR LBL nn). In the first case it is a GTO and therefore execution will stop at the Rtn, in the second case it is a SBR and the RTN will be executed.

The algorithm for this random number generator is straight out of Knut's The Art of Computing, as you can see. Note how TIME is used as a seed for the generator. TIME is always present, because the internal clock is always running, even if you forget to set. It is given in HH.MMSST (hours, minutes, second, tenths of seconds) and when multiplied by 10000 will constitute an ever changing seed.(see steps 0285 and following) Sto 8 means "STO in register 8. The numerous GBR and SBR steps mean respectively "go backwards relative to this step by nn steps" or "go forward relative to this step by nn steps. A "St+ A" reminds me of RPN nomenclature (forgive me Mr. TI). It stands for SUM into register A. StF 0 you should know: Set Flag 0, of course. But IfR 0 is a little more obscure. It stands for "if flag 0 is reset." In order to distinguish between REGISTER C and LABEL C the latter one carries an apostrophe. The same goes for "letter" flag designations.

To guide you trough this strange maze of new calculator jargon you will have to learn (it realy is easy, believe me) I will attempt to put into plain English a short segment of code, starting at step 0389: Store in register C, bring a zero into the display, if that zero is equal to the contents of register C, go to step 0424, otherwise do subroutine 0447 (after which you come back to step 0404, of course. Remember the Rtn at the end of a subroutine) . Then, if flag 0 is reset, do subroutine 0518, otherwise print what is in the display. Then, if flag C is reset do OP 07, that is CONTINUE ON CUE, otherwise do a DSZ on register C. If register C is NOT zero, go backward relative to this step by 22 steps. Otherwise, that is if register C IS zero, go backwards relative to this step by 41 steps. Then add 1 to register A. And so on, and so on.

I hope this will give you some pretaste as to what is at stake. With a little bit of ingenuity and some good, old sleuthing, you will be able to devise some other peculiarities from this and from the Jive Turkey program, somewhere else in this issue.
Maurice Swinnen.

See program on next page.

COMMENTS ON FAST MODE,- Björn Gustavsson offers the following thoughts on that subject: "As you may have guessed, I am an enemy of all unncessary button pushing. In my opinion, a fast mode entry should look like this:
at 000: 0 0 0 LBL A PGM 02 SBR 239 9 0 INV FIX INV ENG 2 R/S 3 R/S 4 R/S (last step = 021) It starts with LBL A and therefore you do not have to press RST R/S; just press A. When bank 1 is reread, the display will show a 2, telling you to enter bank 2. The display will show a 3, telling you to enter bank 3, and so on, up to 4. I see no reason to print the bank you just entered. I think that the Neef routine is the most user-friendly of them all.(TI-58 and 58C users better use one of the newer methods) The programmable hex method requires too much button pushing.

But if we could put the sequence in a module, would it work? I think it is questionable. Who can say what happens when the calculator encounters a hex code in module? But if we put an OP 09 in a module, it could be downloaded when running a program in user memory. Then the fast mode program could be run like any normal mode program! The first 25 or so program steps would be overwritten, but the rest of the program would run like a perfectly normal program."

Random number generator pgm of M-L module.

```
0000 Lbl 0
0002 %%
0003 R
0004 A
0005 N
0006 D
0007 □
0008 M
0009
0010 N
0011 U
0012 M
0013 B
0014 E
0015 R
0016 S
0017 %%
0018 Rtn
0019 Lbl 1
0021 %%
0022 E
0023 N
0024 T
0025
0026 S
0027 E
0028 E
0029 D
0030 %%
0031 Rtn
0032 Lbl 2
0034 %%
0035 U
0036 N
0037 I
0038 F
0039 □
0040 R
0041 M
0042 ?
0043 %%
0044 Rtn
0045 Lbl 3
0047 %%
0048 N
0049 □
0050 R
0051 M
0052 A
0053 L
0054 ?
0055 %%
0056 Rtn
0057 Lbl 4
0059 %%
0060 E
0061 N
0062 T
0063
0064 Q
0065 T
0066 Y
0067
0068 N
0069 E
0070 E
0071 D
0072 E
0073 D
0074 %%
0075 Rtn
0076 Lbl 30
0079 %%
0080 Z
0081 U
0082 F
0083 A
0084 L
0085 L
0086 S
0087 Z
0088 A
0089 H
0090 L
0091 E
0092 N
0093 %%
0094 Rtn
0095 Lbl 31
0098 %%
0099 →
0100
0101 Z
0102 A
0103 H
0104 L
0105
0106 X
0107 %%
0108 Rtn
0109 Lbl 32
0112 %%
0113 G
0114 L
0115 E
0116 I
0117 C
0118 H
0119 V
0120 E
0121 R
0122 T
0123 .
0124 ?
0125 %%
0126 Rtn
0127 Lbl 33
0130 %%
0131 N
0132 □
0133 R
0134 M
0135 A
0136 L
0137 V
0138 E
0139 R
0140 T
0141 .
0142 ?
0143 %%
0144 Rtn
0145 Lbl 34
0148 %%
0149 →
0150
0151 A
0152 N
0153 Z
0154 A
0155 H
0156 L
0157 %%
0158 Rtn
0159 Lbl 60
0162 %%
0163 N
0164 B
0165
0166 A
0167 L
0168 E
0169 A
0170 T
0171 □
0172 I
0173 R
0174 E
0175 S
0176 %%
0177 Rtn
0178 Lbl 61
0181 %%
0182 →
0183
0184 N
0185 B
0186
0187 S
0188 □
0189 U
0190 R
0191 C
0192 E
0193 %%
0194 Rtn
0195 Lbl 62
0198 %%
0199 D
0200 I
0201 S
0202 T
0203 R
0204 .
0205
0206 U
0207 N
0208 I
0209 F
0210 □
0211 R
0212 M
0213 E
0214 ?
0215 %%
0216 Rtn
0217 Lbl 63
0220 %%
0221 D
0222 I
0223 S
0224 T
0225 R
0226 .
0227
0228 N
0229 □
0230 R
0231 M
0232 A
0233 L
0234 E
0235 ?
0236 %%
0237 Rtn
0238 Lbl 64
0241 %%
0242 →
0243
0244 N
0245 B
0246
0247 V
0248 A
0249 L
0250 E
0251 U
0252 R
0253 S
0254 %%
0255 Rtn
0256 Lbl E
0258 PGM 1
0260 Sbr 519
0264 DRG
0265 Sbl a
0267 %%
0268 ?
0269 %%
0270 DP 5
0272 GFR 02
0275 0
0276 Rtn
0277 Sbl a
0279 Adv
0280 Prt
0281 Adv
0282 1
0283 St+ A
0285 (
0286 Time
0287 ×
0288 1
0289 0
0290 0
0291 0
0292 0
0293 )
0294 Sto B
0296 Sbl a
0298 DP 6
0300 Sto B
0302 1
0303 9
0304 9
0305 0
0306 1
0307 7
0308 If< B
0310 Gto 0323
0315 0
0316 If< B
0318 Gto 0335
0323 1
0324 0
0325 7
0326 %%
0327 *
0328 *
0329 %%
0330 DP 7
0332 GBR 50
0335 B
0336 %%
0337 X
0338 =
0339 %%
0340 Prt
0341 1
0342 St+ A
0344 Sbl a
0346 DP 5
0348 GFR 16
0351 1
0352 St+ A
0354 Sbl a
0356 DP 5
0358 GFR 05
0361 1
0362 ±
0363 GBR 24
0366 Inv
0367 StF 0
0369 Sbl a
0371 %%
0372 ÷'
0373
0374 %%
0375 Adv
0376 Prt
0377 1
0378 IfF 0
0380 2
0381 St+ A
0383 Adv
0384 C
0385 Sbl a
0387 DP 6
0389 Sto C
0391 0
0392 If= C
0394 Gto 0424
0399 Sbr 0447
0404 IfR 0
0406 Sbr 0518
0411 Prt
0412 IfR C'
0414 DP 7
0416 Dsz C
0418 GBR 22
0421 GBR 41
0424 1
0425 St+ A
0427 PGM 1
0429 Sbl a
0431 DP 5
0433 GFR 3
0435 GFR 02
0438 0
0439 Rtn
0440 4
0441 St- A
0443 Gto 285
0447 (
0448 (
0449 2
0450 4
0451 2
0452 9
0453 8
0454 ×
0455 B
0456 +
0457 9
0458 9
0459 9
0460 9
0461 1
0462 )
0463 ÷
0464 1
0465 9
0466 9
0467 0
0468 1
0469 7
0470 Sto B
0472 )
0473 (
0474 Int
0475 (
0476 Frc
0477 ×
0478 St× B
0480 Inv
0481 Log
0482 5
0483 )
0484 ÷
0485 Inv
0486 Log
0487 5
0488 )
0489 Exc B
0491 2
0492 0
0493 If= B
0495 Gto 0512
0500 2
0501 (
0502 Int
0503 (
0504 +
0505 .
0506 5
0507 )
0508 )
0509 Exc B
0511 Rtn
0512 2
0513 Sto B
0515 GBR 71
0518 (
0519 (
0520 (
0521 Ln
0522 ×
0523 2
0524 ±
0525 )
0526 ×
0527 Cos
0528 (
0529 2
0530 ×
0531 π
0532 ×
0533 Sbr 447
0537 )
0538 )
0539 Rtn
0540 Lbl A
0542 Sto B
0544 Rtn
0545 Lbl C
0547 Inv
0548 Lbl B
0550 StF 0
0552 Rtn
0553 Lbl D
0555 Sbr 447
0559 IfR 0
0561 Sbr 518
0565 Rtn
```

COMPLETE DATE PRINTER,- About a month after the April joke about the new AF-1 calculator I received from José Gallego in Chula Vista, California, a large package in the mail. In fact the thing was about the size of the magnetic card I had described in the April joke article: 5 by 20 inches. When I opened it it contained a beautiful replica of such a card, nicely hand drawn, color & everything perfect. The title on the card said: Complete Date Printer. In the same package was a smaller, real size mag card with the program on it. So,here it is: Just enter the date in the form MMDD.YYYY and press A. Wait a few seconds and see the date printed out with the correct day of the week. So, always use two digits for the month and the day and four digits for the year. Thus, December 6, 1936 has to entered as 1206.1936.

```
MON. APR. 12, 1982
THU. APR.  1, 1982
WED. JUNE  9, 1926

36133740.   07
36413140.   08
30323140.   09
37411740.   10
43171640.   11
37234140.   12
21352440.   13
25133140.   14
21171440.   15
30133540.   16
13333540.   17
  301345.   18
25413117.   19
25412745.   20
13412240.   21
36173340.   22
32153740.   23
31324240.   24
16171540.   25
```

```
000  76 LBL
001  11  A
002  42 STO
003  04  04
004  36 PGM
005  20  20
006  14  D
007  85  +
008  07  7
009  95  =
010  42 STO
011  06  06
012  73 RC*
013  06  06
014  69 OP
015  01  01
016  43 RCL
017  04  04
018  55  ÷
019  01  1
020  00  0
021  00  0
022  95  =
023  59 INT
024  85  +
025  01  1
026  03  3
027  95  =
028  42 STO
029  06  06
030  73 RC*
```

```
031  06  06
032  69 OP
033  02  02
034  25 CLR
035  42 STO
036  00  00
037  43 RCL
038  02  02
039  71 SBR
040  00  00
041  83  83
042  65  ×
043  01  1
044  00  0
045  00  0
046  95  =
047  85  +
048  05  5
049  07  7
050  95  =
051  65  ×
052  01  1
053  00  0
054  00  0
055  95  =
056  69 OP
057  03  03
058  25 CLR
059  42 STO
060  00  00
061  43 RCL
```

```
062  04  04
063  22 INV
064  59 INT
065  65  ×
066  04  4
067  22 INV
068  28 LOG
069  95  =
070  71 SBR
071  00  00
072  83  83
073  65  ×
074  01  1
075  00  0
076  00  0
077  95  =
078  69 OP
079  04  04
080  69 OP
081  05  05
082  92 RTN
083  29 CP
084  67  EQ
085  01  01
086  22  22
087  55  ÷
088  28 LOG
089  59 INT
090  42 STO
091  05  05
092  69 OP
```

```
093  25  25
094  22 INV
095  28 LOG
096  85  +
097  01  1
098  85  +
099  28 LOG
100  59 INT
101  65  ×
102  01  1
103  00  0
104  00  0
105  49 PRD
106  00  00
107  02  2
108  75  -
109  59 INT
110  44 SUM
111  00  00
112  95  =
113  65  ×
114  01  1
115  00  0
116  97 DSZ
117  05  05
118  00  00
119  96  96
120  25 CLR
121  43 RCL
122  00  00
123  92 RTN
```

JIVE TURKEY.- Most of you have heard the story of Jive Turkey. For those who haven't here goes an abbreviated version: I made the first version for the SR-52 and it was an instant success at PPX, with the exception of their best programmer (whom I have mentioned often in these pages) Don O'Grady, who, by his own admission, hates this program with passion. But Don put it into the Leisure Library module for the 59 nevertheless. So, with trembling heart I dedicate this new TI-88 version to my best friend at TI, Mr. Don O'Grady. I hear he will write the 88 Leisure Library module soon. If he needs the title translated into ideomatically correct German and ditto French, I keep myself available.

To run the program, just press key E, marked CONT. Enter all requested entries through key D, marked ENT.

As you will see, I used the same trick as the Random Number Generator program of the ML module to enter a seed: I used the TIME multiplied by 1EE5. And I employed the same structured sort of programming: each "module" is written as a numerical label, even if you are going to use it only once. We are having so much memory that we wont have to skimp anymore. That doesn't mean we HAVE to program this way. Programming is an art, not an exact science, so each one writes his own way, even old style 59! (hm, hm)

Jive Turkey, Maurice Swinnen, Listing.

```
 ▒ JIVE TURKEY ▒

Probability of
Truth 0 to 100 %
P%=           75.

    READY
Your Guess 0-100
G=            50.
Guess too high

G=            25.
Guess too high

G=            12.
Guess too low

G=            18.
Guess too high

G=            15.
Guess too high

G=            13.
Guess too low

G=            14.
Guess too low

G=            24.
Guess too low

G=            35.
Guess too low

G=            45.
Guess too high

G=            40.
Guess too high

G=            38.
Guess too low

G=            39.
Guess too high

G=            37.
Guess too low

G=            41.
CONGRATULATIONS
ΣG=           15.


For a new game

Press CONT

 ▒ JIVE TURKEY ▒
```

```
0000 Lbl 00
0003 PGM 0109
0008 Sbr 447
0012 ×
0013 1
0014 0
0015 0
0016 =
0017 Int
0018 =
0019 Rtn
0020 Lbl 01
0023 R/S
0024 G
0025 u
0026 e
0027 s
0028 s
0029
0030 t
0031 o
0032 o
0033
0034 h
0035 i
0036 g
0037 h
0038 R/S
0039 Prt
0040 Adv
0041 RF 0
0043 Gt| 10
0046 Lbl 02
0049 R/S
0050 G
0051 u
0052 e
0053 s
0054 s
0055
0056 t
0057 o
0058 o
0059
0060 !
0061 o
0062 w
0063 R/S
0064 Prt
0065 Adv
0066 RF 0
0068 Gt| 10
0071 Lbl 03
0074 5
0075 Sto C
0077 DP 81
0080 R/S
0081 C
```

```
0082 D
0083 N
0084 G
0085 R
0086 A
0087 T
0088 U
0089 L
0090 A
0091 T
0092 I
0093 D
0094 N
0095 S
0096 R/S
0097 Pau
0098 Pau
0099 Prt
0100 D
0101 R/S
0102 Σ
0103 G
0104 =
0105 R/S
0106 Prt
0107 Adv
0108 Adv
0109 DP 81
0112 Dsz C
0114 GBR 7
0116 Pau
0117 Pau
0118 R/S
0119 F
0120 o
0121 r
0122
0123 a
0124
0125 n
0126 e
0127 w
0128
0129 s
0130 a
0131 m
0132 e
0133 R/S
0134 Pau
0135 Prt
0136 Adv
0137 R/S
0138 P
0139 r
0140 e
0141 s
0142 s
0143
0144 C
```

```
0145 D
0146 N
0147 T
0148 R/S
0149 Pau
0150 Prt
0151 Adv
0152 Adv
0153 R/S
0154 Lbl 04
0157 R/S
0158 ▒
0159
0160 J
0161 I
0162 V
0163 E
0164
0165 T
0166 U
0167 R
0168 K
0169 E
0170 Y
0171
0172 ▒
0173 R/S
0174 Prt
0175 Pau
0176 Pau
0177 Adv
0178 Rtn
0179 Lbl 05
0182 D
0183 R/S
0184 Σ
0185 G
0186 =
0187 R/S
0188 Prt
0189 Adv
0190 Rtn
0191 Lbl 06
0194 R/S
0195 P
0196 r
0197 o
0198 b
0199 a
0200 b
0201 i
0202 l
0203 i
0204 t
0205 y
0206
0207 o
0208 f
0209 R/S
```

```
0210 Prt
0211 Pau
0212 R/S
0213 T
0214 r
0215 u
0216 t
0217 h
0218
0219 0
0220
0221 t
0222 o
0223
0224 1
0225 0
0226 0
0227
0228 %
0229 R/S
0230 Prt
0231 Pau
0232 DP 81
0235 Dfn P
0237 R/S
0238 P
0239 %
0240 =
0241 R/S
0242 Prt
0243 Adv
0244 Rtn
0245 Lbl 07
0248 R/S
0249
0250
0251
0252
0253 R
0254 E
0255 A
0256 D
0257 Y
0258 R/S
0259 Prt
0260 Pau
0261 Rtn
0262 Lbl 08
0265 R/S
0266 Y
0267 o
0268 u
0269 r
0270
0271 G
0272 u
0273 e
0274 s
0275 s
```

```
0276
0277 0
0278 -
0279 1
0280 0
0281 0
0282 R/S
0283 Prt
0284 Pau
0285 Rtn
0286 Lbl 09
0289 DP 81
0292 Dfn G
0294 R/S
0295 G
0296 =
0297 R/S
0298 Prt
0299 Rtn
0300 Lbl E
0302 DP 01
0305 DP 84
0308 RF 0
0310 CMs
0311 Sbl 04
0314 Sbl 06
0317 Time
0318 ×
0319 1
0320 H
0321 5
0322 =
0323 Inv
0324 H
0325 Sto B
0327 Sbl 00
0330 Sto S
0332 Sbl 07
0335 Sbl 08
0338 Lbl 10
0341 Sbl 09
0344 Sbl 00
0347 Sto L
0349 If> P
0351 StF 0
0353 1
0354 St+ D
0356 G
0357 If= S
0359 Gt| 03
0362 If> S
0364 Gt| 11
0367 IfF 0
0369 Gt| 01
0372 Gt| 02
0375 Lbl 11
0378 IfF 0
0380 Gt| 02
0383 Gt| 01
```

Book Report (W. J. Widmer)

ENGINEERING STATISTICS WITH A PROGRAMMABLE CALCULATOR by William Volk (1982; McGraw-Hill Book Co., 1221 Avenue of Americas, New York, N.Y. 10020). 362 + iv pages; 6" x 9", hard cover. $19.95. Mr. Volk is a chemical engineer, 35 years in research/development/ statistical analysis, including teaching. And he is the author of another book, Applied Statistics for Engineers.

This book has 7 chapters, 2 appendices, references, index as follows:

Chapter 1--Introduction: objectives, methodology
" 2--Statistical Parameters
" 3--Probability Distributions
" 4--The t-Test
" 5--Chi Square (χ^2) Test
" 6--Variance and the Analysis of Variance
" 7--Regression
Appendix A--Hewlett-Packard (HP-97) Calculator Programs
" B--Texas Instruments (TI-59) Calculator Programs

Each chapter presents "methods of statistics that will be helpful to the engineer in the analysis of experimental data. The statistical methods are presented from an applications point of view without detailed theoretical development" but "sufficient theory is included so that the application may be understood." Quotes are from the author's introduction and I feel that he has done this in a clear, readable manner. Programs are presented in detail, with explanations and flow diagrams (the latter will be of help to those readers wishing to extend the coverage to minicomputers as well). The numerical examples for particular programs include run-times for both HP and TI instruments, a highly appreciated feature. All-in-all, this is a lucid exposition of working statistics and useful application programs at a reasonable price for such a high-quality text.

Book Report (W. J. Widmer)

SCIENTIFIC ANALYSIS FOR PROGRAMMABLE CALCULATORS (with algebraic operating systems) by H. R. Meck (1981; Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632). Spectrum Book No. S-808. 175 plus xi pages; 7" x 9", soft cover. $7.95. Subtitle note: Programming Techniques, Science and Engineering Applications. 5 chapters, Appendix, Solutions to Selected Problems.

Chap. 1--Introduction: Polynomials, functions, transfers, recurrence, etc.
Chap. 2--Roots of Equations: Method of Iteration; Newton-Raphson; Secant.
Chap. 3--Higher Transcendental Functions; Sine, Cosine, Exponential, Elliptic & Dawson Integrals; Error, Factorial, Bessel & Beta Functions.
Chap. 4--Numerical Integration: Simpson, Gauss, Romberg, etc.
Chap. 5--Differential Equations: 1st Order, 2nd Order, Special Order, 4th Order Equations; Initial Value & Boundary Value Problems.
Appendix--Numerical Methods: Gauss Integration; Differential Equations.

Continues next page.

Book report, W.J. Widmer, cont.

References.

Each chapter has discussion, program steps for TI-58/59, sample problems, and problems for user work-out. Suggested solutions to selected (not all) problems are given in the post-appendix. Index aids in ready location of topics.

This is a handy little book which, in addition to serving as a concise summarizing reference for functions, provides useful routines for PPC application work. ("The programs are written in short segments. Each segment, which appears as a horizontal line, performs a specific function within the program. Each program is accompanied by comments to explain the function of each line."-- author's quote.) The reader is assumed to be familiar with undergraduate engineering mathematics. This book focuses on topics largely neglected in TI manuals and similar PPC books. In my opinion, an excellent book for advanced calculator programming. Well-printed, clear sketches and, at today's general prices, a bargain. On page 70, in Eq. 33 the - sign just left of 1/12x should be +.

PROGRAMS FOR $$$$.- In a past issue I reported the fact that Digital Design, a well known technical magazine, is looking for circuits and computer or programmable calculator programs. Again in the June 1982 issue I found an ad asking for the same.

Programmers, here is your chance to make some "dough", to pay for that TI-88 you are soon going to buy.

Application Notebook Circuits Wanted

We are paying $70 per Application Notebook entry that is submitted to us and accepted for publication. Have you designed and breadboarded a novel and useful circuit that would interest other engineers? Or have you written a unique microcomputer (or even programmable calculator) subroutine or program? If so, then we invite you to share your experiences with our readers.

What if your circuit is simple? Don't despair, big is not better, and circuit complexity doesn't guarantee a winning entry. What do we look for in a design entry? Both simple and complex winning entries should pass one test—fulfilling a need. So, if that circuit idea tucked away in your notebook or file is an ideal "cookbook" circuit, it will be clipped, saved, referred to and used by engineers for years to come.

When you send in your program or circuit idea, include enough detailed program or circuit description. "Should I worry about grammar?" is a question we're often asked. The answer is "No," since we will edit it. Include flow diagrams, printout, schematics, block diagrams and a detailed description. Please type and leave a space between each line.

Send your ideas to: Circuits Editor, *Digital Design*, 1050 Commonwealth Ave., Boston, MA 02215.

Paul Snigier, Editor

---

100 ITEM INVENTORY CONTROL.- Michael Malak, a high school student from Alexandria, Virginia wrote this nice and handy program. It will store up to 100 10-digit stock numbers plus the number of items, from 0 to 999. This 10:3 ratio may be changed to, for example, 9:4 by changing just one program step. The number of items of any stock item may be extracted in about 12 sec. The number of items may then be changed at will. All the data may be listed with a separate 39-step program.

The program uses PGM 06 of the Math/Utility module, the Shell sort. Code numbers are searched by the bisection method. If a code number is not found within 15 sec it means that that one does not exist. The program might now print out an (obviously) erroneous number. To change the 10:3 to a 9:4 ratio, change step 018 to read 4. Note that the sum of both numbers, either 10 and 3 or 9 and 4 always must be equal to 13.

User instructions:

1. To enter data: enter code # and press A. This is the register number data will be stored in.
2. Conclude the data entry by pressing PGM 06 B, then RST.
3. Search for data: Enter the code # and press C. Data is printed.
4. Enter new # of items and press D. New data is printed.
5. Change code # of a specific register: enter the register # and press D, then CLR. Ignore flashing -1. Enter data via step 1 and 2.
6. List all data: Load program from separate card (39-step one) Enter the starting register and press A.

000	00	0
001	92	RTN
002	76	LBL
003	19	D'
004	82	HIR
005	15	15
006	55	÷
007	02	2
008	85	+
009	93	.
010	05	5
011	95	=
012	59	INT
013	82	HIR
014	05	05
015	92	RTN
016	76	LBL
017	17	B'
018	03	3
019	22	INV
020	28	LOG
021	95	=
022	92	RTN
023	76	LBL
024	16	A'
025	75	-
026	59	INT
027	99	PRT
028	95	=
029	65	×
030	17	B'
031	99	PRT
032	43	RCL
033	00	00

034	99	PRT
035	98	ADV
036	92	RTN
037	76	LBL
038	11	A
039	85	+
040	53	(
041	82	HIR
042	12	12
043	42	STO
044	00	00
045	85	+
046	01	1
047	54	)
048	92	RTN
049	55	÷
050	17	B'
051	36	PGM
052	06	06
053	11	A
054	69	OP
055	19	19
056	87	IFF
057	07	07
058	01	01
059	41	41
060	16	A'
061	43	RCL
062	00	00
063	82	HIR
064	02	02
065	81	RST
066	76	LBL
067	13	C

068	82	HIR
069	07	07
070	32	X:T
071	82	HIR
072	13	13
073	59	INT
074	67	EQ
075	01	01
076	44	44
077	05	5
078	00	0
079	42	STO
080	00	00
081	82	HIR
082	05	05
083	73	RC*
084	00	00
085	59	INT
086	32	X:T
087	82	HIR
088	17	17
089	67	EQ
090	01	01
091	21	21
092	77	GE
093	01	01
094	25	25
095	19	D'
096	22	INV
097	44	SUM
098	00	00
099	61	GTO
100	00	00
101	83	83

102	76	LBL
103	14	D
104	55	÷
105	17	B'
106	85	+
107	73	RC*
108	00	00
109	69	OP
110	19	19
111	87	IFF
112	07	07
113	01	01
114	37	37
115	59	INT
116	95	=
117	72	ST*
118	00	00
119	16	A'
120	81	RST
121	73	RC*
122	00	00
123	16	A'
124	81	RST
125	73	RC*
126	00	00
127	29	CP
128	67	EQ
129	00	00
130	95	95
131	19	D'
132	44	SUM
133	00	00
134	61	GTO
135	00	00

136	83	83
137	82	HIR
138	13	13
139	59	INT
140	95	=
141	82	HIR
142	03	03
143	24	CE
144	01	1
145	00	0
146	00	0
147	42	STO
148	00	00
149	82	HIR
150	13	13
151	16	A'
152	81	RST
153	76	LBL
154	12	B
155	82	HIR
156	02	02
157	01	1
158	82	HIR
159	52	52

000	76	LBL
001	16	A'
002	75	-
003	59	INT
004	99	PRT

005	95	=
006	65	×
007	03	3
008	22	INV
009	28	LOG
010	95	=
011	99	PRT
012	43	RCL
013	00	00
014	99	PRT
015	98	ADV
016	92	RTN
017	76	LBL
018	11	A
019	42	STO
020	00	00
021	69	OP
022	20	20
023	73	RC*
024	00	00
025	69	OP
026	19	19
027	87	IFF
028	07	07
029	00	00
030	35	35
031	16	A'
032	61	GTO
033	00	00
034	21	21
035	82	HIR
036	12	12
037	16	A'
038	92	RTN

HELP WANTED - Robert J. Schmeelk, 31 Brooks Street, Hicksville L.I. N.Y. 11801 would like to meet other calculator fans in his area to exchange ideas of accounting, finance, taxation and find ways to employ the 59 in these fields. Robert bought his calculator recently and would appreciate any help and tips on programming in general. Please write him or look him up in the phone book and give him a call.

CREATING HEX-KEYCODES- Patrick Acosta

There are several minor irritations in creating hex-codes via the ROM method. For instance,

(1) Hex-codes can only be created up to step 312.

(2) Different methods of getting into ROM must be used depending on which CROM is installed.

(3) It is not a trivial exercise to determine which keycode to write in RAM and how many times to press Ins to create the hex-code you want.

These problems are alleviated somewhat by a way of implanting hex-codes which doesn't even require you to leave the comfort of user RAM. The general method is to jump to the first step of an octet without re-loading the command buffer register. Then, assuming the command buffer register and the RAM octet contain the proper keycodes, one Ins will give the desired hex-code.

For example, to create the hex-code h12 (= 0C hexidecimal) at step 400, write the following program in an otherwise cleared memory; 000: Lbl B 5 0 9 EE 1 1 +/- + 4 = Deg Fix 0 Nop R/S +/-. Then from the keyboard press B. This puts the number 4.000000005090 into the display register. Then press STF Ind 7 INV. This loads the above number into the flag register. Since the thirteenth digit is zero, the calculator stays in user memory. The tenth and eleventh digits send the calculator to the fiftieth octet, and the twelfth digit (which holds the byte within the octet), being a 9, sends the calculator to the first step of the octet but <u>without</u> re-loading the command buffer register. Thus, when you press LRN, you will see the keycode 94 at step 400. This keycode is left in the command buffer register from step 017. Now the situation is similar to that when creating hex-codes while in ROM in that the keycode seen in LRN mode is not the keycode actually in that location in RAM. Now, while in LRN mode, press Ins and h12 appears at step 400 (according to the generalized rule: hex-code = 00 minus the keycode apparently at step 400 plus the keycode actually at step 400 using hexidecimal arithmetic for the ones digits and decimal arithmetic for the tens digits, In this example, 00-94+00= 0C.) Before running any program, get out of LRN mode and press CLR. Otherwise, the first keycode the calculator encounters will not be executed.

Any other hex-code can be created at step 400 by changing the keycode following the R/S of routine B and the hex-code can be put at any step divisible by eight up to step 872 (392 on the TI-58) by changing the ninth through eleventh digits of the number generated by routine B. Just be sure the R/S does not fall on the last step of an octet. That would leave the command buffer register empty which is no help at all.

This "RAM method" of creating hex-codes enables you to put hex-codes up to step 872 and in fewer keystrokes than the ROM method. Also, since this method is CROM independent, the instructions for hex-initializing a program are always the same no matter which CROM is installed. The only restrictions on partitioning are that the step you want the hex-code at must be in the partition and a 0 Op 17 partition can not be used since STF Ind 7 INV gives an error condition in that case.

The following program is one that I've found useful for a quick look at any desired hex-code. It uses the RAM method and dynamic code generation to implant the hex-code at step 176. The instructions are simple.

(1) Enter the hex-code into the display using the hnn notation. The tens digit can be anything. The ones digit can be 0 through 5, corresponding to the hex digits A through F respectively.

(2) Press A. After two seconds, 4. will appear in the display.

(3) Press STF IND 7 INV LRN. You'll be at step 176.

(4) Press Ins and the hex-code appears immediately.

(5) Get out of LRN mode and press CLR before experimenting.

HEX-KEYCODE CREATOR- P. W. ACOSTA

```
000  76  LBL
001  11   A
002  94  +/-
003  85  +
004  01  1
005  00  0
006  69  OP
007  17  17
008  09  9
009  02  2
010  00  0
011  06  6
012  93  .
013  09  9
014  01  1
015  00  0
016  07  7
017  04  4
018  03  3
019  95  =
020  42  STO
021  99  99
022  02  2
023  02  2
024  09  9
025  52  EE
026  01  1
027  01  1
028  94  +/-
029  85  +
030  04  4
031  95  =
032  42  STO
033  07  07
034  25  CLR
035  42  STO
036  97  97
037  09  9
038  69  OP
039  17  17
040  60  DEG
041  58  FIX
042  00  0
043  61  GTO
044  01  01
045  63  63
```

DANISH NEWSLETTER.- Most of the articles in this issue are done by means of my newly acquired wordprocessor program, a real Godsend. But occasionally one of the members presents his copy in such a neat fashion, that retyping it would consitute a capital sin, even if it is not done in two columns.

A case in point is Patrick Acosta's CREATING HEX-KEYCODES. Typing is perfect and I have enough typing to do as it is, so I left it in its original state, even the program on the left. You see, Patrick has done all of his discoveries of hex codes on a plain vanilla TI-58, without even the benefit of a printer. So, Patrick either hand-writes or types his program submissions.

Because there was still this enormous space left over to the right hand side of Patrick's program listing, I thought you would find it interesting to see a (reduced) copy of the front page of issue 9 of the Danish newsletter.

The color is green, therefore the slight haze. But everything is still quite readable. Our LRN newsletter figures prominently among the PPX Exchange, Programbiten and PGM. Poor Thomas Coppens and his TI-SOFT newsletter in Belgium got left out. Watch out, Hans Peter Nielsen, Copenhagen is not so far away from Antwerp. Tom might even visit you by bicycle!(The Belgians are famous for being crack cycle racers. Witness the yearly Tour de France.)

CHESS 2.1. Michael Sperber in Fuerth, West Germany, of Graphics Mode fame, is the author of this program.

Yes, you are seeing right: this is an honest to goodness chess program. Not a scaled down version that can play a few moves with lots of restrictions. No, this one is the first and only one so far real chess game program. Michael calls it, of course, Schach 2.1. If you want to know how to pronounce it correctly (we calculator nuts are sticklers for accuracy, aren't we?) you just think about the pronounciation of the American word for disgust: "Yagh." Then place a "sh" in front of it instead of the "Y" and "I think she's got it!"

Before you get your hopes up too high with respect to speed of execution of the game, let me tell you that a move by the calculator may take anywhere from 3 to 6 hours. Some moves have been clocked to take 12 hours! The program, according to Michael,"finds its move by trying any possible move and all possible countermoves that you, the player, may make and by choosing the move that it thinks best."

The format for any move, input and output, is xx.yy, in which xx is the number of the square you move from and yy the number of the square you want to move to. These numbers may be found in the table below. You never indicate which PIECE is actually moved. By you giving the number of the square, the calculator knows which piece you mean.

Be extremely careful with entries as the calculator does not check them in any way and it is virtually impossible to correct a wrong entry.

RECORDING INSTRUCTIONS: Partition to 1 OP 17 and key in program steps 000 to 855. Repartition to 6 OP 17 and load registers 00 through 12 with the data as shown. Record the entire program on two mag cards, four sides.

USER INSTRUCTIONS:

1. Read in the program, four sides, and start by pressing E. Now prepare your chessboard to trace the game by means of pencil and paper.
2. For a normal game: Enter your move in the form xx.yy, and press A. Now have some patience. See the countermove by the program printed and displayed. Repeat # 2 as often as necessary.
3. Normally you play white while the TI-59 plays black. If you want to change it, press C.
4. If you want to make a move without requiring the calculator to make a countermove, use key B instead of A as you dis in # 1, above. Thus you enter xx.yy B.
5. If you want to find out how the program works you may elect to have the TI-59 play against itself. Press D and enter one initial move in the form xx.yy and press A. From now on the program make moves and countermoves all by itself. (Don't expect a large number of moves made, even if you go on a weekend vacation and leave your calculator having fun on the home front all by itself. My harvest after three days was rather meager. Ed.) With the PC 100 connected, all moves will be printed. If you don't have a printer available, don't despair. Once a move is displayed you may restart with R/S. (Won't work over the weekend when you're on the beach somewhere, of course.Ed.)

NOTE: When the printer is connected, you will see a one-digit print-out each time a piece is thrown out. The code for this digit is:
1= pawn, 2= knight, 3= bishop,
4= castle, 5= queen, 6= king,
+ means white, - means black.

	1	2	3	4	5	6	7	8	
8	81	82	83	84	85	86	87	88	8
7	71	72	73	74	75	76	77	78	7
6	61	62	63	64	65	66	67	68	6
5	51	52	53	54	55	56	57	58	5
4	41	42	43	44	45	46	47	48	4
3	31	32	33	34	35	36	37	38	3
2	21	22	23	24	25	26	27	28	2
1	11	12	13	14	15	16	17	18	1
	A	B	C	D	E	F	G	H	

Listing Schach 2.1

```
000  98 ADV
001  02  2
002  69 DP
003  17  17
004  25 CLR
005  91 R/S
006  81 RST
007  76 LBL
008  16 A'
009  65  ×
010  93  .
011  01  1
012  75  -
013  59 INT
014  42 STO
015  00  00
016  95  =
017  32 X:T
018  01  1
019  03  3
020  45 Y^x
021  32 X:T
022  45 Y^x
023  01  1
024  00  0
025  94 +/-
026  65  ×
027  32 X:T
028  73 RC*
029  00  00
030  95  =
031  22 INV
032  59 INT
033  65  ×
034  01  1
035  03  3
036  95  =
037  52 EE
038  22 INV
039  52 EE
040  59 INT
041  75  -
042  06  6
043  95  =
044  92 RTN
045  76 LBL
046  17 B'
047  16 A'
048  75  -
049  48 EXC
050  18  18
051  95  =
052  94 +/-
053  55  ÷
054  32 X:T
055  55  ÷
056  01  1
057  03  3
058  95  =
059  52 EE
060  22 INV
061  52 EE
062  74 SM*
063  00  00
064  43 RCL
065  18  18
066  92 RTN
067  16 A'
068  42 STO
069  00  00
070  65  ×
071  01  1
072  94 +/-
073  32 X:T
074  43 RCL
075  15  15
076  69 DP
077  10  10
078  95  =
079  92 RTN
080  76 LBL
081  19 D'
082  65  ×
083  43 RCL
084  15  15
085  69 DP
086  10  10
087  85  +
088  43 RCL
089  15  15
090  50 I×I
091  95  =
092  65  ×
093  93  .
094  01  1
095  95  =
096  42 STO
097  16  16
098  59 INT
099  32 X:T
100  08  8
101  22 INV
102  77 GE
103  81 RST
104  00  0
105  77 GE
106  81 RST
107  43 RCL
108  16  16
109  22 INV
110  59 INT
111  32 X:T
112  93  .
113  09  9
114  67 EQ
115  81 RST
116  00  0
117  67 EQ
118  81 RST
119  22 INV
120  76 LBL
121  81 RST
122  86 STF
123  01  01
124  01  1
125  00  0
126  49 PRD
127  16  16
128  43 RCL
129  16  16
130  92 RTN
131  76 LBL
132  15  E
133  22 INV
134  58 FIX
135  01  1
136  69 DP
137  17  17
138  61 GTO
139  82 HIR
140  76 LBL
141  18 C'
142  87 IFF
143  00  00
144  02  02
145  36  36
146  50 I×I
147  42 STO
148  15  15
149  22 INV
150  86 STF
151  02  02
152  22 INV
153  86 STF
154  03  03
155  86 STF
156  00  00
157  16 A'
158  42 STO
159  14  14
160  69 DP
161  10  10
162  49 PRD
163  15  15
164  43 RCL
165  14  14
166  50 I×I
167  32 X:T
168  01  1
169  67 EQ
170  02  02
171  67  67
172  06  6
173  67 EQ
174  02  02
175  60  60
176  02  2
177  67 EQ
178  02  02
179  62  62
180  32 X:T
181  85  +
182  07  7
183  95  =
184  42 STO
185  14  14
186  73 RC*
187  14  14
188  50 I×I
189  22 INV
190  59 INT
191  29 CP
192  67 EQ
193  03  03
194  61  61
195  65  ×
196  01  1
197  00  0
198  00  0
199  95  =
200  42 STO
201  14  14
202  43 RCL
203  15  15
204  50 I×I
205  42 STO
206  16  16
207  43 RCL
208  16  16
209  85  +
210  43 RCL
211  14  14
212  59 INT
213  71 SBR
214  00  00
215  91  91
216  87 IFF
217  01  01
218  02  02
219  50  50
220  16 A'
221  42 STO
222  00  00
223  69 DP
224  10  10
225  32 X:T
226  43 RCL
227  15  15
228  69 DP
229  10  10
230  67 EQ
231  02  02
232  50  50
233  43 RCL
234  00  00
235  92 RTN
236  87 IFF
237  03  03
238  03  03
239  65  65
240  22 INV
241  87 IFF
242  06  06
243  02  02
244  50  50
245  22 INV
246  87 IFF
247  02  02
248  02  02
249  07  07
250  43 RCL
251  14  14
252  94 +/-
253  29 CP
254  77 GE
255  01  01
256  89  89
257  61 GTO
258  02  02
259  00  00
260  05  5
261  32 X:T
262  86 STF
263  02  02
264  61 GTO
265  01  01
266  80  80
267  86 STF
268  03  03
269  43 RCL
270  15  15
271  50 I×I
272  85  +
273  01  1
274  00  0
275  65  ×
276  43 RCL
277  15  15
278  69 DP
279  10  10
280  95  =
281  42 STO
282  16  16
283  16 A'
284  29 CP
285  67 EQ
286  03  03
287  29  29
288  61 GTO
289  03  03
290  32  32
291  02  2
292  00  0
293  19 D'
294  05  5
295  93  .
296  01  1
297  87 IFF
298  01  01
299  03  03
300  29  29
301  43 RCL
302  15  15
303  50 I×I
304  65  ×
305  93  .
306  01  1
307  95  =
308  59 INT
309  32 X:T
310  02  2
311  67 EQ
312  03  03
313  19  19
314  07  7
315  22 INV
316  67 EQ
317  03  03
318  32  32
319  43 RCL
320  16  16
321  16 A'
322  29 CP
323  22 INV
324  67 EQ
325  03  03
326  32  32
327  93  .
328  01  1
329  42 STO
330  00  00
331  92 RTN
332  09  9
333  19 D'
334  87 IFF
335  01  01
336  03  03
337  46  46
338  71 SBR
339  00  00
340  67  67
341  69 DP
342  10  10
343  67 EQ
344  02  02
345  33  33
346  01  1
347  01  1
348  19 D'
349  87 IFF
350  01  01
351  03  03
352  61  61
353  71 SBR
354  00  00
355  67  67
356  69 DP
357  10  10
358  67 EQ
359  02  02
360  33  33
361  22 INV
362  86 STF
363  00  00
364  92 RTN
365  43 RCL
366  16  16
367  75  -
368  43 RCL
369  15  15
370  50 I×I
371  95  =
372  50 I×I
373  32 X:T
374  01  1
375  00  0
376  67 EQ
377  02  02
378  91  91
379  09  9
380  67 EQ
381  03  03
382  46  46
383  01  1
384  01  1
385  67 EQ
386  03  03
387  61  61
388  61 GTO
389  03  03
390  32  32
391  82 HIR
392  18  18
393  65  ×
394  93  .
395  01  1
396  95  =
397  59 INT
398  32 X:T
399  08  8
400  67 EQ
401  04  04
402  08  08
403  01  1
404  22 INV
405  67 EQ
406  07  07
407  53  53
408  05  5
409  48 EXC
410  18  18
411  69 DP
412  10  10
413  49 PRD
414  18  18
415  61 GTO
416  07  07
417  53  53
418  76 LBL
419  11  A
420  42 STO
421  14  14
422  43 RCL
423  09  09
424  29 CP
425  77 GE
426  04  04
427  33  33
428  07  7
429  68 NOP
430  68 NOP
431  85  +
432  22 INV
433  86 STF
434  05  05
435  03  3
436  06  6
437  95  =
438  69 DP
439  04  04
440  43 RCL
441  14  14
442  69 DP
443  06  06
444  71 SBR
445  07  07
446  35  35
447  25 CLR
448  82 HIR
449  04  04
450  06  6
451  07  7
452  00  0
453  94 +/-
454  82 HIR
455  03  03
456  08  8
457  08  8
458  82 HIR
459  06  06
460  16 A'
461  69 DP
462  10  10
463  32 X:T
464  43 RCL
465  09  09
466  69 DP
467  10  10
468  22 INV
469  67 EQ
470  06  06
471  97  97
472  82 HIR
473  16  16
474  18 C'
475  42 STO
476  19  19
477  22 INV
478  87 IFF
479  00  00
```

```
480  06  06
481  97  97
482  33  X²
483  33  X²
484  75  -
485  82  HIR
486  13  13
487  95  =
488  82  HIR
489  08  08
490  00  0
491  42  STO
492  13  13
493  87  IFF
494  02  02
495  06  06
496  89  89
497  87  IFF
498  03  03
499  06  06
500  91  91
501  00  0
502  42  STO
503  18  18
504  82  HIR
505  16  16
506  50  I×I
507  17  B'
508  43  RCL
509  16  16
510  82  HIR
511  07  07
512  17  B'
513  22  INV
514  86  STF
515  00  00
516  43  RCL
517  14  14
518  82  HIR
519  05  05
520  43  RCL
521  15  15
522  82  HIR
523  06  06
524  02  2
525  00  0
526  00  0
527  00  0
528  94  +/-
529  42  STO
530  17  17
531  08  8
532  08  8
533  42  STO
534  15  15
535  16  A'
536  69  DP
537  10  10
538  32  X:T
539  43  RCL
```

```
540  09  09
541  69  DP
542  10  10
543  94  +/-
544  22  INV
545  67  EQ
546  05  05
547  83  83
548  43  RCL
549  15  15
550  18  C'
551  22  INV
552  87  IFF
553  00  00
554  05  05
555  83  83
556  29  CP
557  67  EQ
558  05  05
559  64  64
560  33  X²
561  33  X²
562  32  X:T
563  22  INV
564  86  STF
565  06  06
566  43  RCL
567  17  17
568  77  GE
569  05  05
570  50  50
571  82  HIR
572  18  18
573  22  INV
574  77  GE
575  06  06
576  41  41
577  32  X:T
578  42  STO
579  17  17
580  61  GTO
581  05  05
582  50  50
583  43  RCL
584  15  15
585  75  -
586  69  DP
587  10  10
588  95  =
589  29  CP
590  67  EQ
591  06  06
592  06  06
593  42  STO
594  15  15
595  50  I×I
596  71  SBR
597  00  00
598  91  91
599  87  IFF
```

```
600  01  01
601  05  05
602  83  83
603  61  GTO
604  05  05
605  35  35
606  82  HIR
607  18  18
608  75  -
609  43  RCL
610  17  17
611  95  =
612  22  INV
613  77  GE
614  06  06
615  41  41
616  22  INV
617  67  EQ
618  06  06
619  24  24
620  87  IFF
621  05  05
622  06  06
623  41  41
624  52  EE
625  82  HIR
626  33  33
627  25  CLR
628  82  HIR
629  17  17
630  65  ×
631  93  .
632  00  0
633  01  1
634  85  +
635  82  HIR
636  16  16
637  50  I×I
638  95  =
639  82  HIR
640  04  04
641  43  RCL
642  19  19
643  29  CP
644  67  EQ
645  06  06
646  48  48
647  22  INV
648  86  STF
649  06  06
650  42  STO
651  18  18
652  22  INV
653  59  INT
654  67  EQ
655  06  06
656  60  60
657  00  0
658  42  STO
659  18  18
```

```
660  82  HIR
661  17  17
662  42  STO
663  16  16
664  17  B'
665  82  HIR
666  16  16
667  42  STO
668  15  15
669  50  I×I
670  17  B'
671  82  HIR
672  15  15
673  42  STO
674  14  14
675  86  STF
676  00  00
677  22  INV
678  86  STF
679  02  02
680  22  INV
681  86  STF
682  03  03
683  86  STF
684  40  IND
685  13  13
686  61  GTO
687  04  04
688  74  74
689  02  2
690  93  .
691  03  3
692  42  STO
693  13  13
694  61  GTO
695  05  05
696  01  01
697  82  HIR
698  16  16
699  75  -
700  69  DP
701  10  10
702  95  =
703  82  HIR
704  06  06
705  29  CP
706  67  EQ
707  07  07
708  20  20
709  50  I×I
710  71  SBR
711  00  00
712  91  91
713  87  IFF
714  01  01
715  06  06
716  97  97
717  61  GTO
718  04  04
719  60  60
```

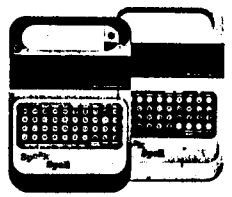
```
720  82  HIR
721  14  14
722  87  IFF
723  07  07
724  07  07
725  89  89
726  87  IFF
727  04  04
728  07  07
729  90  90
730  76  LBL
731  12  B
732  99  PRT
733  42  STO
734  14  14
735  29  CP
736  67  EQ
737  00  00
738  06  06
739  88  DMS
740  25  CLR
741  42  STO
742  18  18
743  43  RCL
744  14  14
745  59  INT
746  17  B'
747  50  I×I
748  32  X:T
749  01  1
750  67  EQ
751  03  03
752  91  91
753  82  HIR
754  18  18
755  17  B'
756  29  CP
757  67  EQ
758  07  07
759  67  67
760  99  PRT
761  50  I×I
762  32  X:T
763  06  6
764  67  EQ
765  00  00
766  06  06
767  43  RCL
768  14  14
769  92  RTN
770  76  LBL
771  13  C
772  01  1
773  94  +/-
774  49  PRD
775  09  09
776  92  RTN
777  76  LBL
778  14  D
779  25  CLR
```

```
780  86  STF
781  04  04
782  02  2
783  00  0
784  69  DP
785  07  07
786  69  DP
787  18  18
788  25  CLR
789  92  RTN
790  13  C
791  82  HIR
792  14  14
793  61  GTO
794  11  A
795  00  0
796  00  0
797  00  0
798  00  0
799  00  0
800  76  LBL
801  82  HIR
802  25  CLR
803  01  1
804  05  5
805  55  ÷
806  09  9
807  09  9
808  95  =
809  98  ADV
810  82  HIR
811  05  05
812  98  ADV
813  82  HIR
814  08  08
815  03  3
816  06  6
817  01  1
818  05  5
819  02  2
820  03  3
821  01  1
822  03  3
823  01  1
824  05  5
825  69  DP
826  02  02
827  02  2
828  03  3
829  00  0
830  00  0
831  00  0
832  03  3
833  04  4
834  00  0
835  00  0
836  02  2
837  69  DP
838  03  03
839  69  DP
```

```
840  05  05
841  69  DP
842  00  00
843  43  RCL
844  00  00
845  69  DP
846  01  01
847  03  3
848  00  0
849  03  3
850  06  6
851  69  DP
852  04  04
853  69  DP
854  05  05
855  81  RST
```

```
3724200612.          00
 669809813.          01
 475842920.          02
 407865360.          03
 407865360.          04
 407865360.          05
 407865360.          06
 339887800.          07
 145920907.          08
-0.08121921          09
    0.0911           10
     0.011           11
 0.01091011          12
```

```
008  16  A'
046  17  B'
081  19  D'
121  81  RST
132  15  E
141  18  C'
419  11  A
731  12  B
771  13  C
778  14  D
801  82  HIR
```



●本体寸法：25.4×17.8×3.3cm

WAR GAMES COMBAT RATIOS.- Bill Beebe reworked that program from v5n4/5p29. The rules are still the same, though. Enter the attacker/defender ratio as an integer.fraction number. For example, a combat ratio of 4 to 1 is entered as 4.1

ALWAYS start the game by entering a number for a seed and press E. after the combat ratio is entered and key A pressed, the program will print two lines on the printer. The first line contains the entered ratio and the program dice throw on one line i.e. by means of OP 04 OP 06 printing. The second line contains the outcome.

As an added feature, the dice subroutine is separately accessible on key A'. Pressing A' will randomly produce a single number from 1 to 6, but in the display ONLY.

The directions are then:

1. Enter a seed number and press E.
2. Enter the combat ratio and press A. Program prints outcome and returns with entered ratio in display.
3. For separate random number generation between 1 and 6, inclusive, press A'.

Program setup:

The program itself is fairly straightforward, with the possible exception of the HIR keycodes. These may be synthesized by STO 82 STO nn, then going back and deleting both STOres. The following data should be stored in the indicated data registers and recorded on bank 4 on the lower side of the mag card, with the program, which should be on the top edge of the card. The program itself is, of course, in bank 1.

REG #	DATA
1	112223333
2	11222333
3	1222233
4	1122233
5	1142443
6	114443

The following is alpha code prepared for direct entry into the print registers.

```
10      1 + 13373713 EE 12 +/- = STO 10
11      1 + 15261735 EE 10 +/- = STO 11
12      1 + 16172117 EE 12 +/- = STO 12
13      1 + 39161735 EE 10 +/- = STO 13
14      1 + 173441   EE 12 +/- = STO 14
15      1 + 13272745 EE 10 +/- = STO 15
16      1 + 1727243024  divide 12 INV *log = STO 16
17      1 + 2913371716  divide 12 INV *log = STO 17
18      1 + 3517373517  divide 12 INV *log = STO 18
19      1 + 133736 EE 8 +/- = STO 19 INV EE
```

See program on next page, please.

ERRATUM,- At the last moment before pouring this issue in concrete, I received a letter from Björn Gustavsson with respect to his SUPERCHECKSUM-FAST MODE program, somewhere else already in this issue. He says it contains an easily correctibel bug: Just load bank 1 and key: GTO 162 LRN CLR RCL 35 LRN 1 WRT and everything will be OK. While running the program an error may be created while executing LOG at step 165. But the result will still be correct and the error condition will be cleared (without the users knowledge.ED) at step 186. He realized his mistake when he took a closer look at Lem Matteson's IMPROVED ALPHA & NUMERIC LIST. Lem used a sequence very similar to Bjorn's. But Lem was correct. Bjorn further thinks that Lem's program could be made more friendly to the user as follows: Suppose you want to list registers 07 through 99. First, you record these registers on mag cards. Then you load the listing program and enter the lower (07) and upper register numbers (99). Then the program writes: ENTER BANK1, ENTER BANK 2, ENTER BANK 3, ENTER BANK 4. Then finally you enter these banks and the program will list all registers.

War Games Combat Ratios, Bill Beebe, listing.

```
000  76 LBL
001  16 A'
002  06  6
003  82 HIR
004  44  44
005  82 HIR
006  14  14
007  59 INT
008  82 HIR
009  54  54
010  85  +
011  01  1
012  95  =
013  92 RTN
014  76 LBL
015  15  E
016  38 SIN
017  82 HIR
018  04  04
019  25 CLR
020  92 RTN
021  76 LBL
022  11  A
023  42 STO
024  07  07
025  59 INT
026  75  -
027  43 RCL
028  07  07
029  22 INV
030  59 INT
031  65  ×
032  01  1
033  00  0
034  75  -
035  05  5
036  95  =
037  94 +/-
038  32 X:T
039  16 A'
040  42 STO
041  08  08
042  73 RC*
043  08  08
044  55  ÷
045  32 X:T
046  22 INV
047  28 LOG
048  52 EE
049  22 INV
050  52 EE
051  95  =
052  59 INT
053  55  ÷
054  01  1
055  00  0
056  95  =
057  22 INV
058  59 INT
059  65  ×
060  09  9
061  00  0
062  85  +
063  07  7
064  08  8
065  95  =
066  69 OP
067  28  28
068  48 EXC
069  08  08
070  69 OP
071  04  04
072  43 RCL
073  07  07
074  69 OP
075  06  06
076  83 GO*
077  08  08
078  43 RCL
079  10  10
080  82 HIR
081  05  05
082  43 RCL
083  11  11
084  61 GTO
085  01  01
086  20  20
087  43 RCL
088  10  10
089  82 HIR
090  05  05
091  43 RCL
092  11  11
093  61 GTO
094  01  01
095  35  35
096  43 RCL
097  12  12
098  82 HIR
099  05  05
100  43 RCL
101  13  13
102  61 GTO
103  01  01
104  35  35
105  43 RCL
106  12  12
107  82 HIR
108  05  05
109  43 RCL
110  13  13
111  61 GTO
112  01  01
113  20  20
114  43 RCL
115  14  .14
116  82 HIR
117  05  05
118  43 RCL
119  15  15
120  82 HIR
121  06  06
122  43 RCL
123  16  16
124  82 HIR
125  07  07
126  43 RCL
127  17  17
128  82 HIR
129  08  08
130  69 OP
131  05  05
132  43 RCL
133  07  07
134  92 RTN
135  82 HIR
136  06  06
137  43 RCL
138  18  18
139  82 HIR
140  07  07
141  43 RCL
142  19  19
143  61 GTO
144  01  01
145  28  28
```

```
 112223333.        01
  11222333.        02
   1222233.        03
   1122233.        04
   1142443.        05
    114443.        06
        1.2        07

1.000013373713     10
1.0015261735       11
1.000016172117     12
1.0039161735       13
1.000000173441     14
1.0013272745       15
1.001727243024     16
1.002913371716     17
1.003517373517     18
1.00133736         19
```

```
        4.1        1
DEFENDER ELIMINATED
        2.1        3
DEFENDER RETREATS
        1.1        4
ATTACKER RETREATS
        1.4        1
ATTACKER RETREATS
        1.4        3
ATTACKER ELIMINATED
        1.2        3
ATTACKER RETREATS
```

M-L SAMPLE.- I have had this TI-88 only a few days and I am hurrying to get this issue to the printer. I have discovered some good and practical programs in the Master Library module and would like to give you a sample of the print out. Not only does this calculator print out (intelligently done this time) everything YOU want it to do from the module, it will also tell you everything about its own status. A real taddler, this one. For example, just press OP 00 and out comes about 3 feet (one meter) of paper with all the op code definitions. The list is somewhere else in this issue. When prompted, it will also tell, in three lanuages if you want, a complete list of its programs. It will also give flag definitions, calculator status and what keys to press to obtain unusual characters in alpha mode.(bottom left corner) Just above it I have done an INV LIST.

Then I tried the random number generator , a print out of which you can see in the lower right hand corner. Next I tried program 12, the Function Evaluator. I entered the function as Lbl A in a 24 step SBR you can see in the center of column 2. It is the function:

$e^{(-X/270)}\sin 2X$ and $-1 \leq Y \leq 1$

In one case I let X run from 0 to 270 degrees and in the second one from 0 to 720 degrees. The printing of the values following the graph can be done at will.

This is by no means the full story. But from what I have seen so far, this TI-88 is a winner. Very flexible programming, for the advanced hacker as well as for the novice or even for the complete "nurd". The prompting is so easy even an ignoramus can make sense of it.(that is a English, German or French speaking one. I have heard that there is a second version of the ML module with Italian, Swedish and Dutch (Flemish)!!!! Thank you, Mr.TI, for that last inclusion and final recognition. You're beautiful, baby!!!

Maurice Swinnen.

M-L Sample

TBL OF CONTENTS

PROGRAM 01
TBL OF CONTENTS

PROGRAM 02
DIAGNOSTIC

PROGRAM 03
FINANCE

PROGRAM 04
MOVING AVERAGES

PROGRAM 05
ROOT FINDER

PROGRAM 06
INTEGRATION

PROGRAM 07
MATRICES

PROGRAM 08
REGRESSION

PROGRAM 09
RANDOM NUMBERS

PROGRAM 10
CODEBREAKER

PROGRAM 11
SORTING

PROGRAM 12
FCN EVALUATOR

INDEX

PROGRAMME 01
INDEX

PROGRAMME 02
DIAGNOSTIC

PROGRAMME 03
FINANCE

PROGRAMME 04
MOY. MOBILES

PROGRAMME 05
RACINES DE f(x)

PROGRAMME 06
INTEGRATION

PROGRAMME 07
MATRICES

PROGRAMME 08
REGRESSION

PROGRAMME 09
NB ALEATOIRES

PROGRAMME 10
TROUVEZ LE CODE

PROGRAMME 11
TRI

PROGRAMME 12
TRACE DE f(x)

INHALT

PROGRAMM 01
INHALT

PROGRAMM 02
RECHNER-TEST

PROGRAMM 03
FINANZEN

PROGRAMM 04
GL.DURCHSCHNITT

PROGRAMM 05
NULLST. f(x)

PROGRAMM 06
INTEGRATION

PROGRAMM 07
MATRIX

PROGRAMM 08
REGRESSION

PROGRAMM 09
ZUFALLSZAHLEN

PROGRAMM 10
CODE-BRECHER

PROGRAMM 11
SORTIEREN

PROGRAMM 12
LOESEN f(x)

FCN EVALUATOR

```
Xmn=              0.
Xmx=            270.
∆x=              30.
Ymn=              -1.
Ymx=               1.
∆y=   .1666666667

 n  |-----+-----|
00          *
01                *
02                *
03          *
04     *
05     *
06          *
07             *
08             *
09          *
 n  |-----+-----|


n                  0.
x                  0.
f(x)               0.

n                  1.
x                 30.
f(x)  .7749535807

n                  2.
x                 60.
f(x)  .6934539327

n                  3.
x                 90.
f(x)               0.

n                  4.
x                120.
f(x)-.5552735048

n                  5.
x                150.
f(x)-.4968850379

n                  6.
x                180.
f(x)               0.

n                  7.
x                210.
f(x)  .3978744348

n                  8.
x                240.
f(x)  .3560336874

n                  9.
x                270.
f(x)               0.
```

FLAG DEFINITIONS
C: PRINTER ON
D: TRACE
E: ERROR
F: HALT ON ERROR

```
000                1.
001                0.
002          -0.0008
003            0.018
004                0.
005                1.
006             360.
007   29.91666667
008              22.
009 -2.444281*10^-18
010  5.785476*10^-18
011             359.
012                0.
```

```
0000 Lbl A
0002 Sto B
0004 DEG
0005 (
0006 Exp
0007 (
0008 B
0009 ±
0010 ÷
0011 2
0012 7
0013 0
0014 )
0015 ×
0016 Sin
0017 (
0018 2
0019 ×
0020 B
0021 )
0022 )
0023 Rtn
```

FCN EVALUATOR

```
Xmn=              0.
Xmx=            720.
∆x=              30.
Ymn=              -1.
Ymx=               1.
∆y=   .1666666667

 n  |-----+-----|
00          *
01                *
02                *
03          *
04     *
05     *
06          *
07             *
08             *
09          *
10          *
11          *
12             *
13               *
14               *
15             *
16          *
17          *
18            *
19              *
20              *
21            *
22            *
23            *
24            *
 n  |-----+-----|
```

RANDOM NUMBERS

X= 131305.2

UNIFORM

```
0.56383
0.71784
0.65648
0.69251
0.20042
0.43124
0.88291
0.55955
0.45292
0.57948
```

ᵃ⁄ₐ 2ND ENTRIES

```
ROW1  F G H I J
ROW2  x̄ Σ x² ⁻¹
ROW3    ! % #
ROW4  μ √ ↑ ↓
ROW5  e ∄ ( ) π
ROW6    ? $ @ ₹
ROW7    > ≥ ≠ <
ROW8  ▪ ° ' " &
ROW9    : , ± ∆
```

ᵃ⁄ₐ SHIFT 2ND

```
ROW1  f g h i j
ROW6  → ᴺ [ ] ≈
ROW7  ← σ ᵃ⁄ₐ ∈
```

```
PGM STEPS:0-0479
REGISTERS:0-207
PAU= 1.5
24 HOUR CLOCK
ALARM= 22:00
CLOCK ALARM OFF
ANGLE MODE:DEG
DECIMAL MODE
TONE ON ERROR
TONE ON CUE
```

NAVAL ARCHITECTURE: J. Huntington Lewis of Norfolk, VA says that the technical journal of the Society of Naval Architecture & Marine Engineers (SNAME) has had several technical papers with the TI-59 as a base.

1. Vol.18, April 1981, pp 188-206, "Calculating the Cross-Curve of Stability using the Hand-Held Programmable Calculator and Printer", Paul B. Cromer.
2. Vol.19, No.2, April 1982, pp 140-158, "A Calculator-based Preliminary Ship Design Procedure", Tim Lyon.
3. Vol.17, No.3, July 1979, pp 260-269, "Calculating Curves of Form Design using the Hand-Held Programmable Calculator", Paul B. Cromer.

NEW PRODUCTS FROM TI: According to the Wall Street Journal of May 27, 1982 Texas Instruments Inc. introduced 67 new products, including a programmable calculator for scientists and engineers. The calculator, the TI-88, will be in the retail stores by Christmas, Texas Instruments said. The company further plans to announce a portable computer, aimed at business and commercial costumers, to complement the calculator.

Among the new products is a $ 120 Magic Wand Speaker, which uses an optical wand to change printed words in a book into spoken words.

AGRICULTURE: Don Loggins in Brooklyn, NY brings to my attention a series of agriculture decisions programs available by subscription from The Programmable Calculator Library, NRAES, Riley-Robb Hall, Cornell University, Ithaca, NY 14843 US. The subscription costs $ 20.00.

There are also agriculture related programs available (cost unknown) from the University of Iowa, Ames, Iowa 50011. Next there a is special Agricultural Decisions module available from ISU Research Foundation, 213 Beardshear Hall, Ames, Iowa 50011. The module costs $ 42.50 mailing included.

The University of Illinois at Urbana IL also has several TI-59 and HP-97 programs available that deal with agriculture.

The University of Nebraska (no address given) has a TI-59 program on Irrigation Scheduling.

And finally, Michigan State University has the so-called Tel Cal programs (about 21 of them) also dealing with the above subject.

Please don't write to me. Write to the addresses given or find out the correct address (and tell me about it, so that I may publish it). Thank you.

SUPERCHECKSUM. I constantly seem to mistake Björn Gustavsson for Lars Hedlund and vice versa. Believe me, I met Lars personally and I have seen pictures of Björn. They don't look alike at all, although they have a few things in common, such as having Swedish as a common mother tongue and both being fanatic calculator nuts. In last issue, v7n6p6 I again assigned Lars as being the author of a Fast Mode program by the above name, when in reality Björn is the one who did it. I think the fault lies with my poor Swedish ability and not being able to translate the fine nuances when I read Programbiten. What I need is a good Swedish-English dictionary.(hint, hint!) Hans Peter Nielsen, the editor of PGM, the Danish TI-59 newsletter, knew what he was doing when he sent me a good Dansk-engelsk Ordbog. (lit. Wordbook) Forget the English-Danish or -Swedish. I need the inverse one.

OPTICS.- In the EOSD journal, Robert T.Pitlak publishes another one of his very good optics programs: Laser Pumping Cavities. It permits rapid estimation of the radiation transfer efficiency of two popular optical geometries for laser cavities. These are the geometries of single- or multiple-elliptical cavities pumped by linear flashlamps, and a diffusely reflecting cylindrical cavity pumped by a helical flashlamp.

The program is for the TI-59 with a PC-100 printer. Equations and sample problems are included, so that it would be easy to translate the program for another type of calculator.

CALCULATOR PUZZLE # 23 from TI: On the right is a reproduction of what I found in, of all places, the German (sensation) magazine STERN. The puzzle presented goes as follows: When will the storm be right above us? 20 Seconds after ten thirty the first stroke of lightning illuminates the sky. Exactly 28 seconds later we hear the accompanying thunder. The next lightning follows at 22.31.02 , with the thunder clap 25 seconds later. Then we have lightning at 22.31.18 and thunder 24 seconds later. And finally we have a lightning flash at 22.32.10 with a thunder clap 20 seconds later. The ad further says that with the TI-54 (a non-programmable but fantastic calculator, check it out, you'll be surprised) this problem is just a cinch.

If you put the page upside down you'll be able to read (provided you understand German) that the solution to this problem is: the storm will be exactly above us at 22.36.46.

Who writes a program to compute the above in an easy-to-use way? If you are unfamiliar with physics: Light travels at 300,000,000 meters per second or 186,000 miles per second. Sound on the other hand is much slower at 340 meters per scond or 1100 feet per second.

PROGRAMMING PUZZLE, Besides the one above I would like to present you with one that depends entirely on your programming savvy. (no physics formulas needed) It has been proposed by Myer Boland from Englishtown, New Jersey. Myer simply states: "Press A and the calculator flashes 1, 2, 3, 4, 5....ad infinitum. No numerical keys are to be used, with the exception of the PI key. That one may be used. "
Myer gives several of his discoveries. One of his shorter ones, and reasonbly fast one is this: PAU + CE DIV CE = LBL A CMs RST

These little programming puzzles and their solutions are not earth shattering but constitute great fun at times and surely show the user's knowledge about what exactly each key does.

MODULE SELECTOR.- Bill Manley of Graham Magnetics in North Richland Hills, Texas reports that he has been using both the manual and the automatic module selectors as noted in v5n8p3. Bill says that he has found it relatively easy to connect both selectors to one, single TI-59/PC100 combination and make a fairly neat job of it. The automatic selector goes best into the #2 slot of the manual unit. Three more modules can then be installed in the manual unit, and four more in the automatic one. When the calculator is first turned on, the selection of the automatic unit defaults to the "0" module, so that there are effectively four modules to be selected manually, without need of program control. But program control can additionally be used to select the three additional modules contained in the automatic unit.

Installation is done by first loading the auto module per the manuafacturer's instructions. Close the auto module. Next we do a bit of "surgury" (Bill used an X-Acto #11 knife blade.How cruel! Ed.) Fit the auto connector into the #2 slot of the manual unit and carefully cut away a notch on the cover to let the cable come out. Load the other three modules into the manual unit and close the cover. Attach the manual unit to the printer per the manuafacturer's instructions. Now close the manual unit and mark the door where the cable should come out. A notch can be cut in the OUTER part of the door ONLY to let the cable come out. It is not necessary to cut on any part of the printer at all, nor on any other part of the manual unit. The manual unit binds a bit when it is being closed, but frees up just before it is closed completely. Now install the TI-59 on the printer. Lastly, run diagnostics on all the modules to see if they are connected correctly and you are ready to go!

DRAW POKER FOR THE TI-59.- Under the title DESK TOP WONDERS Byte magazine sometimes presents calculator programs. About a year ago some of the programs were, to put it mildly, "horrendous": no sign of any reviewing by competent programmers whatsoever. I wrote them a few times, pointing out that we would be glad to do the reviewing for them. No answer was ever received. Now I see with with great satisfaction that the programs published have improved one hundred percent.

This program is a long one: 480 steps and 53 data registers, almost the entire capacity of the calculator! The author, Lee Boyle of Tucson, Arizona is unknown to me. He did a superb job: nice, concise routines, all subroutines at the beginning (for speed) and the whole thing easy and friendly to the user. No printing, though, just for calculator-only use.

You can find this program in Byte, July 1982 on pages 434 through 440.

PPC CHECKBOOK,- Panos Galidas wrote a program by that name a few years ago. It is now available from PPX under # 148007. In my opinion it is THE program you'll need to keep your checkbook up to date.

Now Panos has improved it even more. To make the new version available Panos will send you the new listing and a new set of instructions including a sample data file for $ 3.00 check or postal money order to Panos Galidas, 150 Monroe Street # 302, Rockville MD, 20850. If you think it would eat into your precious time to key in the new listing, Panos will record the program and the sample data file on mag cards, provided you send him an extra $ 2.00. At a total of $ 5.00 this is a real bargain. Even if you never even used his first version it is possible to learn how to use it from the instructions Panos will send you.

Please bear in mind that this program requires the use of the Math/Utility module.

ERRATUM- In last issue, v7n6p3 we published the BLACKJACK TUTOR program by the Snow brothers. Many of you wrote me, some even called me, to say that either they couldn't make it work, as the thing refused to be initialized, or they found that after pressing E and entering a seed, pressing R/S would make it work as a charm. Needless to say, I love these latter people, as that was the typo I managed to get in this time. Some of you send me long letters, saying how it could be rewritten so as to make it work according to the original instructions. Thanks, but no thanks. Please, initialize by pressing E. Then enter a seed number and press R/S. It works. Sorryabout that, fellas.
