TI PPC NOTES

NEWSLETTER OF THE TI PROGRAMMABLE CALCULATOR CLUB

P.O. Box 1421, Largo, FL 34294

Volume 8, Number 3 May/June 1983

The BIG NEWS is the Texas Instruments Compact Computer 40 (CC-40). Maurice Swinnen and I had received engineering models some time ago for evaluation. As a result this issue of TI PPC Notes contains our preliminary impressions together with some sample programs. There seems to be an emphasis on scientific applications as evidenced by thirteen (sometimes fourteen) digit arithmetic, trigonometric functions such as arcsin and arccos, use of trigonometric arguments in radians, degrees, or grads, and the like. A calculator mode is provided which has an unexpected quirk for a TI machine (see page 5). Example speed checks show that the CC-40 is much faster than the TI-59. The keyboard is small--too small to touch type, but large enough to not feel cramped. The CC-40 is not a pocket computer--but then neither are most other so-called "pocket" computers, unless one is talking about the pockets in the winter overcoats of Russian infantrymen. The announcements of the peripherals describe a complete capability including Wafertape™ drives for recording, RS-232 interfaces for printing, and even a video interface which will circumvent one of the major limitations of the baseline CC-40, namely the single line display. It is downright difficult to debug programs without a printer and only a single line display. While the CC-40 is now available from retailers the peripherals are not, at least not in the Tampa Bay area.

The second news item is that we are to get a new TI programmable calculator after all. TI News Release CGG-115 describes the TI-66. Many of the features are similar to those we are used to with the TI-58/59, but some are new such as merged code, automatic insert during edit (like that in the TI-57LCD, I presume), etc. The TI-66 will interface with the PC-200 printer.

This issue also contains relatively extensive treatment of the accuracy of the TI-59, other hand-held programmables, the CC-40 and other small computers. The impetus for this was the large number of favorable comments on the accuracy discussions in V8N1 and V8N2.

Programs in this issue include a new game by Dave Lane, an exact factorial program by new member Peter Messer which will calculate up to 461 factorial, and the Science et Vie program which finds up to 1287 digits of pi. Finally, Maurice Swinnen has provided a JIVE TURKEY program for the CC-40.

ERRATUM - Programming Workbook - Maurice Swinnen offered these workbooks to members for one dollar postage in V8N1P10. You should change the listing for Exercise 10-1 on Page I-27 before giving the workbook to a beginner. The problem arises because the workbook shows a correct sequence on page I-27 which is identical to the edited sequence shown as a solution on page A-8. Therefore, the edit sequence shown on page A-8 does not apply to the problem as listed on page I-27. An initial program sequence which is incorrect and which can be changed to a correct sequence by the edit keystrokes shown on page A-8 is listed at the right.

```
000  43  RCL
001  01   01
002  65   ×
003  53   (
004  43  RCL
005  02   02
006  75   -
007  53   (
008  53   (
009  01   1
010  75   -
011  53   (
012  01   1
013  85   +
014  43  RCL
015  03   03
016  54   )
017  45   Y^x
018  43  RCL
019  03   03
020  54   )
021  54   )
022  95   =
023  91  R/S
```

ERRATUM - George Thomson noted that I had misspelled the name of the author of "The Ambidextrous Universe" in the discussion of palindromic numbers on V8N2P10. The author's name is Martin Gardner, not Gardiner as in the discussion. My apologies to Mr. Gardner.

TI-57 LISTING ON A TI-59 - Frederic De Mees writes that when using this program to generate an INV FIX statement, the program waits for a numeric entry after the FIX key is pressed. The user need only enter a -1 in order to have a blank field printed.

DIGIT REVERSER PUZZLE - Frederic De Mees notes that Reinhold Patzer's digit reverser solution in V7N10P3 will not work unless the t register has been cleared by some earlier action. Changing the CE at location 003 to a CP will eliminate that problem. Reinhold's solution was originally published in V3N3P4 of 52 Notes, not in V3N4P4 as indicated on V7N10P3.

FAST-GRAFIK-3-D-PLOT - This program by Peter Poloczek appeared in V8N2P11. Letters from various members show that it would have been helpful if I had included an example printout of the function used for the plot on V8N2P14 and for the lower plot on the cover. One possible function is at the right. Note that the function begins at location 473 and ends with a GTO 288 sequence.

```
473  53   (
474  01   1
475  02   2
476  55   ÷
477  53   (
478  01   1
479  85   +
480  43  RCL
481  40   40
482  33   X²
483  85   +
484  43  RCL
485  41   41
486  33   X²
487  54   )
488  54   )
489  61  GTO
490  02   02
491  88   88
```

V8N2P14 also incorrectly identifies which function goes with which plot on the cover. The upper plot was done with the function

$$z = (x^2 - y^2)/4$$

while the lower plot on the cover was done with the function

$$z = 12/(1 + x^2 + y^2)$$

INVESTIGATION OF QUIRKS - Palmer Hanson. Newcomers to TI PPC Notes may wonder at the amount of coverage given to the investigation of "quirks". In the article "Psychology and the Programmable Calculator" (PPX Exchange, November/December 1980) the author J. M. Gallego, who is also a member of our club, observed that PC owners seemed to fit into various categories. He defined one category as

> "... a group that likes to push all the keys of their PC at the same time or to follow illogical steps to see what turns up. To their amazement, nothing important usually happens. The origin of this technique is due to the astounding discoveries made by our German colleagues of the 'Display' newsletter. Some of their ideas have been translated and published in 'TI PPC Notes' ..."

I confess to being an addicted member of that group. Admittedly, apparently unimportant things turn up. One example was proposed as a trivia award candidate in V2N8P6 of 52 Notes. Dallas Egbert had found that if you pressed *read on an SR-52, and then simultaneously pressed the keys B, INV, sin, STO, EE, 4, and 0 then the drive motor would turn on. In V5N8P3 of TI PPC Notes I reported that a synthesized IND (code 40) after a GTO (code 61) would act the same as a GO* (code 83); that is, a GTO-IND-XX (61-40-XX) acts the same as a GO*-XX (83-XX). To synthesize such a sequence, say a GTO-IND-10, you may use the key-in sequence GTO-0-4-0-BST-BST-Del-SST-E' . Similar equivalences apply for the other merged codes which use indirect addressing. I challenged the readers to find a use for that function. To my knowledge no one has reported a use for either Dallas Egbert's method for turning on the drive motor, or for the GTO-IND-XX sequence.

But triviality is too often an individual perception. In the July/August 1980 issue of PPX Exchange I reported that the use of the sequence EE-INV-EE to truncate the guard digits also will change the display register to a live state. At the time I didn't see any use for that "quirk". But in the January/February 1981 issue of PPX Exchange member Milton Cragg reported that the technique could be useful to correct a mistake in a number being entered without CE'ing and starting from scratch. Many other highly useful unannounced characteristics of the TI-59 have been discovered:

* The use of the Pgm-nn-SBR-nnn sequence to enter a library program at any location.
* The ability to use the Dsz function on any register other than register 40.
* The use of the hierarchy registers.
* Magnetic card reading under program control.
* The extended print code table.
* Additional labels.
* Transfer to the middle of a merged code to obtain a different effect.

INVESTIGATION OF QUIRKS (cont)

* Fast mode entry using the Pgm-02-SBR-239-9 method with the Master Library module installed.
* High resolution graphics through the use of hexadecimal code h25 (also called pseudocode in PPX Exchange).
* Fast mode entry using hexadecimal code h12. This technique does not require that the Master Library be installed, and does not clear all memory as with the Pgm-02-SBR-239-9 method.
* Branching from the keyboard as illustrated in the Super TI-59 Test by Dejan Ristanovic (V7N9P9).

Current investigations of "quirks" center on the effects of other hexadecimal codes. The impetus for continuing the search was well described by Maurice Swinnen in V5N6P5:

> "... You might have asked yourself why I am so interested in these seemingly, useless quirks. May I remind you, however, that thanks to the Pgm-02-SBR-239 quirk we are able to fill one bank easily with SST,s, to be used as the best diagnostic we ever had. Also it permits us to write a row of HIR's anywhere in a program, and using Richard Snow's trick, perform any HIR function from the keyboard, using the SST key as the HIR button. Even an IND HIR is possible now. So keep on searching!"

The first practical application of fast mode, a fast mode calendar, was published in the following issue of TI PPC Notes. Branching from the keyboard was demonstrated later that year, and high resolution graphics was demonstrated in 1981. So I repeat Maurice's admonition--keep on searching!

SYNTHETIC CODES ON A TI-57 - In V8N2P23 Dejan Ristanovic discussed the use of the sequence EXC SST on a TI-57 to yield synthetic codes. While reading back issues of 52 Notes for another reason I discovered that a similar effect was discussed by Steve Halko in August 1978 (V3N8P1 of 52 Notes) under the heading "Suppressed Operand Instructions". Steve reported that the key-in sequence GTO SST 2 produced a 51 (GTO) at one step and a 2 at the next step. On encountering this sequence a running program makes an unconditional branch to the first numeral 2 appearing in the program. Such behaviour effectively doubles the number of available labels for unconditional branches. But the effects with SBR SST were more complex.

BACK ISSUES OF 52-NOTES - Richard Vanderburgh reports that he will continue to make back issues of 52 Notes available at $1.50 each in the US, $2.00 each abroad. Write to:

Richard Vanderburgh
9459 Taylorsville Road
Huber Heights OH 45424

TI-58/59 users would be interested in the series from Volume 2 Number 6 through Volume 4 Number 3, a total of 22 issues with 130 typewritten pages. Earlier issues were devoted to the SR-52 and SR-56.

A CC-40 QUIRK - Palmer Hanson. The second chapter of the TI Compact Computer User's Guide describes how to use the CC-40 as a calculator. The discussion of chain calculations on page 2-8 cautions "...A loss of accuracy occasionally results when you chain calculations. See Appendix F for accuracy information. ..." The discussion of accuracy in Appendix F begins with a discussion of the 5/4 rounding technique which will remind the TI-58/59 user of a similar discussion on page C-1 of Personal Programming. As with the TI-58/59 the CC-40 uses a minimum of 13 digits to perform calculations and rounds the results to 10 digits for the normal display format. Actually, some calculations are carried to 14 digits as in the example on page F-1:

2/3 = .66666666666667 and 1/3 = .33333333333333

2/3 - 1/3 - 1/3 = .00000000000001 which is dispalyed as 1.E-14

Note that both fractions yield fourteen digit values. Furthermore, the fraction 2/3 yields a 7 in the fourteenth or least significant place. The TI calculators have typically yielded a 6 in the least significant place of the display register in response to the sequence 2 DIV 3 = . The fact that the TI calculators truncated to the display register was sometimes useful. An example appeared in my article "There's Gold in Those Guard Digits" in the May/June 1982 issue of PPX Exchange, where I described the use of the truncation feature to implement an effective integer function when a thirteen digit integer was divided by a small integer such that the quotient still had a thirteen digit number to the left of the decimal point.

Now if we alter the sequence above slightly in order to view the intermediate result, say to the sequence

2 / 3 ENTER - 1 / 3 - 1 / 3 ENTER

then the result in the display will be 3.334E-11 . Insertion of = before each ENTER will not change the result. Investigation will reveal that the different result occurs because the ENTER command causes the calculator mode to truncate to the display value. TI-58/59 users will recognize this effect as being similar to the use of an EE-INV-EE sequence to truncate to the display value. If one performs the sequence

2 DIV 3 = EE INV EE - 1 DIV 3 - 1 DIV 3 =

with a TI-58 or TI-59 the result will be 3.34E-11 where the difference from the CC-40 result above is due to the use of fourteen digits by the CC-40 and thirteen digits by the TI-58/59. This effect of the interruption of a chain calculation to display an intermediate result is an important difference between the use of the CC-40 in the calculator mode and the use of TI calculators. The equivalent sequence in a BASIC mode does not yield the truncation effect. The sequence

```
Y = 2/3
PRINT Y
X = Y - 1/3 - 1/3
PRINT X
```

yields 1.E-14 in the display. We will discuss other aspects of accuracy of the CC-40 in future issues.

PALINDROMIC NUMBERS - Albert Smith of Brooksville, Florida became fascinated with Myer Boland's palindromic number generator (V8N2P10). He modified Myer's program slightly so that it would increment the input integer, print the input integer, perform the sequence, print the palindromic number, and print the number of steps. He started with the input value 1, and let his calculator run until he had tested the integers from 1 to 1900. That took a full roll of printer paper--a monumental effort. He found 24 input integers that yielded a printout in scientific notation which were decoded as follows:

89	8813200023188	24 steps
98	8813200023188	24 steps
187	8813200023188	23 steps
286	8813200023188	23 steps
385	8813200023188	23 steps
484	8813200023188	23 steps
583	8813200023188	23 steps
682	8813200023188	23 steps
781	8813200023188	23 steps
869	8813200023188	22 steps
880	8813200023188	22 steps
899	133697796331	17 steps
968	8813200023188	22 steps
989	89540004598	19 steps
998	133697796331	17 steps
1297	8813200023188	21 steps
1387	8813200023188	21 steps
1477	8813200023188	21 steps
1567	8813200023188	21 steps
1657	8813200023188	21 steps
1747	8813200023188	21 steps
1798	89540004598	18 steps
1837	8813200023188	21 steps
1888	89540004598	18 steps
1897	133697796331	16 steps

The preponderance of answers which were 8813200023188 is striking. Albert also verified that 196 did not reach a palindromic number within the single precision range of the TI-59. He also found the following other input integers which did not reach a solution:

196		689	790	978	1495	1587	1765	1857
295	493	691	879	986	1497	1675	1767	
394	592	788	887		1585	1677	1855	

After reading of Albert's result I decided to test the numbers which did not arrive at a solution on the TI-59. I wrote BASIC programs for the CC-40 and for the Radio Shack Model 100. Testing showed that not one of the input integers which failed to reach a palindromic number within the range of the TI-59 would reach a palindromic number with over 100 steps. The range was limited to about 140 steps by the normal string length of those computers. The BASIC programs are presented on page 24 together with a discussing of execution speeds.

MORE PALINDROMES - Joseph Hansen, Myer Boland, and George Thomson. On V8N2P10 I posed a challenge for palindromic words as: write down all the palindromic words you can. Give yourself one point for a one letter word, four points for a two letter word, nine points for a three letter word, etc., where the value of each word is the square of the number of its letters. No proper names allowed. I claimed a score over 600. That paltry score was outclassed by Myer Boland's 992 and by Joseph Hansen's 1529. It seems that Joseph Hansen has been a long time collector of palindromes. Here is a composite list:

a	aa	aha	mum	anna	alula	denned	deified	semitimes
I		bib	nun	boob	carac	millim	repaper	
O		bob	pap	deed	civic	pullup	reviver	
		bub	pep	kook	dewed	redder	rotator	
		dad	pip	noon	etete	retter	shamahs	
		did	pop	otto	finif	succus	wow-wow	
		dud	pup	peep	kayak	terret		
		eke	tat	poop	lemel	marram		
		ere	tit	sees	level			
		eve	tot	toot	madam			
		ewe	tut		minim			
		eye	ulu		radar			
		gag	vav		refer	sohos		
		gig	waw		rever	solos		
		mom	wow		rotor	stets		
					sagas	tebet		
					sexes	tenet		
					shahs	ululu		

You aren't going to find those in just any dictionary. For example, Joseph writes that "semitimes" was listed in Webster's New International Dictionary (Unabridged), Second Edition. But it isn't in the third edition. Finally, George Thomson proposes the sentences "No, it is opposed. Art sees trade's opposition."

DELIVERY BY BULK RATE - The Post Office doesn't promise to expedite bulk rate mail; however, the variations in delivery seem almost ridiculous. I placed the first issue for 1983 in the mail on February 24. Some individuals had their copy by March 1, but several subscribers in California reported delivery as late as March 15. The second issue was placed in the mail on April 14. One California subscriber did not receive his copy until May 4. So, if you are a bulk rate subscriber, particularly if you are a bulk rate subscriber living in California, be patient.

HP ON ALGORITHM ACCURACY - The May 1983 issue of the Hewlett-Packard Journal contains a well written discussion of the accuracy of various calculator algorithms in a description of the HP-15 entitled "Scientific Pocket Calculator Extends Range of Built-In Functions". The authors are E. A. Evett, P. J. McClellan, and J. P. Tanzini, all of HP. Worth reading. I am in the process of working through the examples on my TI-59.

1287 DIGITS OF PI - Renaud de La Taille. This is the program which was mentioned in V8N2P4. Pierre Flener obtained permission from Science et Vie to reprint the program in TI PPC Notes. In his letter to Pierre the author stated that he couldn't have achieved the result without hints given by M. Colmont and M. Brombeck.

The program is capable of delivering 1287 digits of pi--thirteen digits per register times 99 registers, with register 00 reserved for Dsz control. To obtain the entire 1287 digits requires a run time of about twenty-four and one-half days. The program also provides options for obtaining lesser numbers of digits with substantial reductions in run time. To run the program:

(1) Enter the program.

(2) Press GTO-125-LRN and see a 3 in the display. This value will permit running the program for up to 507 digits. You may change this value for other limits according to the following table:

Contents of Location 125	Number of Places
Nop	117
1	247
2	377
3	507
4	637
5	767
6	897
7	1027
8	1157
9	1287

You should set this value to the lowest value consistent with the number of digits you wish to calculate since execution is faster as the value in location 125 becomes smaller. This is because the program only carries the calculations to the precision required to obtain the number of places indicated in the table. In other words, if you only want fifty digits, but put the value at location 125 at 9, then the program will calculate as if it needed to deliver all 1287 digits proceeding at the same rate as if you wanted that many digits. After setting the appropriate value press LRN to return to program control.

(3) Press 10 Op 17 Cms CLR

(4) Calculate the number of iterations required from the formula

N = ((Number of places/log(2)) + 1)INT

As an example, if you want 100 digits then N will be 333.

(5) Press RST R/S to begin program operation. You can monitor the progress of the calculation by observing the value of the remaining iterations as displayed by the Pause command at location 008.

(6) The program stops with 314159.2654 in the display, where you will recognize those digits as the leading digits of pi.

<u>1287 DIGITS OF PI</u> (cont)

(7) If you do not have a printer you can read out the solution with the following procedure:

(a) Starting with the 314159.2654 in the display,

(b) Press - 2nd INT and record the first six digits,

(c) Press = and see the next seven digits preceded by 0.

(d) Press D and repeat steps b, c, and d as required.

(8) If you have a printer you may record the solution on magnetic cards and use one of the thirteen digit list programs to print the solution. The first thirteen digits will be found in data register 01 with subsequent digits in groups of thirteen in the following registers.

<u>Sample Execution Times</u>:

Contents of Location 125	Number of Places Wanted	N from Step 4	Execution Time
Nop	20	67	52.5 minutes
Nop	100	333	4h 33m
3	500	1661	3d 18h 12m
9	1287	4276	24.55 days

By timing the interval between the display of the remaining number of iterations you can obtain a good estimation of run time and completion time.

<u>Program Listing</u>:

```
000 82 HIR    027 75  -     054 74 SM*    081 82 HIR    108 65  x     135 67  67
001 06  06    028 22 INV    055 00  00    082 16  16    109 32 X:T    136 03  3
002 69 OP     029 59 INT    056 97 DSZ    083 54  )     110 82 HIR    137 82 HIR
003 20  20    030 72 ST*    057 00  00    084 59 INT    111 16  16    138 56  56
004 25 CLR    031 00  00    058 00  00    095 65  x     112 95  =     139 02  2
005 29 CP     032 85  +     059 12  12    086 32 X:T    113 32 X:T    140 82 HIR
006 82 HIR    033 32 X:T    060 02  2     087 82 HIR    114 55  ÷     141 66  66
007 16  16    034 59 INT    061 82 HIR    088 16  16    115 01  1     142 52 EE
008 66 PAU    035 65  x     062 46  46    089 85  +     116 52 EE     143 05  5
009 67  EQ    036 82 HIR    063 01  1     090 32 X:T    117 07  7     144 44 SUM
010 01  01    037 16  16    064 82 HIR    091 63 EX*    118 95  =     145 01  01
011 49  49    038 95  =     065 36  36    092 00  00    119 74 SM*    146 61 GTO
012 73 RC*    039 55  ÷     066 32 X:T    093 22 INV    120 00  00    147 00  00
013 00  00    040 01  1     067 69 OP     094 59 INT    121 43 RCL    148 04  04
014 32 X:T    041 52 EE     068 20  20    095 95  =     122 00  00    149 58 FIX
015 55  ÷     042 06  6     069 65  x     096 65  x     123 32 X:T    150 07  07
016 01  1     043 75  -     070 01  1     097 01  1     124 85  +     151 42 STO
017 52 EE     044 59 INT    071 52 EE     098 52 EE     125 03  3     152 00  00
018 07  7     045 65  x     072 06  6     099 07  7     126 09  9     153 76 LBL
019 85  +     046 32 X:T    073 85  +     100 75  -     127 67  EQ    154 14  D
020 73 RC*    047 01  1     074 73 RC*    101 53  (     128 01  01    155 69 OP
021 00  00    048 95  =     075 00  00    102 24 CE     129 36  36    156 20  20
022 22 INV    049 65  x     076 59 INT    103 55  ÷     130 65  x     157 73 RC*
023 59 INT    050 01  1     077 75  -     104 82 HIR    131 00  0     158 00  00
024 65  x     051 52 EE     078 53  (     105 16  16    132 95  =     159 91 R/S
025 82 HIR    052 06  6     079 24 CE     106 54  )     133 61 GTO
026 16  16    053 95  =     080 55  ÷     107 59 INT    134 00  00
```

We have not yet obtained a translation of the text describing the theory supporting this method. Is there an accomplished translator among the readers?

EXACT FACTORIALS FOR THE TI-59 - Peter Messer of Milwaukee, Wisconsin submitted this "number cruncher" which will calculate exact values for either large n! or large n!/m!. The program performs the required multi-precision arithmetic and accumulates the answers in blocks of ten integers each. The lowest ten digits are stored in data register R01, the next higher ten digits are stored in R02, etc. But, as a block of ten lower order zeroes is accumulated it is converted to an exponent notation and removed from the normal sequential storage. The maximum limits for the program are:

* The number of places preceding blocks of ten zeroes is 920 (92 registers times ten places per register).
* The value for n must be less than 1000.
* The largest factorial which can be obtained is 461! which fills 92 blocks with an exponent of 110, ≈ 3.0717746 E 1029 .

To run the program:

(1) Enter the program.

(2) Press A to initialize. See 1.10 in the display.

(3) To find n! :

(a) Enter the first factor and press B. You may use either a 1 or a 2.

(b) Enter the second factor and press C. You may use a 2 if you used a 1 in step 2.a. or a 3 if you used a 2 in step 2.a.

(c) Enter n and press D. On completion of the calculation the highest order block will appear in the display. To display the remaining blocks press R/S for each remaining block. A "-0" in the display indicates that all the blocks have been displayed. To display the number of trailing zeroes press E. The exponent will always be a multiple of 10.

(4) To find n!/m! , first initialize as in step 2 above. Then

(a) Enter m+1 and press B.

(b) Enter m+2 and press C.

(c) Enter n and press D. On completion of the calculations read out the solution as in step 3.c above.

Without reinitializing the user may calculate solutions for N greater that the previously entered n. For example, to find N! after having found n!, do not initialize but

(a) Enter n+1 and press C.

(b) Enter N and press D. On completion of the additional calculations read out the new solution as in step 3.c. above.

Similarly, to find N!/m! after having found n!/m! , do not initialize but enter n+1 and press C. Then, enter N and press D. Read out the solution as in step 3.c. above.

The program appears on the next page.

EXACT FACTORIALS FOR THE TI-59 (cont)

Program Listing:

```
000  76 LBL     023  69 OP      046  43 RCL     069  63 EX*     092  01  01     115  00  00
001  11  A      024  20  20     047  93  93     070  00  00     093  17  17     116  38  38
002  47 CMS     025  61 GTO     048  95  =      071  69 OP      094  01  1      117  73 RC*
003  01  1      026  00  00     049  67 EQ      072  20  20     095  44 SUM     118  00  00
004  00  0      027  20  20     050  00  00     073  61 GTO     096  98  98     119  91 R/S
005  69 OP      028  76 LBL     051  76  76     074  00  00     097  61 GTO     120  97 DSZ
006  17  17     029  13  C      052  55  ÷      075  40  40     098  00  00     121  00  00
007  01  1      030  42 STO     053  43 RCL     076  69 OP      099  38  38     122  01  01
008  42 STO     031  98  98     054  99  99     077  30  30     100  43 RCL     123  17  17
009  00  00     032  91 R/S     055  95  =      078  43 RCL     101  94  94     124  43 RCL
010  52 EE      033  76 LBL     056  42 STO     079  00  00     102  42 STO     125  00  00
011  01  1      034  14  D      057  97  97     080  67 EQ      103  00  00     126  94 +/-
012  00  0      035  42 STO     058  59 INT     081  01  01     104  00  0      127  91 R/S
013  42 STO     036  95  95     059  42 STO     082  00  00     105  63 EX*     128  76 LBL
014  99  99     037  01  1      060  93  93     083  42 STO     106  00  00     129  15  E
015  22 INV     038  42 STO     061  43 RCL     084  94  94     107  97 DSZ     130  43 RCL
016  52 EE      039  00  00     062  97  97     085  43 RCL     108  00  00     131  96  96
017  91 R/S     040  73 RC*     063  22 INV     086  95  95     109  01  01     132  65  ×
018  76 LBL     041  00  00     064  59 INT     087  75  -      110  05  05     133  01  1
019  12  B      042  65  ×      065  65  ×      088  43 RCL     111  01  1      134  00  0
020  72 ST*     043  43 RCL     066  43 RCL     089  98  98     112  44 SUM     135  95  =
021  00  00     044  98  98     067  99  99     090  95  =      113  96  96     136  91 R/S
022  91 R/S     045  85  +      068  95  =      091  67 EQ      114  61 GTO
```

Sample Execution Times:

n	time
10	30 seconds
20	82 seconds
34	3 min 15 sec
50	6 min 37 sec
100	25 min 30 sec
200	

Users who have a printer available may wish to change the program to provide an automatic printout. A suggested modification starting at location 119 is shown below with illustrations of representative printouts. Note in the second line of the 34! printout that leading zeroes in each block are not indicated.

Listing

```
117  73 RC*     130  69 OP
118  00  00     131  04  04
119  99 PRT     132  43 RCL
120  97 DSZ     133  96  96
121  00  00     134  65  ×
122  01  01     135  01  1
123  17  17     136  00  0
124  01  1      137  95  =
125  07  7      138  69 OP
126  04  4      139  06  06
127  04  4      140  98 ADV
128  03  3      141  91 R/S
129  03  3
```

34!

```
 295232799.
 396041408.
4761860964.
3520000000.
         0.     EXP
```

70!

```
         1.
1978571669.
9698917960.
7278372168.
9098736458.
9381425464.
2585755536.
2864628009.
5827898453.
1968000000.
        10.     EXP
```

An error will occur if a block of zeroes is encountered which is not the last block of ten digits, or if a factorial greater than 461 is entered. The block of zeroes condition is believed not to occur.

EXACT FACTORIALS FOR THE TI-57 - Reginald van Genechten of Belgium. At age 16 Reginald is one of our younger contributors. His program for the TI-57 (not LCD) will obtain factorials of up to 40 digits. That accomodates up to 34! . On the previous page the execution time for Peter Messer's TI-59 program for 34! was listed as 3 minutes 15 seconds. This TI-57 program will obtain 34! in just 2 minutes 30 seconds. To run the program:

(1) Enter the program and initialize by pressing RST INV C.t .

(2) Enter n and press R/S. When calculations are complete 1.-08 will appear in the display.

(3) Read out the solution from data registers 1 through 5 with the highest digits in R5 and the lowest digits in R1. For example, for 34! :

RCL 5	2952327
RCL 4	99039604
RCL 3	14084761
RCL 2	86096435
RCL 1	20000000

(4) Press RST INV C.t to initialize for a new entry.

Program Listing:

```
00  66        X=T      13  39 4      PRD 4   26 -49    INV INT     38 -39 1 INV PRD 1
01  01          1      14  39 5      PRD 5   27  38 3      EXC 3   39 -39 2 INV PRD 2
02  32 0    STO 0      15  33 1      RCL 1   28  61 9      SBR 9   40 -39 3 INV PRD 3
03  08          8      16 -49    INV INT     29  34 4      SUM 4   41 -39 4 INV PRD 4
04  84        +/-      17  38 1      EXC 1   30  33 4      RCL 4   42 -39 5 INV PRD 5
05 -18    INV LOG      18  61 9      SBR 9   31 -49    INV INT     43  81         R/S
06  32 1    STO 1      19  34 2      SUM 2   32  38 4      EXC 4   44  86 9     LBL 9
07  32 6    STO 6      20  33 2      RCL 2   33  61 9      SBR 9   45  49         INT
08  86 8    LBL 8      21 -49    INV INT     34  34 5      SUM 5   46  55           x
09  33 0    RCL 0      22  38 2      EXC 2   35  56        DSZ     47  33 6     RCL 6
10  39 1    PRD 1      23  61 9      SBR 9   36  51 8      GTO 8   48  85           =
11  39 2    PRD 2      24  34 3      SUM 3   37  33 6      RCL 6   49 .61     INV SBR
12  39 3    PRD 3      25  33 3      RCL 3
```

THE TI-66 PROGRAMMABLE CALCULATOR - A new programmable calculator for the advanced engineering student or professional who needs sophisticated full keystroke programming capabilities in a low cost machine is being announced by TI at the Consumer Electronics Show in Chicago June 5-8. The TI-66 calculator offers a wide horizontal case, a 10 digit LCD display, AOS™ , access to more than 500 merged program steps and more than 170 built-in scientific, engineering and statistical functions. The battery operated calculator provides users with 72 useful labels within each program, as well as 10 user defined label keys. Ten user flags and six levels of subroutines are available, and nine sets of parenthesis allow for eight pending operations. Ten test registers are directly accessible for looping, incrementing, and decrementing. Complete program editing capabilities include automatic insert, delete, single-step, back-step, and no-hyphen operation. When used with the TI PC-200 Thermal Printer the TI-66 is able to provide hard copy through a low-cost, battery-powered printer that is controlled from the calculator keyboard. ... The suggested retail price for the TI-66 is $70.00. (TI News Release CG-115)

So we are to have a new programmable after all. The photo accompanying the news release seems to indicate that it will be truly shirt pocket size--clearly a competitor with the HP-1X series. The PC-200 has already been available for use with the TI-BA-55.

MORE ON NUMERICAL PRECISION - Palmer Hanson. In V8N2P3 I introduced the subject of the comparative numerical precision of various hand-held calculators. This is not a new issue. V2N4P2 of 52 Notes reported that a 5 X 5 determinant and inverse matrix problem had been accepted as an appropriate subject for the so-called "friendly competition" between users of TI and HP programmable hand-held calculators. V2N12P5 of 52 Notes reported that Hal Brown had obtained an HP version which would maintain five place accuracy for ill-conditioned matrices, but that the ML-02 program on the TI-59 would maintain eight place accuracy. Editor Richard Vanderburgh went on to speculate that "...Hal's (HP program) and ML-02 both employ pivoting, and I would expect ML-02 results to be similar to Hal's if produced with 10-digit precision."

In 1981 the relative precision of ten digit and thirteen digit calculators received recurring attention in the pages of BYTE. The comment which incited the discussion appeared in the article "The HP-41C: A Literate Calculator" by Brian Hayes in the January 1981 issue of BYTE, page 136. In an otherwise complimentary article the author commented:

> "...The most fundamental defect in the architecture of the HP-41C, inadequate numerical precision, is a serious flaw indeed. ..."

The author continued with the example that $(\sqrt{2})^2$ is evaluated by the HP-41C as 1.999999999, and that for some chain calculations the inaccuracy is more severe and 2 or 3 low order digits may become incorrect. The ensuing discussion in the Letters column of BYTE included input from both HP and TI proponents. Unfortunately, the introduction of such non-technical, emotionally charged words such as "absurd", "funny arithmetic", and "braggadocio" did little to resolve the differences. I took another approach. I ran tests with my TI-59. I also performed the equivalent tests with the SR-51A, the HP-67, the HP-41C, and the HP-11. These calculators fall into two distinct classes: those manufactured by Texas Instruments which retain guard digits in an attempt to ensure numerical precision in the display, and those manufactured by Hewlett-Packard which do not. The material in V8N2P3 was an introduction to my results.

When one performs the $(\sqrt{2})^2$ test with the HP-41C, and with its companion ten digit calculators the HP-67 and HP-11, all calculate the result as 1.999999999. But the display depends on the display control in force at the time. In the Fix 9 mode for the HP-41C and the HP-11, or in the Display 9 mode of the HP-67 the user can observe all the information available, that is the 1.999999999 answer. There are no guard digits which can somehow be brought to the display, or even used for further chain calculations. In the Fix 2 mode, the display will be 2.00, and in the Fix 0 mode the display will be 2. But those displays do not indicate "funny arithmetic".

With the TI-59, or even with the SR-51A, the $(\sqrt{2})^2$ test will yield a 2. in the display in the turn-on condition, but the display register will contain the value 1.999999999995 . As with the HP product line switching to the Fix 2 display mode will yield 2.00 in the display. Again, different displays in response to different display modes do not indicate "funny arithmetic".

MORE ON NUMERICAL PRECISION (cont)

With the thirteen digit calculators manufactured by Texas Instruments it is the contents of the display register, which may be different from the display, which is used for further chain calculations. The serious users of the thirteen digit calculators, and of the ten digit calculators as well, understand that the display does not always indicate the exact result which resides in the display register.

An extended series of chain calculations will illustrate just how quickly the corruption due to discarding the guard digits can extend into the ten digit display. As an illustration, start with a small integer in the display, then take the square root five times in succession, followed by squaring five times. Pertinent results for the two classes of calculators are:

Input Integer	Result from HP-41C Family	Display from TI-59 Family	Display Register from TI-59 Family
1.	1.000000000	1.	1.000000000000
2.	2.000000022	2.	1.999999999917
3.	2.999999991	3.	2.999999999806
4.	4.000000088	4.	3.999999999654
5.	4.999999931	4.999999999	4.999999999240
6.	5.999999923	6.	5.999999999641
7.	7.000000143	6.999999999	6.999999998864
8.	8.000000007	7.999999999	7.999999999351
9.	8.999999946	8.999999999	8.999999998827
10.	9.999999929	9.999999999	9.999999998748
11.	11.00000003	11.	10.99999999866
12.	12.00000008	12.	11.99999999807
13.	13.00000000	13.	12.99999999745
14.	13.99999990	14.	13.99999999811
15.	15.00000018	15.	14.99999999864
16.	16.00000022	16.	15.99999999810
17.	16.99999994	17.	16.99999999667
18.	17.99999980	18.	17.99999999665
19.	18.99999989	19.	18.99999999679
20.	20.00000027	20.	19.99999999778
21.	21.00000016	21.	20.99999999848
22.	21.99999994	22.	21.99999999597
23.	22.99999982	22.99999999	22.99999999482
24.	23.99999998	24.	23.99999999711
25.	24.99999998	25.	24.99999999863

The TI-59 user can obtain the HP-41 results by truncating the guard digits after each operation using the EE-INV-EE sequence. It is not quite so easy for the HP-41 user to obtain the TI-59 results. The table shows that corruption has extended into the eighth digit with the ten digit calculators, but only to the tenth digit with the thirteen digit calculators. The results from the thirteen digit calculators are always low, probably because those calculators truncate not round in determining the lowest order digit in the display register. The results from the ten digit calculators are sometimes too high, and sometimes too low, probably due to the rounding to obtain the low order digit.

MORE ON NUMERICAL PRECISION (cont)

Discussions of inadequate numerical precision are not limited to hand-held calculators. A recent question for the A.P.P.L.E. Doctor in the June 1982 issue of Call--A.P.P.L.E. on page 77 asked:

> "I wonder if you could explain why the Apple languages Applesoft, Pascal and Fortran have used only 32-bit floating point numbers which will give only six-seven place accuracy. Microsoft Basic-80 and Fortran used with the Z-80 Softcard offer the additional double precision numbers with 64 bits. ..."

Not surprisingly, the proponents of the APPLE answered with a defense of lesser numerical precision with words like "It is derned difficult to measure something to nine significant figures ..." and the like. Even the square root of two example is used. But the author used his Texas Instruments SR-56 " ... which calculates to 13 places internally and displays to 10 places ... " to check his results. That clearly illustrates one important use for the calculators with more numerical precision. One can use them to check the calculations of calculators or computers which have less adequate precision. And the users of the thirteen digit machines have found other uses for those guard digits as described in my article "There's Gold in Those Guard Digits" in the May/June 1982 issue of PPX Exchange.

The accuracy and consistency of the trigonometric functions provide another interesting comparison of numerical precision. Consider the material in Appendices C and D of Personal Programming. On page C-1 there is the example where t register comparisons of the whole 13 digit values of sin 45° and cos 45° will not show equality. The exzmple shows that the difference between sin 45° and cos 45° is 7E-13, and goes on to state that when doing t register comparisons precautions should be taken to prevent improper evaluation due to the guard digit differences. The use of the sequence EE-INV-EE to truncate the guard digits of a result and leave only the rounded display is suggested. That works perfectly well for the case of 45 degrees. But consider the case of sin 39° and cos 51° using the value from AMS 55 as a reference:

AMS 55	sin 39°	=	cos 51°	=	0.62932 03910 49837
TI-59	sin 39°	=			0.62932 03910 495
TI-59			cos 51°	=	0.62932 03910 514
TI-59	39 sin EE INV EE				0.62932 03910
TI-59	51 cos EE INV EE				0.62932 03911
HP-11	sin 39°	=	cos 51°	=	0.62932 03910

In the INV FIX or Fix 9 mode the EE-INV-EE sequence does not ensure that the sin 39° will equal the cos 51°. Now you might think that we can solve that problem by doing the EE-INV-EE sequence while in the Fix 8 display mode. But some experimentation and thought will show that there will be cases where that will not work. For example, for the TI-59

sin 32.9999999989°	=	0.54463 90349 985
cos 57.0000000011°	=	0.54463 90350 005

where the argument can be synthesized by the sequence 32 + .9999999989 =.

MORE ON NUMERICAL PRECISION (cont)

The results at the bottom of the previous page are those which are in the display register. If the calculator is in the FIX 9 or INV FIX display mode, then the displays will be

sin 32.9999999989 = 0.54463 9035

cos 57.0000000011 = 0.54463 9035

and the EE-INV-EE sequence will truncate to the display value such that the two functions are equal. But, if the calculator is in the FIX 8 mode, then the displays will be

sin 32.9999999989 = 0.54463 903

cos 57.0000000011 = 0.54463 904

and the EE-INV-EE sequence will truncate to the display values which are not equal. Some thought will reveal that a similar condition can be found no matter what FIX mode the calculator is in. Clearly, t register comparisons of trigonometric results can never be entirely safe. But the statement on trigonometric function accuracy at the top of page C-2 of Personal Programming seems to be correct:

> "TRIGONOMETRIC FUNCTIONS - All displayed digits in standard format are accurate to +/- 1 in the 10th digit for a +/- 36,000 degree range .... ."

By contrast, the HP-11 (and I assume the HP-41, since to date my limited tests show that the two calculators do exactly equivalent arithmetic) seems to always generate $\sin\theta = \cos(90 - \theta)$. But for the sin 31° and cos 59° example above the solution of 0.62932 03910 from the HP-11 is in error by 498 E-13. For the TI-59 the two solutions are not equal, but each solution is correct to about 3 E-13 for sin 39°, and to about 16 E-13 for cos 51°. In other words, although the two solutions are not equal, either is more accurate than the HP-11 by a factor of 30 or so. The user has to decide whether the availability of safe comparisons or the availability of extra precision is more desireable. Of course, the best situation would have been for the TI-59 to obtain equal results to thirteen digits.

There are other interesting differences between the two classes of calculators. The Powers of Minus One puzzle (V7N1/2P9, V7N3P12/13, V7N4/5P24, and V8N1P5) generated wide-spread interest for TI-59 users. Charlie Williamson points out that for the HP calculators the puzzle is trivial. With those calculators a negative number raised to an <u>integer</u> power is correctly calculated; a negative number raised to a power that is not an integer generates an error indication.

Subsequent pages of this issue contain preliminary results from a much more thorough examination of the sine and cosine functions of the TI-59 by George Thomson. In future issues I will examine the idiosyncrasies of other transcendental functions of the TI-59.

PROGRAMMING CHALLENGE - Jim McDermott of <u>EDN</u> suggested that a list processor would be a useful subroutine for the TI-59. The value in the display should go to the bottom of the list. Each entry in the list should move up one location. The value which had been at the top of the list should return to the display. How can we do this in a minimum of steps?

BOOK REVIEW - Use of the TI-59 with Applications to Probability and Statistical Analysis. G. R. Nelson and E. E. Stanton. 1980. National Technical Information Service (NTIS), Springfield, VA 22161. 161 pages. Paperback, $15.00 shipping included.

This treatment of statistics on the TI-59 was submitted by the authors in partial fulfillment of the requirements for the degree of Master of Science in Management at the Naval Postgraduate School, Monterey, California. The book includes exhaustive treatments of three statistics problems including theoretical background, equations, flow charts, program listings and results. The use of the Applied Statistics library module is illustrated. The abstract states:

> This thesis demonstrates through three comprehensive examples, the capabilities of the TI-59 programmable hand-held calculator as an analytical tool. One example is a probability application while the other two examples entail use of the TI-59 in statistical inference and data analysis. The probability example involves the use of the Monte Carlo technique to simulate stochastically the detection, identification and engagement of a cruise missile by an Improved Hawk Air Defense Battery. The second example illustrates a TI-59 program which is designed to analyze sample data. The data used for this illustration were gathered by the authors in an experiment which encompassed the testing of thirty-six male subjects to determine the extent to which their training routines influenced their strength, endurance, and cardiovascular fitness. The third example involves the use of an ANOVA routine and Scheffe's multiple contrasts to demonstrate how the TI-59 may be used to facilitate statistical inference. The fitness data are also used for this purpose. The intent throughout the thesis is to exemplify the capabilities of the TI-59 as a viable, real world analytical tool rather than emphasize particular results of the simulation or experiment.

COMPUTERS IN BIOLOGY AND MEDICINE - On V6N6/7P24 and V8N1P11 member Clifford Lieberman invited club members to submit programs for publication in that journal. Volume 12 Number 4 of the journal contained two articles:

A Pocket Calculator Program to Evaluate Confidence Limits for the Relative Risk. A. J. Macleod. Pages 323-330.

The Newton-Raphson method of solution is used in this 518 step program.

A Numerical Method for Biphasic Curve Fitting with a Programmable Calculator. D. Ristanovic, D. Ristanovic, J. Malasevic, and B. Milutinovic.

One of the authors is club member Dejan Ristanovic of Yugoslavia. The 431 step program includes a fast mode option using the Stflg Ind at the end of the current partition technique.

ELECTRO-OPTICS - The April 1983 issue contains an article Measuring Rigid-Body Motion by R. Page, pages 24 to 28. A TI-59 program is included to aid in the analysis of holographic testing. Program steps 322 through 383 illustrate the use of the ML-02 routine in the Master Library module to solve a set of linear equations.

EXAMINATION OF THE ACCURACY OF THE TI-59 SINE AND COSINE FUNCTIONS

by George Wm. Thomson. The TI-59 computes the three trigonometric functions and their inverses using wired-in software ("firmware") which cannot be altered by the user. In this investigation the accuracy of the sine and cosine functions was evaluated over the 0 to 90 degree range at one degree intervals. The sine function was fould to be more accurate that the cosine. Simple corrections were devised to provide somewhat improved accuracy over the rnage.

Excellent tables for comparisons to well beyond the accuracy needed are readily available. The earliest goes back to 1633. (Editor's Note: Pages 189 through 197 of AMS-55 provides 15 place tables at 0.1 degree increments.) In the following discussion all differences are expressed from fifteen place values rounded to the 13 place TI-59 limit. That is, a difference of -5 means that the TI-59 value is 5 lower than the correct value in the 13th place.

How good is the sine function? In general within +/- 15 E-13 but with an error of 17 E-13 for 72 degrees. The deviations seem somewhat cyclic with a cross-over near 45 degrees. The RMS error for 89 points from 1 degree to 89 degrees is 6.77 E-13.

How good is the cosine? On the whole, the cosine is much less accurate and considerable more erratic than the sine. The values are systematically high. The mean for the 89 points is +11.3 while the mean for the sine was only +1.6 . There was no negative difference for the 89 points. The scatter of the points was also large. For example, between 58 and 66 degrees the values range from 3 to 26. Over the whole range the use of sin(90 - x) is recommended instead of cos x . Despite using a square, a subtraction and a square root, calculation of the cosine from SQRT(1 - $\sin^2 x$) is as good as using sin(90 - x) for angles from 0 to 45 degrees.

The cyclic form of the sine differences suggest use of a correction function. The choice of the right functional form is always difficult. A good one for an additive correction to the firmware sine function should be simple in form, should satisfy the boundary conditions so that the curve has the right shape, and, let us dream, have simple constants. I will not take you through the steps of the curve fitting which was done by graphical methods using Chebysheff principles, where the effort is to contain the errors within a minimum plus-and-minus band over the range rather than minimize in some least squares sense. Concepts of randomness are not very meaningful when the details of firmware arithmetic are hidden from view and differences are being expressed at the extreme limit of the TI-59 capability.

Equation A is simply the addition of 10 sin 4x times 1EE-13 giving values which are slightly high numerically for 0 to 30 degrees, and low at the high end, but a fairly good compromise.

Equation B has a second term added to allow for some asymmetry; (9 sin 4x - 3 sin 2x) times 1EE-13. This improved the fit below 45 degrees.

Finally I fitted a "constrained cubic" ... the cubic had so many conditions or constraints on it that there were only two constants to establish: subtract x(x - 40)(90 - x)/4EE16. The cubic fit in general provides an error within 10 E-13 over the whole range. Error curves and additional comments appear on the next page.

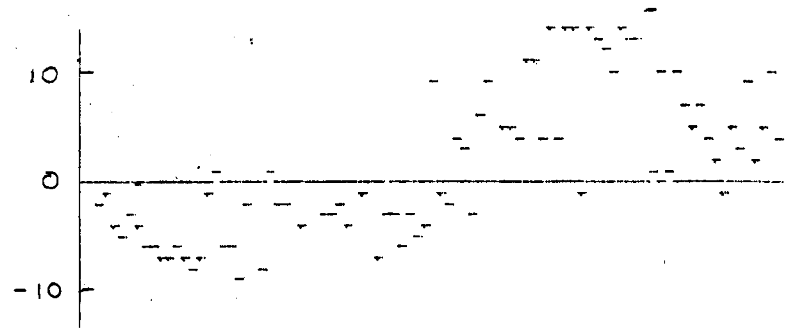
EXAMINATION OF ... THE SINE AND COSINE FUNCTIONS (cont)

The error plots for the different sine functions were obtained using Michael Sperber's PLOT 60 high resolution graphics routine (V6N4/5P5).

TI-59 Errors without any correction

Mean Error = 1.6E-13

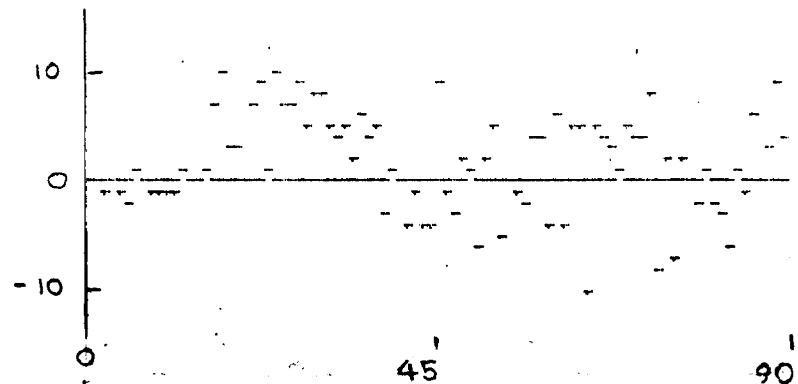
RMS Error = 6.8E-13



TI-59 Errors corrected with 10(sin 4x)(1E-13)

Mean Error = 1.6E-13

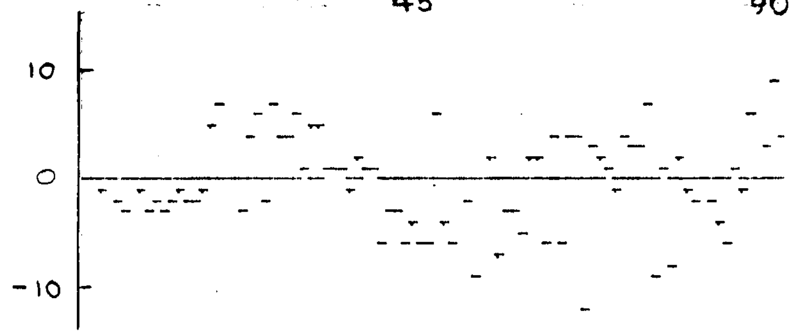
RMS Error = 4.6E-13



TI-59 Errors corrected with (9sin 4x - 3sin 2x)(1E-13)

Mean Error = -0.3E-13

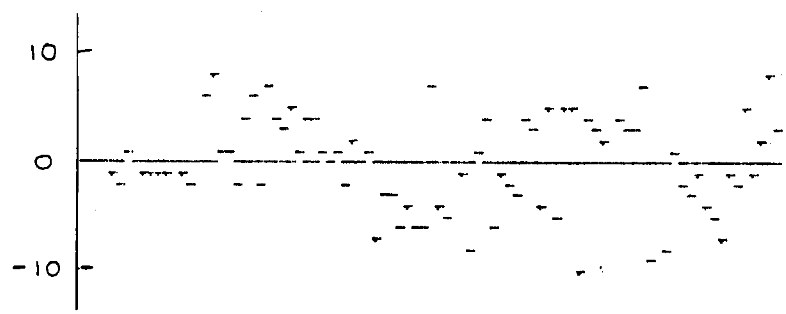
RMS Error = 4.2E-13



TI-59 Errors corrected with x(x-40)(90-x)/4E+16

Mean Error = -0.04E-13

RMS Error = 4.0E-13



Editor's Note: Any of these correction formulas are appropriate for the one degree intervals from which they were derived. They may not be appropriate for all arguments. My tests show that while the errors for the even degrees from 66 through 71 are +13, +12, +10, +14, +13 and +13 the errors for the 0.1 degree increments from 66.0 through 71.9 range from -3 to +20 with a mean of 7.6 and an RMS of 9.4 . This suggests that for all arguments another correction formula is needed. More on that in the next issue.

ANOTHER GAME - David Lane. Now that we have a rampant stock market, it may benefit you to hone your financial savvy. Thus, I offer the following game:

DJA - Dow Jones Average

This is not just a random stock market. The DJA is driven by two cycles - a long cycle of approximately one month, and a short cycle of about one week. Each stock has its own Beta - which is a multiplier of the DJA movement. The calculator determines for each day the DJA and the current price of five stocks.

You start our with $10,000 for each of five stocks. You (or five different individuals) can buy or sell stock each day. You can't buy more stock than you have money for, and you can't sell more stock than you own. If you are playing by yourself, you can't transfer funds from one stock account to another.

A month's market is played. At the end, the calculator shows how much of the $10,000 you have left, or how much it has grown to. On any day you can request your current cash position, including the number of shares you own.

User Benefits:

Learn that the market is predictable (and also unpredictable). Realize that you must sell, as well as buy, to win.

See how easy it is to lose your shirt in the market.

Rules of the Game:

For each simulated day, the calculator will provide the DJA (Dow Jones Average) and the current price of 5 stocks labeled A, B, C, D, and E.

Each day you may buy or sell stock. You can't spend more money than you have, and you can't sell more stock than you have.

You have 20 days (one working month) to win your fortune; then a new market with new stocks must be played.

Sorry, no option trading - but no commissions are charged.

The DJA should stay between 20 and 2,000. The stock prices will stay between 0 and 500.

Up to 5 players can play at once - if each chooses a different stock.

The calculator will keep track of your stock transactions and your cash (profit/loss) position.

Background:

For those who want to understand what is happening in this market, the following information is given (but not justified).

The DJA is driven by two cycles: a long and a short. Each of the two cycles (some number of days) is randomly initialized, but stays fixed during a month. The starting day of the cycles is also random. The initial value of the DJA is also random, of course, so that you don't know if it's really high or low. For each subsequent day, the sum of the magnitudes of the 2 cycles is used as a multiplier for the DJA calculation.

ANOTHER GAME (cont)

The stocks are driven by the DJA. Each has its own starting random value beta (multiplier of DJA), and each has its own initial value.

That's all there is to it. Now that you understand all of the forces that drive the prices of the stocks, you should be able to intelligently win money all the time. (If you do, let me know so I can win too).

Instructions:

(1) Read in the card, sides 1 and 2.

(2) Enter a random number seed and press E' to run. The calculator will print out 5 days of the DJA and stock prices to give you a feel for the market. It then stops with 1.00 in the display. If you start with a seed of pi, then the first printout will appear as at the right. Note that the days will count down starting at 25. After printing the information for day 21 the calculator waits for instructions.

```
  25.00   DAY
1237.54   DJA
  87.01     A
  79.54     B
  53.79     C
  74.16     D
  86.30     E
```

(3) To buy stock you enter a positive number equal to the number of shares you wish to buy and press the User Defined Key with the letter of the stock you wish to buy. The calculator responds with a printout as shown at the right. The first line shows the stock and the price per share. The second line shows the number of shares purchased. The third line shows the dollars spent.

```
   80.52    A
  100.00
-8051.64
```

(4) To sell stock you enter a negative number equal to the number of shares you wish to sell and press the appropriate letter key. Again, the calculator responds with a printout describing the transaction. An example appears at the right.

```
  88.22    E
-110.00
9704.54
```

(5) To examine the cash position of each stock account press A'. The calculator will print a summary for each stock account as illustrated at the right. The first line is the number of shares owned. The second line is the dollar value of the stock held. The third line is the dollar value of cash, and the fourth line is the total dollar value of the account.

```
  120.00    E
10586.77
  177.23
10764.00
```

(6) When all players have completed their transactions for the day then press R/S. The calculator will print the next day's data as in step (2) above, and the players can buy and sell at the new prices. To run for more than one day enter the number of days to run and press R/S.

ANOTHER GAME (cont)

(7) At day 1 the game ends. The calculator prints the market summary (step 2) and the cash position of each stock (step (5).

Program Listing:

```
000  76 LBL     080  24 CE      160  76 LBL     240  00  00     320  65  ×      400  93  .
001  23 LNX     081  98 ADV     161  79  x̄      241  01  1      321  43 RCL     401  05  5
002  73 RC*     082  01  1      162  01  1      242  00  0      322  02  02     402  18 C'
003  00  00     083  06  6      163  03  3      243  00  0      323  99 PRT     403  93  .
004  69 OP      084  01  1      164  42 STO     244  33 X²      324  74 SM*     404  03  3
005  04  04     085  03  3      165  00  00     245  72 ST*     325  00  00     405  95  =
006  69 OP      086  04  4      166  05  5      246  00  00     326  95  =      406  42 STO
007  20  20     087  05  5      167  42 STO     247  69 OP      327  94 +/-     407  47  47
008  73 RC*     088  69 OP      168  01  01     248  20  20     328  99 PRT     408  94 +/-
009  00  00     089  04  04     169  76 LBL     249  97 DSZ     329  69 OP      409  85  +
010  65  ×      090  43 RCL     170  55  ÷      250  01  01     330  20  20     410  01  1
011  43 RCL     091  04  04     171  98 ADV     251  43 RCL     331  74 SM*     411  95  =
012  48  48     092  69 OP      172  73 RC*     252  98 ADV     332  00  00     412  42 STO
013  85  +      093  06  06     173  00  00     253  61 GTO     333  98 ADV     413  51  51
014  69 OP      094  01  1      174  69 OP      254  24 CE      334  61 GTO     414  05  5
015  20  20     095  06  6      175  04  04     255  76 LBL     335  32 X:T     415  00  0
016  73 RC*     096  02  2      176  03  3      256  15  E      336  76 LBL     416  18 C'
017  00  00     097  05  5      177  44 SUM     257  69 OP      337  16 A'      417  05  5
018  95  =      098  01  1      178  00  00     258  22  22     338  71 SBR     418  00  0
019  50 I×I     099  03  3      179  73 RC*     259  76 LBL     339  79  x̄      419  95  =
020  69 OP      100  69 OP      180  00  00     260  14  D      340  61 GTO     420  42 STO
021  20  20     101  04  04     181  69 OP      261  69 OP      341  32 X:T     421  49  49
022  72 ST*     102  43 RCL     182  20  20     262  22  22     342  76 LBL     422  07  7
023  00  00     103  43  43     183  65  ×      263  76 LBL     343  78 Σ+      423  00  0
024  69 OP      104  44 SUM     184  73 RC*     264  13  C      344  71 SBR     424  00  0
025  06  06     105  44  44     185  00  00     265  69 OP      345  79  x̄      425  18 C'
026  03  3      106  43 RCL     186  69 OP      266  22  22     346  98 ADV     426  05  5
027  44 SUM     107  45  45     187  06  06     267  76 LBL     347  98 ADV     427  00  0
028  00  00     108  44 SUM     188  95  =      268  12  B      348  98 ADV     428  00  0
029  97 DSZ     109  46  46     189  99 PRT     269  69 OP      349  61 GTO     429  95  =
030  01  01     110  43 RCL     190  69 OP      270  22  22     350  44 SUM     430  42 STO
031  23 LNX     111  44  44     191  20  20     271  76 LBL     351  76 LBL     431  50  50
032  22 INV     112  38 SIN     192  85  +      272  11  A      352  10 E'      432  42 STO
033  97 DSZ     113  65  ×      193  73 RC*     273  48 EXC     353  98 ADV     433  44  44
034  04  04     114  43 RCL     194  00  00     274  02  02     354  99 PRT     434  42 STO
035  78 Σ+      115  51  51     195  99 PRT     275  65  ×      355  85  +      435  46  46
036  05  5      116  85  +      196  95  =      276  06  6      356  07  7      436  01  1
037  18 C'      117  43 RCL     197  99 PRT     277  85  +      357  95  =      437  03  3
038  00  0      118  46  46     198  69 OP      278  07  7      358  42 STO     438  42 STO
039  95  =      119  38 SIN     199  20  20     279  95  =      359  59  59     439  00  00
040  42 STO     120  65  ×      200  97 DSZ     280  42 STO     360  07  7      440  42 STO
041  57  57     121  43 RCL     201  01  01     281  00  00     361  22 INV     441  02  02
042  44 SUM     122  47  47     202  55  ÷      282  73 RC*     362  23 LNX     442  05  5
043  44  44     123  95  =      203  92 RTN     283  00  00     363  42 STO     443  42 STO
044  59 INT     124  42 STO     204  76 LBL     284  69 OP      364  58  58     444  01  01
045  65  ×      125  48  48     205  43 RCL     285  04  04     365  76 LBL     445  98 ADV
046  06  6      126  65  ×      206  43 RCL     286  04  4      366  44 SUM     446  61 GTO
047  85  +      127  43 RCL     207  02  02     287  44 SUM     367  22 INV     447  43 RCL
048  01  1      128  49  49     208  72 ST*     288  00  00     368  58 FIX     448  76 LBL
049  05  5      129  85  +      209  00  00     289  73 RC*     369  05  5      449  52 EE
050  95  =      130  43 RCL     210  69 OP      290  00  00     370  69 OP      450  32 X:T
051  42 STO     131  50  50     211  22  22     291  94 +/-     371  17  17     451  07  7
052  03  03     132  95  =      212  69 OP      292  32 X:T     372  47 CMS     452  01  1
053  43 RCL     133  69 OP      213  20  20     293  43 RCL     373  06  6      453  07  7
054  57  57     134  06  06     214  69 OP      294  02  02     374  69 OP      454  01  1
055  22 INV     135  01  1      215  20  20     295  22 INV     375  17  17     455  69 OP
056  59 INT     136  03  3      216  08  8      296  77  GE     376  58 FIX     456  04  04
057  65  ×      137  42 STO     217  00  0      297  52 EE      377  02  02     457  32 X:T
058  01  1      138  00  00     218  18 C'      298  32 X:T     378  02  2      458  69 OP
059  00  0      139  05  5      219  01  1      299  69 OP      379  05  5      459  06  06
060  75  -      140  42 STO     220  00  0      300  20  20     380  42 STO     460  61 GTO
061  05  5      141  01  01     221  95  =      301  73 RC*     381  04  04     461  32 X:T
062  95  =      142  61 GTO     222  72 ST*     302  00  00     382  05  5      462  00  0
063  74 SM*     143  23 LNX     223  00  00     303  55  ÷      383  42 STO     463  00  0
064  03  03     144  76 LBL     224  69 OP      304  69 OP      384  06  06     464  00  0
065  44 SUM     145  18 C'      225  30  30     305  30  30     385  08  8      465  00  0
066  46  46     146  48 EXC     226  55  ÷      306  69 OP      386  18 C'      466  00  0
067  97 DSZ     147  59  59     227  02  2      307  30  30     387  01  1      467  00  0
068  06  06     148  65  ×      228  94 +/-     308  73 RC*     388  01  1      468  00  0
069  24 CE      149  43 RCL     229  85  +      309  00  00     389  95  =      469  00  0
070  76 LBL     150  58  58     230  05  5      310  95  =      390  42 STO     470  00  0
071  32 X:T     151  95  =      231  05  5      311  32 X:T     391  43  43     471  00  0
072  01  1      152  22 INV     232  95  =      312  77  GE     392  02  2      472  00  0
073  42 STO     153  59 INT     233  18 C'      313  52 EE      393  04  4      473  00  0
074  02  02     154  65  ×      234  05  5      314  73 RC*     394  18 C'      474  00  0
075  98 ADV     155  48 EXC     235  95  =      315  00  00     395  03  3      475  00  0
076  91 R/S     156  59  59     236  72 ST*     316  69 OP      396  06  6      476  00  0
077  42 STO     157  95  =      237  00  00     317  06  06     397  95  =      477  00  0
078  06  06     158  85  +      238  04  4      318  69 OP      398  42 STO     478  00  0
079  76 LBL     159  92 RTN     239  44 SUM     319  20  20     399  45  45     479  00  0
```

NEWCOMER'S SPECIAL - Maurice Swinnen forwarded a number of copies of the V6N9/10 issue of TI PPC Notes to me for disposition. These are single issues which are not parts of complete sets. Since it seems a waste to have these taking up space in my garage I have decided to offer them to newcomers for two dollars on a while they last basis.

Highlights of the issue were Patrick Acosta's discussion of hexadecimal codes (but see V8N1P17) and Dejan Ristanovic's description of branching from the keyboard (but see V8N2P2). A copy of the index for the issue is at the right.

The third item "Codes 21 and 26" is Maurice's admission that he had been taken in by the counter April Fool's story by John Worthington and Emil Regelman.

Send two dollars (no checks, please) to PPC Publications, P.O. Box 1421, Largo, Florida 34294-1421.

THE CC-40 - Maurice Swinnen writes: The CC-40 is a good computer ... the keyboard is smaller than the one on the typewriter. It has a lot of one-keystroke entries for programming such as PRINT, FOR, NEXT, etc. The Basic is enhanced by a lot of subprograms which you can reach by CALL XXXXX. All information on memory mapping is given such that it is easy to do assembly language programming. It has both CALL PEEK and CALL POKE commands, plus a CALL DEBUG. I wrote several programs--JIVE TURKEY and others. Because I sorely missed a printer I concocted an RS-232 interface and now I can use any printer on it. (Editor's Note: Late news releases from TI indicate that peripherals for the CC-40 should be available. As I write this the CC-40 is available in retail stores in this area, but the peripherals are not.)

The speed on the CC-40 is much faster than on the 59, of course. Counting from 1 to 100 was fast this time, too fast to clock directly. So I put it in a loop and let it count to 100 one hundred times. That took 34 seconds, which makes the time for counting to 100 equal to 0.34 seconds. Not bad! Then I tried to compute factorials. The highest factorial I could generate directly before overflow was 84. It took exactly 1.37 seconds, again measured in a loop of 100 for accuracy.

Editor's Note: Maurice's JIVE TURKEY program appears on the following page. I have also had an engineering model of the CC-40 for about a month, and performed other speed comparisons. The keyboard is what TI calls a 3/4 keyboard, meaning it is 3/4 the distance between the keys relative to a full size keyboard. That means it is essentially impossible to touch type. The HP-75 has approximately a 0.8 keyboard. Touch typing is trying at best. The Radio Shack Model 100 has a full size keyboard.

JIVE TURKEY on the CC-40. Maurice E.T. Swinnen

```
100 DISPLAY AT(6)"* JIVE TURKEY GAME *":PAUSE 2
110 SCORE=0:FIB=0:RANDOMIZE:SECRET=INTRND(100)
120 DISPLAY ERASE ALL"PROBABILITY OF TRUTH? 0-100?;
130 ACCEPT AT(29)BEEP VALIDATE(DIGIT);PROB
140 ROLL=INTRND(100):SCORE=SCORE+1:DISPLAY"YOUR GUES? 0-100";
150 ACCEPT AT(20)BEEP VALIDATE(DIGIT);GUESS:IF GUESS=SECRET THEN 190
160 IF PROB>ROLL THEN FLAG=1 ELSE FLAG =0:IF FLAG=0 THEN FIB=FIB+1
170 IF GUESS<SECRET THEN IF FLAG=1 THEN 240 ELSE 230
180 IF GUESS>SECRET THEN IF FLAG=1 THEN 230 ELSE 240
190 PRINT"CONGRATULATIONS! YOU DID IT!":PAUSE 3
200 DISPLAY AT (3)"SCORE=";SCORE,"# OF FIBS=";FIB:PAUSE
210 DISPLAY"SAME GAME AGAIN? Y/N";:ACCEPT AT(22)BEEP VALIDATE("YNyn"),ANSWER$
220 IF ANSWER$="Y" OR ANSWER$="y" THEN 110 ELSE 250
230 PRINT"GUESS TOO HIGH":PAUSE 1:GOTO 140
240 PRINT"GUESS TOO LOW":PAUSE 1:GOTO 140
250 DISPLAY AT(5)ERASE ALL"BYE, HAVE A NICE DAY!":PAUSE 3:END
```

PALINDROMIC NUMBERS IN BASIC - Palmer Hanson. Page 6 of this issue reports the results of some extensive tests of the TI-59 generating palindromic numbers using digit reverser techniques. Albert Smith found 23 numbers between 1 and 1900 which would not reach a palindromic number within the range of the TI-59. I wrote the following BASIC program for the CC-40 to investigate those numbers further.

```
10 INPUT "A$ =";A$
15 N = 0
20 L = LEN(A$)
25 B$ = ""
30 FOR I = L TO 1 STEP -1
35 B$ = B$ & SEG$(A$,I,1)
40 NEXT I
50 IF A$ = B$ THEN 200
100 C$ = "":A10 = 0
105 FOR I = L TO 1 STEP -1
110 A = VAL(SEG$(A$,I,1)) + VAL(SEG$(B$,I,1)) + A10
115 IF A > 9 THEN C = A - 10 ELSE C = A
120 C$ = STR$(C) & C$
125 IF A > 9 THEN A10 = 1 ELSE A10 = 0
135 NEXT I
140 IF A10 = 1 THEN C$ = "1" & C$
145 N = N + 1
150 PRINT N
155 A$ = C$
160 GOTO 20
200 PRINT A$;N
210 PAUSE 10
220 GOTO 10
999 END
```

PALINDROMIC NUMBERS IN BASIC (cont)

The program uses digit by digit string manipulation such that its operation is independent of the word length of an individual computer. Variations of the program were also run on a Radio Shack Color Computer, a Radio Shack TRS-80 Model 100 Portable Computer, and an Apple. The relative execution times to change 89 into 8813200023188 in 24 steps were:

TI-59 in normal mode	4 min 51 sec
TI-59 in EE mode	4 min 37 sec
TI-58C in normal mode	6 min 7 sec
CC-40	27 seconds
Color Computer	18 seconds
Apple	10 seconds
Model 100	18 seconds

With the insertion of a CLEAR 1024 command at line number 5 the string limitation which limited the number of iterations to about 140 was removed with the Model 100 and raised to about 580 iterations. Tests showed that not one of the 23 numbers listed on page 6 would reach a palindromic number where the final number prior to string overflow was 255 digits long! I also noticed that there was a pattern in the numbers 1495 through 1857 on page 6 which suggested that the numbers 1945 and 1947 would also fail to yield a palindromic number, and verified that with the Model 100.

FINDING PI IN BASIC - Palmer Hanson. The CC-40 implementation of BASIC provides a PI function and permits the arguments for the trigonometric functions to be entered in degrees, radians, or grads--one indication of the emphasis on scientific useage for the CC-40.

For those BASIC mechanizations which do not provide a PI function and which are limited to radian arguments for the trigonometric functions the programmer often wants the value of PI for use in conversions from degrees to radians. An old programmer's trick which recovers the value of PI to the accuracy of the individual machine is to use the function PI = 4*ATN(1). I had used that technique satisfactorily on many computers until I encountered the Radio Shack Model 100. When using the conversion factor derived from ATN(1)/45 (equivalent to 4*ATN(1)/180) I found that the cosine of 60 degrees was returned as .5000000001147 , which is simply not consistent with a fourteen digit machine. After some experimentation I found that the use of a conversion factor derived from ATN(3E13)/90 would result in the cosine of 60 degrees being returned as .49999999999998 --respectable accuracy in anyone's book. Similar improvements in the accuracy of the trigonometric functions on the Model 100 were found for other functions and other arguments. I have tentatively concluded that the ATN function on the Model 100 is weak.

With this information in hand I decided to examine the capability of other calculators and computers to evaluate pi. I found a wide range of capability ranging from the nine digit capability of the Apple II, the Radio Shack Color Computer and the Atari 400, through the ten digit capability of the HP product line of programmable calculators to the fourteen digit capability of the Model 100. The table on the following page summarizes my experience.

DERIVING PI IN BASIC (cont)

	From 4*ATN(1)	From 2*ATN(N)
AMS-55 Reference	3.1415 92653 58979	3.1415 92653 58979
Commodore VIC-20	3.1415 9266	3.1415 9266
Color Computer	3.1415 9266	3.1415 9266
Apple II	3.1415 9266	3.1415 9266
Atari 400	3.1415 9267	3.1415 9264
HP-11	3.1415 92654	3.1415 92654
TI-57	3.1415 92653 2	3.1415 92653 6
TI-55II & TI-57LCD	3.1415 92653 5	3.1415 92653 4
TI-58/58C/59	3.1415 92653 588	3.1415 92653 590
TI-99/4A	3.1415 92653 59	3.1415 92653 59
CC-40	3.1415 92653 59	3.1415 92653 59
Model 100	3.1415 92653 1932	3.1415 92653 5898

In the table the N in 2*ATN(N) is a number sufficiently large such tnat no further changes in ATN(N) will occur with larger N. For the Model 100 that value is about 3E13. For the CC-40 that value is about 2E12. For the TI programmable calculators and the CC-40 the values listed are those internal to the machine not those displayed.

The predominance of TI machines, including the CC-40, at the high accuracy end of the table is as expected. The CC-40 also provides the arcsin and arccos functions which are not available on the other "home" computers--one more instance of attention to scientific applications.

A TI-58/59 ALARM - V8N1P7 described an audible alarm for use with the TI-58/59. Gene Friel reminded me that V6N2P6 of PPX Exchange described the use of radio interference from the TI-58/59 as an audible alarm. The original idea was due to member Laurance Leeds. The article recommended tuning the AM radio to 720 Khz. You will need to experiment. Some radios provide a well defined tone. Others are almost unaffected. The result is a continuous tone when the calculator is running, and disappearance of the tone when the calculator stops. Ending the program with an error will yield a pulsating tone.

TABLE OF CONTENTS

MAGNETIC CARD SERVICE

See V8N1P32 for details. One dollar per card plus a stamped and self-addressed envelope. For programs in this issue:

1287 Digits of Pi	$1.00
Exact Factorials	$1.00
Another Game - DJA	$1.00

No Checks Please.