A Collection of Information on The TI CC-40 Computer

by

Palmer O. Hanson, Jr.

June 1985

****
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.

****

<u>THE CC-40</u> - 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.

<u>Editor's Note</u>: 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
```

<u>PALINDROMIC NUMBERS IN BASIC</u> - 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.

FOURTEEN DIGITS OF PI FROM THE 99/4 AND CC-40 - Myer Boland

"Finding Pi in BASIC" in V8N3P26 reported that both the TI-99/4A and the CC-40 returned the twelve digits 3.1415 92653 59 in response to the BASIC instruction P = 4*ATN(1) . Myer Boland reports that one can recover fourteen digits with the equation P = 4000*ATN(1) on the TI-99/4A, and I verify the same result with the CC-40:

Pi x 1000 exact	=	3141.5 92653 58979 3 ...
4000*ATN(1)	=	3141.5 92653 5898

Unfortunately, at least on the CC-40, if one tries to convert to the value of pi, not 1000xpi, by dividing the result by 1000, the end result reverts to the twelve digit value 3.1415 92653 59 . This is one more illustration of the kind of results which occur with BASIC, but which we would not expect with the typical calculator.

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.

CC-40 GRAPHICS - Maurice Swinnen. These whimsical little programs illustrate the use of the CHAR command (page 5-15 of the CC-40 User's Guide to generate user defined characters. The characters are then called in sequence to provide an illusion of motion. The first program moves a character across the screen while performing the old "jumping jack" exercise. The second program uses seven characters (all that are allowed) to generate a "soccer" figure which moves the ball back and forth across the screen.

JUMPING JACK

```
100 CALL CHAR(0,"0E0E150E04040A11"):CALL CHAR(1,"0E0E040E15040404")
110 FOR I=1 TO 31:FOR J=0 TO 1:DISPLAY AT(I),CHR$(J):PAUSE .3
120 NEXT J:NEXT I
130 FOR K=31 TO 1 STEP -1:FOR L=0 TO 1:DISPLAY AT(K),CHR$(L)
140 NEXT L:NEXT K
150 GOTO 110
```

SOCCER

```
100 CALL CHAR(0,"0E0E150E04040A11"):CALL CHAR(1,"001A1A1B06020509")
110 CALL CHAR(2,"0001050305191919"):CALL CHAR(3,"000105031D191901")
120 CALL CHAR(4,"150E04040A110E0E"):CALL CHAR(5,"0010141814131313")
130 CALL CHAR(6,"000B0B1B0C081412")
140 FOR A=10 TO 21:FOR B=0 TO 6:DISPLAY AT(A),CHR$(B):PAUSE .1
150 NEXT B:NEXT A:PAUSE .5
160 FOR A=21 TO 10 STEP -1:FOR B=6 TO 0 STEP -1
170 DISPLAY AT(A),CHR$(B):PAUSE .1
180 NEXT B:NEXT A:PAUSE .5:GOTO 140
```

What is the memory protection for the CC-40? Can I safely bridge a battery removal by having the AC adapter connected? You will recall that we were cautioned that having the Adapter/Charger connected to a TI-58C or TI=59 with the battery pack removed could damage the calculator. The CC-40 manual provides no information. I did not want to do a test with my CC-40 since I run the risk of destroying all my accumulated programs. Maurice Swinnen says that he has changed batteries without losing his programs. He thinks it took about a minute to make the change. As soon as I have some sort of recording device for the CC-40 I will run the appropriate tests. In the meantime I have asked TI for clarification.

ACCURACY OF THE CC-40 SINE AND COSINE FUNCTIONS - Palmer Hanson

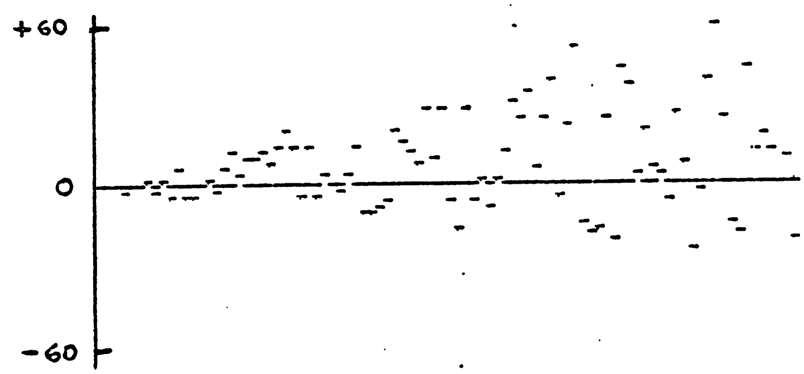
V8N3P18/19 presented George Thomson's analysis of the accuracy of the sine and cosine functions of the TI-58/59. The CC-40 calculates the trigonometric functions to fourteen places and might be expected to yield more accurate results than the TI-59. Examination of the CC-40 sine function for one degree increments from 0 through 90 degrees shows the following errors:

CC-40 Sine Errors

Mean Error = 8.2E-14

RMS Error = 18.3E-14

Peak Error = 59E-14



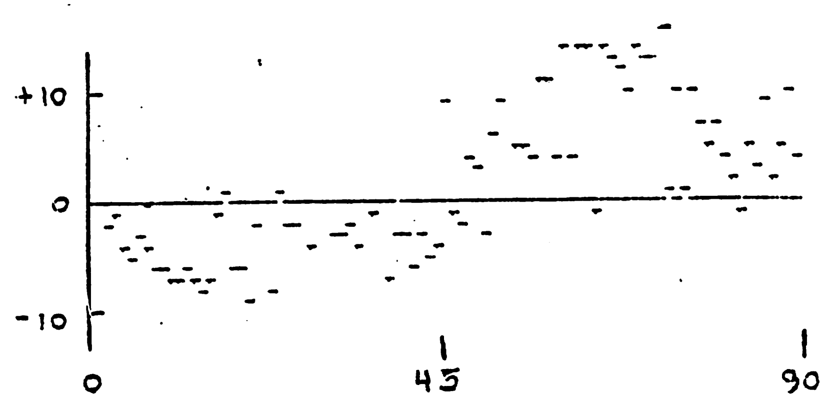
The peak error of 59E-14 occurs at 79 degrees. For a graphic comparison with the TI-59 results the following plots show the TI-59 errors without compensation (same as the top plot on V8N3P19) and the CC-40 errors using the same scale for both plots:

TI-59 Errors without any compensation

Mean Error = 1.6E-13

RMS Error = 6.8E-13

Peak Error = 17E-13

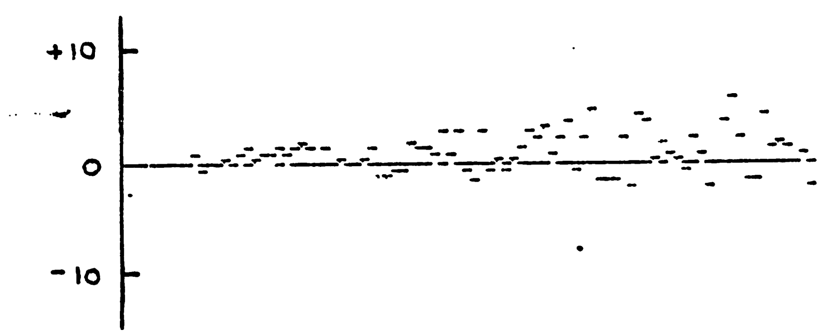


CC-40 Errors

Mean Error = 0.8E-13

RMS Error = 1.8E-13

Peak Error = 5.9E-13



Over the examined range the CC-40 results are nearly four times more accurate than the TI-59. As with the TI-59 the cosine function is less accurate over the same range. The mean cosine error is 5.8E-14, but the RMS cosine error is 37.1E-14, nearly twice that of the sine.

PROMPTING ON THE CC-40 - In V7N7/8P24 Maurice Swinnen described a multi-language capability built into the TI-88 such that prompting could be in English, German or French. The CC-40 provides an extended multi-language prompting capability through the use of the CALL SETLANG(n) command. The assigned language codes are:

0	English
1	German
2	French
3	Italian
4	Dutch
5	Swedish
6	Spanish

For n = 1 the system messages and error messages are in German. For example, the response to the incorrect entry sequence ATN(ENTER is "ungleiche Klammern". For any other value of n the system messages and error messages are in English. In response to the incorrect sequence ATN(ENTER the English response is "Unmatched parenthesis". This output of error messages in text is one of the attractive features of the CC-40. The user need not memorize error codes or translation tables to avoid frequent reference to the manual. The manual does provide extended discussion of each error message.

For programs from a Solid State Software™ module the prompts and messages from the module may be in any of the languages if supported by the particular module. My Mathematics module supports English, German and French. For the Prime Factors program the various messages are:

English	German	French
PRIME FACTORS	PRIMZAHLEN	FACTEURS PREMIERS
Use Printer?	Drucker benutzen?	Utilisation d'une Imprimante?
Enter # To Be Factored:	-> Zahl:	-> Nb a Decomposer:
Exit Program?	Programm verlassen?	Fin du Programme?

The responses to the questions asking for yes/no answers are Y or N in English, J or N in German, and O or N in French. I have not found any information in the manual for the Mathematics module which would tell me which languages are supported. Language codes 3 through 6 result in English messages for that module.

PRIME FACTORS WITH THE CC-40 MATHEMATICS MODULE - The speed of the prime factors program in the CC-40 Mathematics module is disappointing, about ten to forty percent faster than the fastest program for the TI-59, but substantially slower than some programs for the HP-41. Representative speeds for some of the standard problems are:

Program/machine	111111111111	103569859	987654321	9999999967
CC-40	11 sec	32 sec	41 sec	1 hr 55 min
Fast Mode Modulo 210		45 sec	58 sec	2 hr 8 min
Leeds FM (V8N2P26)	17 sec	46 sec	61 sec	2 hr 31 min
Acosta FM 58C	27 sec	61 sec	79 sec	3 hr 6 min
M/U Module - 59	43 sec	163 sec	215 sec	

PRIME FACTORS ON THE CC-40 (cont)

For large primes such as 9999999967 the execution speed of the CC-40 Mathematics module program is about $0.069\sqrt{N}$. Page 19 of the July 1981 issue of the PPC Calculator Journal reported a speed of $0.035\sqrt{N}$ for the HP-41C; but the HP-41C cannot maintain that speed for input integers of more than ten digits.

The CC-40 Mathematics module program has other deficiencies:

* The program stops as each factor is found. A better technique is to store the factors as they are found and continue the search until all factors are found. This minimizes operator attention. A simple additional routine provides for recall of the factors. The technique was illustrated in Laurance Leeds' Speedy Factor Finder in V8N2P26.

* Multiplicity of factors is not indicated. Indication of the multiplicity using a technique such as that devised by George Vogel in his prime factor program in the article "It Pays to Analyze Your Problem" in the January/February 1981 issue of PPX Exchange would be preferred. As George said in that article "Piecemeal presentation of results is slow and inconvenient (try factoring 7,247,757,312). Yet it is not difficult to make the program count the number of times each prime factor occurs, and output the count." George used a decimal point notation where the number after the decimal point indicated the multiplicity. For the number mentioned the output would be 2.28 and 3.03 meaning $2^{28}x3^{3}$.

* Although the CC-40 program can factor input integers of up to twelve digits, it does not provide an ability to recall the input integer correctly for more than ten digits. For example, factor the number 111,111,111,111. You will obtain the correct solution on the first pass, and an "N" in response to the prompt "Exit Program?" will bring the input value back to the display but in exponential notation 1.1111111E+11 . If you run with that value you will get the factors for 111,111,100,000 !

We will have to wait until someone finds out how to download the programs in the modules before we can know if there will be ways to use segments of the module programs, say in the manner in which we can enter the library modules of the TI-59 with the sequence Pgm-XX-SBR-nnn .

CC-40 PERIPHERALS - Peripherals for the CC-40 include a Printer/Plotter a Wafertape™ Digital Tape Drive, and an RS-232 interface. Currently, none of these are available in the Tampa Bay area. The devices are listed in the Sears Fall/Winter 1983 catalog (page 869), in the Educalc Mail Store catalog issue 16 (page 34), and in the Elek-Tek catalog Volume VI (page 17). Inquiries indicate the peripherals will be available in early fall.

Although the Manufacturer's Suggested Retail Price for the CC-40 is $249.95, catalog prices range from $199.99 (Sears) through $189.95 (Educalc) to $189.00 (Elek-Tek). I have seen the CC-40 at local discount houses for as low as $179.95. The CC-40 packs a lot of "bang for the buck" at those prices.

MATRIX OPERATIONS WITH THE CC-40 MATHEMATICS MODULE - In V8N4P12/13 I reported that the execution speed of the prime factors program in the CC-40 Mathematics module was disappointing, and that the module had other deficiencies as well. I am happy to report that the matrix manipulation programs seem to be more carefully constructed. The capabilities are similar to those of the ML-02 and ML-03 programs in the Master Library module of the TI-59. In fact, the discussion of the use of the lower upper (LU) decomposition method is identical for the CC-40 Mathematics module and the TI-59 ML-02 programs.

Execution speed is substantially improved. The CC-40 finds the determinant for the third order matrix problem on page 12 of the manual for the TI -59 Master Library module in about two seconds, while the TI-59 requires sixteen seconds to complete the same problem. The CC-40 finds the determinant of a fifth order matrix in about six seconds, while the TI-59 requires about fifty-three seconds for the same problem using ML-02.

A deficiency of the CC-40 program is that the result is brought to the display with a BASIC Print command and the user cannot perform any chain calculations on the result without reentering the value. The reentering process necessarily drops any digits which were not displayed. The TI-59 solution is displayed in a manner such that chain calculations on the displayed result is possible. The loss in accuracy for the chain calculations with the CC-40 caused by the reentry can be duplicated with the TI-59 by performing EE-INV-EE to truncate to the displayed value before proceeding with user entered chain calculations. If the variable names of the solution were available the user could recall the solutions as a part of his keyboard BASIC chain calculations and retain the full accuracy; but the documentation with the CC-40 provides no information as to the variable names. To remedy this situation I have written a short demonstration program for solution of a system of linear equations (AX = B) which provides identification of variable names for at least some elements of the solution:

```
100 DIM A(8,9),C(8,8),B(8)
110 R = PI
120 INPUT "Enter Order of Matrix - ";N
150 CALL MI("A",A(,),1,N,N,0)
200 CALL AK("B",B(),1,N,0)
250 PRINT "Solving"
300 CALL MATS(A(,),C(,),B(),1,1,5,1,N,1,R)
350 IF R<>0 THEN 400
360 PRINT "MATRIX IS SINGULAR":PAUSE
400 FOR I = 1 TO N
410 X$ = "X" & STR$(I) & " = "
420 PRINT X$;A(I,1):PAUSE
430 NEXT I
999 STOP
```

Line 100 - The dimension statement sets up the array names to be used in the various subroutine calls. For reasons that are not very clear to me the array for the entry matrix A(m,n) must have one more column than the order of the problem if the MATS subroutine call at line 300 is to operate properly.

Line 110 - The dummy variable R will be used to indicate whether or not the input matrix is singular. See the discussion of the TEST variable on page 94 of the Mathematics Module manual.

Matrix Operations with the CC-40 Mathematics Module - (cont)

Line 120 - Provides operator control of the order of the problem to be solved.

Line 150 - This subroutine call provides for input and edit of the elements of the matrix A into a two dimensional array. See page 95 of the manual. The single line subroutine call provides a thorough set of prompts for entry and editing, including indication of the row and column on each element to be entered.

Line 200 - This subroutine call provides for input and edit of the elements of the vector B into a one dimensional array. See pages 85-86 of the manual. Again, the subroutine call also provides a thorough set of prompts..

Line 250 - This only provides a clear indication that the computer has changed from the edit mode to the solve mode.

Line 300 - This subroutine call provides the solution for the set of linear equations. See page 94 of the manual. The subroutine ends with the elements of the solution in the subscript 1 column of the A array, and with the inverse of the A matrix in array C. If the input A matrix was singular then R is changed to zero.

Line 350 - Tests the value of R to determine if the input matrix A was singular.

Line 360 - Displays an appropriate message if the input matrix was singular.

Lines 400 to 430 - Display the elements of the solution with appropriate annotation.

To illustrate use of the program use the problem on page 12 of the manual for the TI-59 Master library module:

1. Press RUN and ENTER. See the prompt "Enter Order of Matrix - ".

2. Press 3 and press ENTER. See the prompt "Enter A(1,1):"

3. Press 4 and press ENTER to insert the A(1,1) element. The computer accepts the input and returns with the prompt "Enter A(1,2):". Continue to enter the remaining elements of the matrix. Note that the CC-40 accepts the matrix elements by row in contrast with the TI-59 which accepted the elements by column. But note that there is nothing to remember since the MI subroutine call supplies the necessary prompts. When the last element A (3,3) has been entered the computer responds with the prompt "Edit?". If you choose to edit by responding with a Y the computer response is the prompt "Edit All Input?". If you respond with a N the computer response is "Enter Row To Be Edited:". You enter the row number and the computer response is another prompt "Enter Column to Be Edited:". You enter the column number and the computer response is "Enter A(i,j): Aij" where i and j are the row and column you selected, and Aij is the value which was entered for that element earlier. If you decide to edit that element you replace the displayed value with the desired one and press ENTER. If you decide not to change the element you simply press ENTER. In either case the computer responds with the prompt "Edit Other Elements?".

Matrix Operations with the CC-40 Mathematics Module - (cont)

4. When you have completed any editing of the A matrix the final N response to the edit prompts will cause the computer to move forward to the entry of the vector elements. The prompt message will be "Enter B(1)". You proceed to enter the elements of the vector in a manner similar to that used for the matrix. Again, you will be given an opportunity to edit. The important point is that all the prompts for the entry of both matrix elements and vector elements are provided by the module in response to the subroutine calls MI and AK.

5. When you have completed the editing process by responding with an N at the appropriate point the program immediately proceeds to solution of the problem, with the indication "Solving" in the display. When the solution is complete the computer response is the display "X1 = 4" if you entered the problem from page 12 of the Master Library correctly. Press ENTER as many times as needed to see the remainder of the solution.

6. After the display of the solution has been completed you may use keyboard BASIC (or you may add commands to the program) to read out other parameters, or the same parameters in other formats. The elements of the input matrix have been destroyed. The elements of the inverse of the input matrix appear in array C properly located; that is, the i,j element of the inverse can be recalled with the command PRINT C(i,j). For our example, the sequence PRINT C(2,2) will yield a ten digit display of .0416666667. The user can view additional digits with the command

```
PRINT USING ".##############";C(2,2)
```

to yield a fourteen digit display display of .04166666666667 ; or, in a technique similar to that used to observe the guard digits of the TI-59, the user can use the command

```
PRINT (C(2,2)-.04166)*100000
```

to yield a display of .666666667 .

7. If the user changes the sixth element in the argument for the MATS subroutine call from a 5 to a 4, then the program will only proceed through the calculation of the inverse of the input matrix. The elements of the inverse will appear in array C, again with the appropriate subscripts. The elements of the inverse will also appear in array A, but with the first and second columns interchanged. This is exactly the same orientation in which the inverse appears in a TI-59, where there is also an indication of the interchanged columns through observation of the pivoting index; that is, for the particular third order example used here TI-59 memory registers R17, R18 and R19 will contain the numbers 2, 1, and 3 respectively. I have been unable to find a way to recall the pivoting index from the CC-40 solution. Hopefully, this helps to explain the note in the discussion of "Inversion" on page 52 of the manual for the Mathematics module which states "..The inverse of A may be stored with its columns permuted and must be reentered for subsequent calculations." That statement is true if one uses the CALL "MAT" method to obtain the inversion. If one uses the CALL MATS method illustrated here then the columns in array A may (or may not) be permuted depending on the particular input matrix, but the inverse which appears in array C will not have permuted columns and can be used directly for further calculations.

A least squares polynomial curve fitting program using the techniques described here appears on the following page.

LEAST SQUARES POLYNOMIAL CURVE FIT WITH THE CC-40 MATHEMATICS MODULE

This program uses the same techniques described on the previous pages with the addition of a call of subroutine AU (see pages 87-88 of the manual) to provide entry of the data pairs into two one-dimensional arrays. Again, the subroutine call provides valuable prompts. I believe that the prompts with this program are sufficient such that no detailed program description is required. There is one idiosyncrasy of the prompts for editing the entry of the data pairs which is described on page 18 of this issue.

```
100 DIM A(8,9),B(8),C(8,8),H(8),X(50),Y(50)
110 INPUT "Number of Data Pairs? ";K
120 CALL AU("X","Y",X(),Y(),1,K,0)
130 INPUT "Degree of Polynomial? ";N
140 PRINT "Solving"
150 N=N+1:R=1:P$="":Q$=""
160 FOR I=1 TO N:FOR J=1 TO N
170 A(I,J)=0:NEXT J
180 B(I)=0:NEXT I
190 FOR L=1 TO K
200 H(1)=1
210 FOR I=2 TO N
220 H(I)=H(I-1)*X(L):NEXT I
230 FOR I=1 TO N:FOR J=1 TO N
240 A(I,J)=A(I,J)+H(I)*H(J):NEXT J
250 B(I)=B(I)+H(I)*Y(L):NEXT I
260 NEXT L
270 CALL MATS(A(,),C(,),B(),1,1,5,1,N,1,R)
280 IF R<>0 THEN 300
290 PRINT "Matrix is singular":PAUSE:GOTO 470
300 FOR I=1 TO N
310 X$="A"&STR$(I-1)&" = "
320 PRINT X$;A(I,1):PAUSE:NEXT I
330 INPUT "Display Residuals (Y/N)? ";P$
340 S1=0
350 FOR I =1 TO K
360 Y1=A(N,1)
370 FOR J=(N-1) TO 1 STEP -1
380 Y1=A(J,1)+X(I)*Y1:NEXT J
390 D1=Y(I)-Y1
400 IF P$="y" OR P$="Y" THEN 410 ELSE 430
410 A$="d"&STR$(I)&" = "
420 PRINT A$;D1:PAUSE
430 S1=S1+D1*D1:NEXT I
440 PRINT "Standard Error = ";SQR(S1/(K-N)):PAUSE
450 INPUT "Try a Different Degree (Y/N)? ";Q$
460 IF Q$="y" OR Q$="Y" THEN 130
470 STOP
```

LANGUAGES ON THE CC-40 - V8N4P12 discussed the various languages which are available with the CC-40 by using the CALL SETLANG command. The Mathematics and Statistics modules support English, German, and French. The Finance module supports only English and German.

A PROMPTING ANOMALY IN THE MATHEMATICS MODULE FOR THE CC-40

There is an apparent error in that portion of the Mathematics module for the CC-40 which provides for editing of the entry of two one-dimensional arrays. An example occurs when running the Cubic Splines program. Go to page 31 of the manual and follow the example through step 13. At that point the display will read "Edit?". Do not proceed to step 14. Rather respond with a Y for yes and press ENTER. The display will prompt with the message "Edit All Input?". This time respond with an N for no and press ENTER. The display will prompt with the message "Enter Element to Be Edited:". Press 3 and ENTER and see "Enter X(3): 1" in the display. The 1 was loaded into that location by step 10. Press ENTER again assuming that you did not want to edit the value in X(3). The display changes to "Enter X(3): .8413". You would have expected the display to read "Enter Y(3): .8413". Although the indication of which element is available to be edited is incorrect, the value displayed is that which was stored in Y(3) at step 11. There is no harm done by the improper indication, but it will surprise an unwary operator. The same effect can be seen when using the AU routine on page 87 in the manual. Users of the Least Squares Polynomial Curve Fit on page 17 of this issue can expect to encounter this anomaly.

MEMORY PROTECTION ON THE CC-40 - V9N1P19 discussed memory protection on the CC-40 during replacement of the batteries. Maurice Swinnen had reported successful changes without losing memory when the time to replace was less than a minute. In late May I purchased an AC Adapter for my CC-40 from Educalc (Stock No. AC-9201, $14.95 plus shipping and handling). Just in time! In mid June the battery low indicator appeared on my CC-40. I connected the AC adapter, replaced the batteries at a leisurely pace, and found no loss of memory. Further experiments showed that the CC-40 will work satisfactorily with either the batteries or the AC adapter, whichever is available. If the batteries are installed, and you connect the AC adapter cable, but do not plug into AC power, the CC-40 still runs from battery power.

That feature is not available with some other portables. The Radio Shack Model 100 mechanization disconnects the batteries when the AC power adapter is connected. The instructions are very explicit--first, you connect the adapter to an AC outlet, then you connect the adapter cable to the computer. If the adapter cable is connected to the computer without a connection to AC power the computer will not operate. Memory is held up by the NiCad memory retention battery. This would seem to permit a condition in which inadvertently leaving the adapter connected to the computer and not connected to AC power could eventually cause a loss of memory as the NiCad battery runs down. I have written to Radio Shack for information. I have also written to TI for approval of the use of the AC Adapter during battery replacement.

ACCURACY OF THE SOLUTIONS FOR SYSTEMS OF LINEAR EQUATIONS

Several different programs for solution of systems of linear equations with the TI-59 have been discussed in this issue. How does the user decide which program to use? The discussion in previous pages of this issue has addressed considerations such as user friendliness, system size, and the like. Another important issue is accuracy of the solution, and we will see that the Ohlsson program and its derivatives are less accurate. How do we measure accuracy? George Thomson provided some thoughts on that subject.

Here are some practical tips for testers of matrix inversion programs. The workhorse test matrices are the "Hilberts"; the first row is 1, 1/2, 1/3, ..., the second row is 1/2, 1/3, 1/4, ..., the third row is 1/3, 1/4, 1/5, ..., and so on. Their inverses have horrendously huge integers and are available. See for example, I. R. Savage and E. Lukacs, National Bureau of Standards AMS No. 39, pp. 107-108 (1954) for the inverses up to 10 x 10. The seventh row, seventh column of the 10 x 10 inverse is 348 06739 96800 . Others are almost as large. The "sub-Hilberts" with the first row 1/2, 1/3, 1/4, ..., the second row 1/3, 1/4, 1/5, ..., and so on are even harder to invert correctly. I suggest as a guinea pig the 7 x 7 sub-Hilbert, with ones on the right hand side:

1/2	1/3	1/4	1/5	1/6	1/7	1/8	1
1/3	1/4	1/5	1/6	1/7	1/8	1/9	1
1/4	1/5	1/6	1/7	1/8	1/9	1/10	1
1/5	1/6	1/7	1/8	1/9	1/10	1/11	1
1/6	1/7	1/8	1/9	1/10	1/11	1/12	1
1/7	1/8	1/9	1/10	1/11	1/12	1/13	1
1/8	1/9	1/10	1/11	1/12	1/13	1/14	1

The exact solution of the simultaneous equations is 56, -1512, 12600, -46200, 83160, -72072, and 24024. All the elements of the inverse are integers, the largest is 6915 58560. The most practical measure of the accuracy of a solution is to calculate the relative error, i.e., (answer - true result)/(true result) for each element and take the largest value. This measure is related to the number of meaningful significant digits in the results.

Readers who are familiar with 52 Notes will recall that V2N12P5 described the use of the Hilbert matrices ($A_{ij} = 1/(i+j-1)$) as a test of the ability of a matrix inversion routine to handle ill-conditioned matrices.

All the ML-02 deriviatives yield identical results. Therefore, description of the results from any one of the ML-02 programs defines the accuracy of all of them. Similarly, the Ohlsson program and the derivatives by Prins and Ristanovic yield identical results, and a single description of results will suffice for all three. For the 7 x 7 sub-Hilbert test suggested by George Thomson the various algorithms yield the following results:

Accuracy of the Solutions for Systems of Linear Equations - (cont)

Ohlsson/ Ristanovic/ Prins	TI-59 ML-02	Anderson Row Reduction	Nick and Ristanovic "Gauss"	CC-40 Mathematics Module
Programbiten		PPX V4N5P8	V7N6P13	V8N5P14
55.9233	56.0082	56.0081	56.0076	56.000032
-1510.2276	-1512.1896	-1512.1865	-1512.1732	-1512.000787
12587.0911	12601.3863	12601.3511	12601.2536	12600.0059
-46157.9673	-46204.5344	-46204.3822	-46204.0623	-46200.0192
83091.9632	83167.3718	83167.0718	83166.5503	83160.0311
-72018.4333	-72077.8274	-72077.5542	-72077.1412	-72072.0246
24007.6425	24025.7860	24025.6926	24025.5659	24024.0074
1.37E-3	1.46E-4	1.45E-4	1.35E-4	5.71E-7

The ML-02 solution, the Anderson row reduction solution, and the Nick/Ristanovic solution yield nearly identical results from an accuracy standpoint. The Ohlsson program and its derivatives yield a solution that is an order of magnitude less accurate. The CC-40 yields a much more accurate solution than any of the TI-59 programs. This is somewhat surprising since the manual for the CC-40 Mathematics Module indicates that the method of solution is the same as for ML-02, and the CC-40 carries only one additional digit. To attain that level of accuracy with the CC-40 it is necessary to calculate the matrix elements in the program. If one tries to enter the values from the keyboard then the quirk described in V8N3P5 takes over, and only ten digits are used. The error in the resulting solution is 6.94E-3. One can obtain similar errors with ML-02 by pressing EE-INV-EE after calculating each reciprocal, and before entering the element for use by the program.

As an additional comparison of the capability of the CC-40 I entered an old "workhorse" simultaneous equation solution into the CC-40 and several other home/personal computers. Gene Friel also provided a solution using the Math-Pac Application Module with the HP-41C which uses a Gauss elimination method. The results, again using George Thomson's 7 x 7 test were:

HP-41	Color Comp	Apple II+	CC-40	Model 100
56.6667	55.5926	56.1869	56.000198	55.999816
-1527.3832	-1502.465	-1516.2347	-1512.00461	-1511.99596
12712.2414	12529.8262	12630.3122	12600.0337	12599.9716
-46566.4960	-45969.5924	-46297.3343	-46200.1101	-46199.9102
83755.0102	82784.5266	83315.8117	83160.1785	83159.8577
-72541.8140	-71774.7464	-72193.5351	-72072.1406	-72071.8899
24167.8491	23932.811	24060.8602	24024.0429	24023.9669
1.19E-2	7.27E-3	3.34E-3	3.53E-6	3.28E-06

The superiority of the CC-40 and Radio Shack Model 100, both 14 decimal digit computers, is obvious. But this solution on the CC-40 is an order of magnitude less accurate than that from the program in the Mathematics module.

Accuracy of the Solutions for Systems of Linear Equations - (cont)

For reference the common program used to evaluate the four computers is:

```
100 DIM A(10,10),B(10)
110 INPUT "Enter order";N
120 N = N-1
130 K=0
135 FOR I = 0 TO N
140 FOR J = 0 TO N
145 A(I,J)=1/(J+K+2)
150 NEXT J
155 B(I)=1
160 K=K+1
165 NEXT I
200 FOR K = 0 TO N
210 P = A(K,K)
250 FOR J = K TO N
260 A(K,J) = A(K,J)/P
270 NEXT J
280 B(K) = B(K)/P
290 FOR I = 0 TO N
300 IF I = K THEN 360
310 F = A(I,K)
320 FOR J = K TO N
330 A(I,J) = A(I,J) - F*A(K,J)
340 NEXT J
350 B(I) = B(I) - F*B(K)
360 NEXT I
370 NEXT K
490 FOR I = 0 TO N
500 PRINT "X"+STR$(I)+" = "; B(I)
510 NEXT I
600 END
```

Lines 130 through 165 provide automatic entry of the appropriate sub-Hilbert problem as defined by George Thomson on page 18. If you wish to use the program for other solutions simply replace those steps with appropriate steps to accept the appropriate matrix elements.

MORE SUBPROGRAMS FOR THE CC-40 STATISTICS CARTRIDGE - Experiments show that the CC-40 Statistics cartridge has a subprogram for input and edit of a two-dimensional array which is very similar to that in the Mathematics cartridge. Even the call MI is the same. The prompts are the same as those described on V8N5P15 except that at the end of an edit of all input the Statistics cartridge implementation leaves the subprogram, while the Mathematics cartridge implementation returns for additional editing.

There are obviously other unlisted subprograms in the Statistics cartridge. A call for an AK subprogram for input and entry of a one-dimensional array as in the Mathematics cartridge yields the error message "Program not found". A call for an AU subprogram for input and edit of two one-dimensional arrays as with the Mathematics cartridge yields the error message "Illegal Syntax", which suggests there is a subprogram in the Statistics cartridge with the AU name.

SORTING ON THE CC-40 - The Statistics cartridge for the CC-40 has a shell sort subprogram. The program requires that the elements to be sorted have already been assembled into a one dimensional array. The following program provides entry of data into an array, sorting, and display of the sorted elements:

```
100 DIM X(100)
110 INPUT "Number of Elements? ";K
120 FOR I = 1 TO K
130 INPUT "Enter X("&STR$(I)&"): ";X(I)
140 NEXT I: PRINT "Press <ENTER> to Sort":PAUSE
150 PRINT "Sorting"
160 CALL SORT(X(),K)
170 FOR I = 1 TO K
180 PRINT "SX("&STR$(I)&") = ";X(I)
190 PAUSE: NEXT I
200 END
```

This program is much faster than the sorting program in the Math/Utilities module for the TI-59 (MU-06). The CC-40 sorts 60 random numbers in 31 seconds. The TI-59 takes 4 minutes 55 seconds.

FACTORIALS WITH THE CC-40 MATHEMATICS MODULE

Factorials can be calculated with the Mathematics module of the CC-40 by recognizing that N! = Gamma(N+1). With this technique the CC-40 with the Mathematics module installed will return Ln(Gamma(70)) = 226.1905483 in about one second and pressing ENTER will immediately yield Gamma(70) = 1.711225E+98 which is equal to 69!. By comparison the ML-16 program on the TI-59 takes about 16 seconds to obtain the equivalent answer; but, the MU-11 program in the Math/Utilities module for the TI-59 will find 69! in about four seconds with the Gamma function method. The CC-40 can obtain factorials up to 85! = 3.31424E+126 with this method.

CC-40 STATUS - In late November I called Educalc for information on peripherals and supplies for the CC-40. I was told that TI was discontinuing the CC-40. A call to the TI Consumer Hotline, 800-842-2737, confirmed that the CC-40 development had been stopped. There will be repair support for CC-40 hardware both in and out of warranty, but no new products will be released. We will continue to provide coverage of the CC-40 and peripherals in TI PPC Notes. Some peripherals continue to be for sale at the TI exchange centers. Other sources for supplies are available. I have used the Radio Shack ink cartridges successfully in the Printer/Plotter.

ADVANCED ELECTRICAL ENGINEERING module for the the CC-40.

Review by Maurice E.T. Swinnen.

This is the fourth module for the CC-40 I have seen so far, and all prove to be of an extraordinary quality and usefulness. Although I feel a little at home with the Mathematics module, I certainly feel unqualified to review either the Statistics or the Finance module. But Electrical Engineering is a field I eat, drink, and sleep at least ten hours a day, and I have been doing this for the last fourty years. Boy do I wish I had this CC-40 and this EE-module when I started, eons ago! The closest I ever came to it was a slide rule or a Monroe mechanical calculating machine.

The module contains the following programs:

1. Active second-order multiple-feedback (one op-amp) low-pass, high-pass and band-pass filters.
2. Bode-Nyquist calculations.
3. Roots of a polynomial. (Finds all real and complex roots of up to a 20th degree polynomial in one variable with real coefficients)
4. Discrete Fourier transform. (Transforms a sampling of the time domain to the frequency domain and also performs the inverse transform from the frequency domain to the time domain. Six windowing techniques are available for sidelobe suppression.)
5. Passive low-pass filters. (Very handy in very-high frequency computations. Allows design of both Tchebycheff and Butterworth low-pass filters.)
6. Phase-lock loop calculations.(Complete! For both active and passive types)
7. Series/Parallel impedance conversions.(I am not so crazy about this one. Bill Beebe wrote a simpler and more useful one for the TI-59.)
8. Signal detection. (Calculates signal-to-noise ratio, probability of false alarm, probability of detection given any two of the three, and the ratio of the standard deviation of the two signals.)
9. S to and from Y, H, and Z parameter conversion. (This program has Gary Morella written all over it. Gary is no longer working at TI, although the manual of this module names him in the credits list. In my opinion this alone is worth the price of the module. I have seen several attempts to write a program of this magnitude for the TI-59, but they all had serious shortcomings, mostly due to the limited memory available. The only program that did things satisfactorily is contained in the EE-module for the TI-88 and it was written by, you guessed it, Gary Morella. Unfortunately TI made only twenty samples of the TI-88 EE-module, which makes them even rarer than hen's teeth.)

Besides these programs there are several subprograms. They are shared by the main programs, in about the same manner as subroutines are. But they may also be called from a user-written program in RAM. As an example of this technique, I have enclosed at the end a program that uses two subprograms: PR and RP. They do the conversion of Rectangular to Polar and vice-versa for you. They are, of course, built into the firmware of the TI-59, but not in the CC-40. The program is fully prompting, which makes mistakes almost a thing of the past. I admit, with some editing, one could write it on fewer lines, combining several statements on one line each time. But for the gain of a few bytes, readablity would suffer in the process. In this program a technique is used, unique to the TI-99/4A (the home computer) and the CC-40: one-key response. Most computers require you to place your answer in the display, followed by pressing the ENTER key. Here it is possible that simply pressing Y or N allows you to select program sequence. See, for example, line 120. It displays the message"Rectangular to Polar? Y/N", and assigns A$ to KEY$. It waits for your response. If you press the N-key, either in lower or upper case, line 130 sends you to line 320. If you press the Y-key (or any other key for that matter) the program continues with line 140. It is very similar to the user-defined keys in the TI-59, except that all the keys can be used and that the user can select which ones and their effect.

To conclude, this module is well worth investing in, if your game is electrical engineering. If TI would just see fit to finally produce some peripherals for this portable machine I, and a lot of my friends, would be very happy to clear out the nine programs we all have stored in permanent memory. And we finally would be able to sleep tightly again, free of nightmares that someone might type the dreaded word NEW on the keyboard.(For those not familiar with Basic, we will let you in on the joke: NEW, followed by ENTER, wipes out everything in RAM, program and variables, and the mere mention of the word is enough to give me apoplexy.)

```
100  DISPLAY AT(2)"EE module in place? Y/N":A$=KEY$
110  IF A$="N" OR A$="n" THEN 360
120  DISPLAY AT(2)"Rectangular to Polar? Y/N":A$=KEY$
130  IF A$="N" OR A$="n" THEN 320
140  DISPLAY AT(2)"X-coordinate?";
150  ACCEPT AT(18)VALIDATE(NUMERIC)BEEP,X
160  DISPLAY AT(2)"Y-coordinate?";
170  ACCEPT AT(18)VALIDATE(NUMERIC)BEEP,Y
180  CALL RP(X,Y,M,A)
190  DISPLAY AT(2)"Magnitude=";M:PAUSE
200  DISPLAY AT(2)"Angle=";A;"degrees":PAUSE
210  GOTO 120
220  DISPLAY AT(2)"Polar to Rectangular? Y/N":A$=KEY$
230  IF A$="N" OR A$="n" THEN 350
240  DISPLAY AT(2)"Magnitude?";
250  ACCEPT AT(18)VALIDATE(NUMERIC)BEEP,M
260  DISPLAY AT(2)"Angle in degrees?";
270  ACCEPT AT(20)VALIDATE(NUMERIC)BEEP,A
280  CALL PR(M,A,X,Y)
290  DISPLAY AT(2)"X-coordinate=";X:PAUSE
300  DISPLAY AT(2)"Y-coordinate=";Y:PAUSE
310  GOTO 220
320  DISPLAY AT(5)"Exit program? Y/N":A$=KEY$
330  IF A$="N" OR A$="n" THEN 220 ELSE END
340  DISPLAY AT(5)"Exit program? Y/N":A$=KEY$
350  IF A$="N" OR A$="n" THEN 120 ELSE END
360  DISPLAY AT(4)"Insert EE module, please!":PAUSE 4
370  END
```

<u>EDITOR'S NOTE</u> - My sentiments about the lack of peripherals are the same as Maurice's. I am using the Mathematics module and have a set of interacting programs which perform polynomial regressions, compute residuals, solve sets of linear equations by various methods, and the like. One inadvertent NEW would be a disaster. The CC-40 is beginning to get some favorable press. In the article "Choosing a Notebook Computer" in the January 1984 issue of Creative Computing author David Ahl discusses price versus performance:

> "... But perhaps most interesting are the five machines that fall below the curve, and thus represent relative bargains. At the low end is the TI CC-40. For professionals, students, and engineers, this is an unbeatable machine at only $250, frequently discounted to well under $200. ... "

SIMULTANEOUS EQUATIONS WITH THE CC-40 MATHEMATICS MODULE - P. Hanson

V8N5P14-16 discussed the matrix operations programs in the CC-40 mathematics module. V8N6P19 reported excellent results using those techniques to solve the 7x7 sub-Hilbert, but presented the results only to enough digits to establish the relative error. James Walters proposed another method of error evaluation, that is, multiplying the solution vector by the original matrix and comparing the result with the input unity vector. To use that method it was important to use all of the digits of the solution. No difficulties were found in doing that with any of the TI-59 solutions, or for any of the solutions on personal computers using the program on V8N6P20; but the Mathematics module solution from the CC-40 would only yield the 9 to 11 digits shown at the right. After a lot of agonizing over items such as whether my application of "PRINT USING" was proper, and the like, I finally found that the truncated result from the CC-40 Mathematics module is a direct result of the method of solution.

```
   56.00003229
-1512.000787
12600.00591
-46200.0192
 83160.0311
-72072.0246
 24024.007436
```

I had assumed that the method of solution from the CC-40 Mathematics module and from the TI-59 Master Library (ML-02) was the same. The discussions under "Method Used" on page 13 of the Master Library manual and on pages 53-54 of the CC-40 Mathematics module manual are identical. Experiments show that the methods for solution of linear equations must be quite different. The ML-02 solution on the TI-59 does not seem to make direct use of the inverse of the matrix. The CC-40 solution seems to obtain the inverse, and simply multiply the inverse by the vector to get the solution. Where the vector is the unity vector as in our 7x7 sub-Hilbert test problem, the solution may be obtained by simply adding up the rows of the inverse matrix. For the 7x7 problem, the seventh (or bottom row) of the inverse matrix is listed at the right. Now, if you sum the terms in the row from the top, as would be reasonable for a loop in the computer, then you will obtain exactly the solution for the seventh element in the table at the top of the page. Similar results can be obtained for the other elements of the solution by reading out the elements of the inverse matrix. You must remember to always truncate each intermediate sum to the fourteen digit limit of the computer. The truncated output arises because the last summation is between two numbers of about 41 million but of opposite sign, yielding an answer of about 24 thousand. The same sort of result can be obtained with the ML-02 programs by not solving simultaneous equations with ML-02 Program E, but rather obtaining the inverse matrix with ML-02 Program B', and summing the rows. The printout on the left below is the ML-02 solution using the standard method. The printout at the right was obtained using the inverse matrix method. Again, the truncation effect is evident. Until TI choses to release the program details of the CC-40 Solid State Modules we can only continue to try to understand through experimentation.

```
C(7,1) =        168,168.045 045 95
C(7,2) =     -4,036,033.121 075
C(7,3) =     30,270,248.620 461
C(7,4) =   -100,900,829.259 5
C(7,5) =    166,486,368.937 2
C(7,6) =   -133,189,095.560 0
C(7,7) =     41,225,196.345 304
```

```
56.0081897448        56.0081896
-1512.189567429    -1512.189528
12601.38627848      12601.38624
-46204.53435755    -46204.5337
83167.37180486      83167.3708
-72077.82742612    -72077.8273
24025.7860121       24025.78576
```

NUMERIC REPRESENTATION IN THE TI-99/4 AND CC-40 - Laurance Leeds

In V9N4P7 Myer Boland reported that he could recover fourteen digits of pi on the TI-99/4 with the equation P = 4000*ATN(1). If one tries to convert the answer from 1000*pi to pi by dividing by 1000, then the end result reverts to a twelve digit value. The same results were reported for the CC-40. These results follow directly from the radix 100 arithmetic mechanization (see page F-2 of the CC-40 Manual or page III-13 of the TI-99/4 manual.

Both machines use seven radix 100 bytes for the mantissa. This is just another way of saying that the arithmetic is performed using seven blocks, each of two decimal digits, with the value of each block ranging from zero through 99. The exponent is selected so that the decimal point of the mantissa immediately follows the most significant digit. In short, the arithmetic is in base 100. The mechanization explains why

```
   40 * ATN(1) gives 14 digits of pi
  400 * ATN(1) gives 12 digits of pi
 4000 * ATN(1) gives 14 digits of pi
40000 * ATN(1) gives 12 digits of pi
```

and also why (4000*ATN(1))/1000 gives a 12 digit result. We will see that the twelve digit results are thirteen digit results which include a trailing (non-displayed) zero. In the same manner the thirteen digit TI-59 yields an apparent twelve digit value, but actually a correctly rounded 13 digit value for pi. Consider two representative calculations:

```
          400 * ATN(1)

ATN(1) = 78.53 98 16 33 97 45 x 100^-1
   400 =  4.00 00 00 00 00 00 x 100^+1
   -----------------------
   3 12
      2 12
         3 92
            0 64
               1 32
                  3 88
                     1 80
   -----------------------
   3 14.15 92 65 35 89 80 x 100^0
```

```
          4000 * ATN(1)

       = 78.53 98 16 33 97 45 x 100^-1
  4000 = 40.00 00 00 00 00 00 x 100^+1
   -----------------------
   31 20
      21 20
         39 20
             6 40
               13 20
                  38 80
                     18 00
   -----------------------
   31 41.59 26 53 58 98 00 x 100^0
```

Rounding to seven radix 100 digits yields:

```
3 14.15 92 65 35 90                31 41.59 26 53 58 98
```

and scaling the mantissa and exponent yields:

```
3.14 15 92 65 35 90   x 100^+1     31.41 59 26 53 58 98   x 100^+1
```

A similar exercise for dividing 4000*ATN(1) by 1000 is left to the reader.

Numeric Representation in the TI-99/4 and CC-40 - (cont)

Although both the CC-40 and the TI-99/4 allow entry of a fourteen digit base ten number, the storage of the number depends upon the location of the decimal point. For example,

The entry	1234567.8912345
translates to	$1.23\ 45\ 67\ 89\ 12\ 34\ 50 \times 100^3$
which rounds to	$1.23\ 45\ 67\ 89\ 12\ 35 \times 100^3$

The rounding of the seventh radix 100 digit (the 13th and 14th base 10 digits) in accordance with the value of the eighth radix 100 digit occurs immediately after the calculation. This precludes the use of the seventh block for programs which require all of the base 10 digits to be exact, as in multi-precision work.

A safe rule is to program as though the machine is an exact twelve digit calculator, never permitting any overflow into the seventh block if this information can affect the result.

Since the rounding also occurs in division, modulo division may not give the desired result. For example, we ask for the residue when N = 12345678901563 is divided by 547. Since N is a fourteen digit number we use modulo division, say with the algorithm

```
N mod M = N - M * INT(N/M)
```

Both the CC-40 and the TI-99/4A return -2 as the answer. The correct answer is 545. The base 100 arithmetic is not the culprit; the rounding is. While it is true that in this example N is congruent to -2 mod 547, this result could surely mess up program calculations.

Editor's Note: My Radio Shack Model 100 which also does 14 digit arithmetic gets the correct answer using the algorithm above.

MORE ON DATA INPUT TO PERSONAL COMPUTERS - Larry Leeds writes "It came as a shock to discover that the machines would alter input data! Of no useful significance, but of passing interest, is the fact that one can enter a 16 digit base 10 number and the machine will examine the 15th and 16th digits to see if the 7th base 100 block should be rounded.

```
Enter:  A = 123456.7891234599
        PRINT A - 123456
```

The displayed result will be .78912346 "

Editor's Note: Even more entry tricks are available. The CC-40 has an eighty character line which can be scrolled to view any 31 characters.

```
Enter:  B=12345678901234567890123456789012345678901234567890
        PRINT B
```

and see 1.234568E+49 in the display. Also,

```
Enter:  C=.000000000000000000000123456789
        PRINT C
```

and see 1.234568E-21 in the display.

PRINTING WITH THE HX-1000 - P. Hanson

V9N4P26 reported that I had received an HX-1000 Printer/Plotter for use with my CC-40, but that I had been unable to establish communication with my CC-40. With the assistance of the Customer Service Center in Tampa I was able to isolate the problem to the CC-40. Apparently, there was some problem with the hex-bus in the engineering models. As part of the exchange for a working model I also upgraded to the 18K version of the CC-40. The extra memory will permit solution for higher order matrices, a subject I will cover in a future issue.

The HX-1000 permits two print modes: either 18 characters per line or 36 characters per line. The 36 character mode permitted translation of an old calendar program for the Model 100 for use with the CC-40. A full size printout is:

```
   FEBRUARY   1900

 SUN MON TUE WED THU FRI SAT

                  1   2   3
  4   5   6   7   8   9  10
 11  12  13  14  15  16  17
 18  19  20  21  22  23  24
 25  26  27  28
```

Printout of a single month requires about 21 seconds--much slower than the time required with the TI-59 when using one of the fast mode programs. Of course, the printer/plotter output can be expected to be slow since it draws each letter. In the calendar program the month and year are printed in the 18 character per line mode, and the remainder in the 36 character per line mode.

Listings can be obtained in either mode. The listing for the calendar program at the right was printed in the 36 character mode and enlarged for easier reading. Preliminary tests indicate that the automatic printing from the Solid State Software modules will be in the compressed mode. In the next issue I will demonstrate the plotting feature.

```
500 DIM Q(12),C$(5)
505 DATA 31,28,31,30,31,30,31,31,30,
31,30,31
510 FOR I=1 TO 12:READ Q(I):NEXT I
515 DATA "JANUARY  ","FEBRUARY ","MA
RCH    ","APRIL    ","MAY      ","JU
NE     "
520 DATA "JULY     ","AUGUST   ","SE
PTEMBER","OCTOBER  ","NOVEMBER ","DE
CEMBER "
525 INPUT "Enter Month (1-12): ";M
535 IF M<1 OR M>12 THEN 525
540 INPUT "Enter Year (>1582): ";R
545 IF R<1583 THEN 540
550 IF R-4*INT(R/4)=0 THEN Q(2)=29
555 IF R-100*INT(R/100)=0 THEN Q(2)=
28
560 IF R-400*INT(R/400)=0 THEN Q(2)=
29
565 R1=R-1:R2=R+INT(R1/4)-INT(R1/100
)+INT(R1/400)
570 FOR I=0 TO M-1:R2=R2+Q(I):NEXT I
575 D1=R2-7*INT(R2/7)
580 RESTORE 515:FOR I=1 TO M:READ M$
:NEXT I
585 OPEN #1,"10".OUTPUT
590 PRINT #1,TAB(3);M$;" ":R
600 PRINT #1,CHR$(18);
615 PRINT #1," SUN  MON  TUE  WED
THU  FRI  SAT"
620 PRINT #1
625 C$(0)=RPT$("     ".D1)
630 FOR I=1 TO (7-D1):C$(0)=C$(0)&"
  "&STR$(I)&" ":NEXT I
635 PRINT #1,C$(0)
640 C$(0)="":C$(4)="":C$(5)=""
645 I=8-D1
650 FOR K=1 TO 5
653 IF K=1 THEN C$(K)="   " FLSE C$(
K)="  "
655 FOR L=1 TO 7
660 IF I>8 THEN B$="   " FLSE B$="
  "
662 IF L=7 THEN B$=""
665 C$(K)=C$(K)&STR$(I)&B$
670 I=I+1:IF I>Q(M)THEN 700
675 NEXT L
680 NEXT K
700 FOR I=1 TO 5:PRINT #1:PRINT #1,C
$(I):NEXT I
705 PRINT #1
710 CLOSE #1:GOTO 525
```

MODULO 210 SPEEDY FACTOR FINDER IN BASIC - Laurance Leeds

V8N4P12 reported that the speed of the prime factor program in the Mathematics module for the CC-40 was only ten to forty percent faster than the fastest TI-59 programs, and substantially slower than the speeds reported for the HP-41. Laurance Leeds recently obtained a Radio Shack Model 100. One of his first programs was a modulo 210 factor finder which yields some truly impressive execution times.

The program at the right is a modification of Laurance's program to accommodate the single line display of the CC-40.

```
100 DIM X(12),Y(12),E(53)
110 DATA 2,3,5,7,11,2,4,2,4,6,2,6,4,
2,4,6,6,2,6,4,2,6,4,6,8
120 DATA 4,2,4,2,4,8,6,4,6,2,4,6,2,6
,6,4,2,4,6,2,6,4,2,4,2,10,2,10
130 FOR I=1 TO 53:READ E(I):NEXT I
140 PRINT "Modulo 210 Factors Progra
m":PAUSE 2
200 INPUT "N = ";N:N0=N
210 PRINT "Busy Factoring"
220 FOR I=1 TO 5
230 D=E(I)
240 IF INT(N/D)*D=N THEN GOSUB 400
250 NEXT I
300 IF N/D<=SQR(N)THEN X(K)=N:Y(J)=1
:GOTO 500
310 FOR I=6 TO 53
320 D=D+E(I)
330 IF INT(N/D)*D=N THEN GOSUB 400
340 NEXT I
350 GOTO 300
400 X(K)=D:S=S+1:N=N/D
410 IF N=1 THEN X(K)=D:Y(J)=S:GOTO 5
00
420 IF INT(N/D)*D=N THEN 400
430 Y(J)=S:K=K+1:J=J+1:S=0:RETURN
500 DISPLAY BEEP:FOR I=0 TO 12
510 IF X(I)=0 THEN 600
520 PRINT "F("&STR$(I+1)&") ="&STR$(
X(I))&"^"&STR$(Y(I))
530 PAUSE:NEXT I
600 PRINT "N was ";
610 PRINT USING"###############",N0:P
AUSE
620 FOR I=0 TO 12:X(I)=0:Y(I)=0:NEXT
 I : S = 0
630 J=0:K=0:GOTO 200
```

After entering the program, press RUN and see the words "Modulo 210 Factors Program" in the display after about four seconds. During that time the increments used in the program are transferred from the data statements to the E array. In a few more seconds the prompt "N = _" appears in the display. Enter the value to be factored and press ENTER. The display will read "Busy Factoring" until the process is complete. Then, the display will contain the first factor and multiplicity. Press ENTER to display the remaining factors. When the last factor has been displayed, one more ENTER will return the input value to the display. Another enter prepares the program for another problem and stops with the prompt "N = _" in the display.

Sample execution times for programs with comparable capability of twelve digits or more using the same benchmark problems used previously are:

Program/machine	111111111111	987654321	9999999967
TI-59 M/U Module	43 sec	215 sec	
TI-59 13 Digit Mod 210	34 sec	61 sec	3 hr 15 min
CC-40 Mathematics Module	11 sec	41 sec	1 hr 55 min
CC-40 (this program)	4 sec	9 sec	23 m 15 sec
Model 100 program (next page)	3 sec	7 sec	18 m 08 sec

Modulo 210 Speedy Factor Finder in BASIC (cont)

The program for the Model 100 is reproduced below. Laurance wrote an even faster factor finder program which did not recall the increments from an array, but rather used in-line techniques such as those used in the faster TI-59 programs. That program will declare 9999999967 to be prime in only 15 minutes 11 seconds. If you would like a copy of that program send a SASE. Finally, a reduced copy of the CC-40 program is presented below to obtain a comparison of legibility. CC-40 users are invited to comment.

Model 100 Program

```
100 DIM R(12), S(12), E(53)
110 DATA 2,3,5,7,11,2,4,2,4,6,2,6,4,2,4,6,6,2,6
,4,6,2,4,6,2,6,6,4,2,4,6,2,6,4,2,4,2,10,2,10
,4,2,6,4,6,8,4,2,4,2,4,8,6
120 FOR I = 1 TO 53:READ E(I):NEXT I
130 PRINT"FACTORS, N=14 DIGITS,MODULO 210 PGM"
200 PRINT:INPUT"N=";N
205 T1$=TIME$
210 PRINT "BUSY FACTORING"
220 FOR I = 1 TO 5
230 D = E(I)
240 IF INT(N/D)*D=N THEN GOSUB 700
250 NEXT I
300 IF N/D<=SQR(N) THEN R(K)=N:S(J)=1:GOTO 750
310 FOR I = 6 TO 53
320 D = D + E(I)
330 IF INT(N/D)*D=N THEN GOSUB 700
340 NEXT I
350 GOTO 300
700 R(K)=D:S=S+1:N=N/D
710 IF N=1 THEN R(K)=D:S(J)=S:GOTO 750
730 IF INT(N/D)*D=N THEN 700
740 S(J)=S:K=K+1:J=J+1:S=0:RETURN
750 T2$=TIME$
800 H=0
810 IF R(H) = 0 THEN 900
820 PRINT R(H)CHR$(94)S(H),
830 IF R(H+1) = 0 THEN 900
840 PRINT TAB(19)R(H+1)CHR$(94)S(H+1)
850 H=H+2:GOTO 810
900 PRINT:PRINT T2$:PRINT T1$
910 END
```

CC-40 Program

```
100 DIM X[12],Y[12],E[53]
110 DATA 2,3,5,7,11,2,4,2,4,6,2,6,4,
2,4,6,6,2,6,4,2,6,4,6,8
120 DATA 4,2,4,2,4,8,6,4,6,2,4,6,2,6
,6,4,2,4,6,2,6,4,2,4,2,10,2,10
130 FOR I=1 TO 53:READ E[I]:NEXT I
140 PRINT "Modulo 210 Factors Progra
m":PAUSE 2
200 INPUT "N = ";N:N0=N
210 PRINT "Busy Factoring"
220 FOR I=1 TO 5
230 D=E[I]
240 IF INT[N/D]*D=N THEN GOSUB 400
250 NEXT I
300 IF N/D<=SQR[N]THEN X[K]=N:Y[J]=1
:GOTO 500
310 FOR I=6 TO 53
320 D=D+E[I]
330 IF INT[N/D]*D=N THEN GOSUB 400
340 NEXT I
350 GOTO 300
400 X[K]=D:S=S+1:N=N/D
410 IF N=1 THEN X[K]=D:Y[J]=S:GOTO 5
00
420 IF INT[N/D]*D=N THEN 400
430 Y[J]=S:K=K+1:J=J+1:S=0:RETURN
500 DISPLAY BEEP:FOR I=0 TO 12
510 IF X[I]=0 THEN 600
520 PRINT "F["&STR$[I+1]&"] ="&STR$[
X[I]]&"^"&STR$[Y[I]]
530 PAUSE:NEXT I
600 PRINT "N was ";
610 PRINT USING"###############",N0:P
AUSE
620 FOR I=0 TO 12:X[I]=0:Y[I]=0:NEXT
 I : S = 0
630 J=0:K=0:GOTO 200
```

PRIME FACTOR PRINTOUTS WITH THE CC-40 AND HX-1000 PRINTER

The CC-40/HX-1000 combination provides a printout capability on demand. Use of the Call SETLANG command before entering the prime factors program even provides the annotation for the printout in the chosen language. See the examples at the right.

FACTEURS PREMIERS	PRIMZAHLEN
Nb a Decomposer= 987654321	Zahl= 987654321
F1=3	F1=3
F2=3	F2=3
F3=17	F3=17
F4=17	F4=17
F5=379721	F5=379721

MORE ON ACCURACY OF THE LN FUNCTION - Laurance Leeds and Palmer Hanson. The table on page 15 compares the natural logarithm function for several computers for selected Ln(1+x) problems where x is small. The table entries show that the Model 100 and the TI-66 are clearly superior.

V9N4P9 discussed alternate methods of evaluating Ln(1+x) where x is near zero. One method which is described in the HP-15C Advanced Functions Handbook provided improved results for the Bob Fruit benchmark test with the HP-11, the TI-66 and the Model 100. The results with the CC-40 were only slightly improved, and the results with the TI-59 were degraded.

Laurance Leeds provided a 52 step TI-59 routine to improve the calculation of Ln(1+x) where x is near zero (see the left hand listing below). With an input of x, in two seconds the program will provide Ln(1+x) to the accuracy shown in the next to the bottom row on page 15. A 25 step program can provide identical results, but with requires four seconds execution time (see the center listing below). The advantage of this program is that the number of iterations, and hence the accuracy, can be increased with very little penalty in program steps. If the 5 at location 005 is changed to a 7, then the results in the last five columns of the next to last row on page 15 are unchanged, but the results for larger x are much improved:

Ln(1.1) = 0.0953101809524 which is correct to 8 figures, and
Ln(1.01) = 0.00995033085317 which is correct to 12 figures.

An BASIC program which is the equivalent of the shorter TI-59 program, but which accepts (1+x) rather than x, is shown at the right below. Twelve iterations are used for the CC-40 since the response time is still nearly instantaneous. The results are in the bottom row on page 15. The Ln(1.1) is correct to 13 significant figures.

```
000  76 LBL      026  03   3
001  11  A       027  65   x
002  42 STO      028  02   2
003  00  00      029  65   x
004  55  ÷       030  43 RCL
005  05  5       031  00  00
006  65  x       032  95   =
007  04  4       033  94 +/-
008  95  =       034  85   +
009  94 +/-      035  01   1
010  85  +       036  95   =
011  01  1       037  55   ÷
012  95  =       038  02   2
013  55  ÷       039  65   x
014  04  4       040  43 RCL
015  65  x       041  00  00
016  03  3       042  95   =
017  65  x       043  94 +/-
018  43 RCL      044  85   +
019  00  00      045  01   1
020  95  =       046  95   =
021  94 +/-      047  65   x
022  85  +       048  43 RCL
023  01  1       049  00  00
024  95  =       050  95   =
025  55  ÷       051  92 RTN
```

```
000  76 LBL
001  11  A
002  94 +/-
003  42 STO
004  00  00
005  05  5
006  42 STO
007  01  01
008  25 CLR
009  76 LBL
010  12  B
011  85  +
012  43 RCL
013  01  01
014  35 1/X
015  95  =
016  65  x
017  43 RCL
018  00  00
019  95  =
020  97 DSZ
021  01  01
022  12  B
023  94 +/-
024  92 RTN
```

```
1000 IMAGE .######
########^^^^

1000 IMAGE .######
########^^^^
1010 INPUT Z
1020 Z=1-Z
1030 F=0
1040 FOR I=12 TO 1
 STEP -1
1050 F=Z*(F-1/I)
1060 NEXT I
1070 PRINT USING 1
000;F:PAUSE
1080 GOTO 1010
```

Suppose that we use these routines for Ln(1+x) to improve the accuracy of the solutions for Bob Fruit's benchmark test.

Exact Solution	2260.48792 47960 86067 ...
TI-59 (5 iterations)	2260.48972 4844
TI-59 (7 iterations)	2260.48792 4793
CC-40 (12 iterations)	2260.48972 4796

where the CC-40 solution is correct to 13 significant figures.

More on Accuracy of the Ln Function - (cont)

	1e2*Ln(1.1)	1e3*Ln(1.01)	1e4*Ln(1.001)	1e5*Ln(1.0001)	1e6*Ln(1.00001)	1e7*Ln(1.000001)	1e8*Ln(1.0000001)
Exact	9.53101 79804 3248	9.95033 08531 6808	9.99500 33308 3533	9.99950 00333 3083	9.99995 00003 3333	9.99999 50000 0333	9.99999 95000 0003
M-100	9.53101 79804 323	9.95033 08531 677	9.99500 33308 351	9.99950 00333 305	9.99995 00003 332	9.99999 50000 030	9.99999 94999 999
TI-66	9.53101 79804 32	9.95033 08531 68	9.99500 33308 35	9.99950 00333 30	9.99995 00003 33	9.99999 50000 04	9.99999 95000 05
EL-512	9.53101 79804	9.95033 08532	9.99500 33308	9.99950 00333	9.99995 00003	9.99999 5	9.99999 95
HP-11	9.53101 7980	9.95033 0853	9.99500 3331	9.99950 0033	9.99995	9.99999 5	9.99999 95
TI-59	9.53101 79804 3	9.95033 08532	9.99500 3331	9.99950 003	9.99995	9.99999 5	10.
CC-40	9.53101 79804	9.95033 0853	9.99500 333	9.9995	9.99994 9	10.	10.0001
TI-57	9.53101 798	9.95033 08	9.99500 2	9.9995	10.	10.	10.
TI-55II	9.53101 798	9.95033 09	9.99500 3	9.9995	9.9999	10.	9.99
Comm 64	9.53101 797	9.95033 072	9.99500 669	9.99950 29	9.99979 339	10.00431 22	9.98978 747
Color C	9.53101 807	9.95033 104	9.99500 024	9.99950 29	9.99979 34	9.99947 063	9.98978 747

Alternate Programs for Ln(a) where a is near 1:

	1e2*Ln(1.1)	1e3*Ln(1.01)	1e4*Ln(1.001)	1e5*Ln(1.0001)	1e6*Ln(1.00001)	1e7*Ln(1.000001)	1e8*Ln(1.0000001)
TI-59	9.53103 33333 4	9.95033 08533 4	9.99500 33308 4	9.99950 00333 4	9.99995 00003 4	9.99999 50000 1	9.99999 95000 1
CC-40	9.53101 79804 32	9.95033 08531 681	9.99500 33308 35	9.99950 00333 309	9.99995 00003 33	9.99999 50000 034	9.99999 95

The table lists the output from the natural logarithm function for a range of hand-held programmable calculators and personal computers. In the table the Radio Shack Model 100 is clearly superior, closely followed by the TI-66. It seems to be a well-kept secret that the TI-66 mechanization seems to have fixed all of the deficiencies of the TI-59 such as non-commutative multiply, $\sin\theta \neq \cos(90-\theta)$, etc. The Sharp EL-512 also does very well. The last two rows show the results obtained using the programs on page 14 to improve the solution for the natural logarithm when the argument is near unity.