TI PPC NOTES

NEWSLETTER OF THE TI PERSONAL PROGRAMMABLE CALCULATOR CLUB

P.O. Box 1421, Largo, FL 34649

Volume 12, Number 2 — Second Quarter 1987

The feature articles in this issue are the discussions of revealed firmware and memory dumps for the TI-95 by Robert Prins and Ruud Paap. You will note that they have retained the copyright for the material, but have given our club permission to publish it. In the letter which came with the articles Robert wrote "IMPORTANT NOTE: All test functions completely destroy user memory, so before you start using them you MUST safeguard the entire contents of your machine."

On the back page of the last issue I announced the availability of a magnetic tape service for the TI-95. Unfortunately, I garbled the discussion of price, saying ten dollars in one place and five dollars in another. Those who responded sent ten dollars, so I refunded five dollars to them. The offer stands, ten programs on a magnetic tape for five dollars.

With the help of Don and Stan Seba of Atchison, Kansas I have also been able to establish the portability of tapes for the TI-74. Accordingly, I can offer a TI-74 tape with six programs for five dollars.

It took about three-and-one-half months to get out this issue. The next one will certainly not be out until next year, and the last one for this "year" of four issues will probably not be out until April.

TABLE OF CONTENTS

ERRATA 2
MAILBAG 3
THE TI-59 IS NOT AN ORPHAN 4
MORE ON TI-59 DIAGNOSTICS 5
EXPRESSIONS AS RESPONSE TO INPUT WITH TI-74 5
CALENDAR PROGRAM FOR THE TI-74, P. Hanson 6
THE TI-65, R. Stucker 8
THE CASIO fx-8000G, S. Becker 9
MORE USEFUL TI-74 FUNCTIONS 9
ERROR IN PROGRAMMABLE CALCULATOR NEWS 9
HIGH ACCURACY CUBIC IN BASIC, L. Leeds 10
NOTES ON PI-FINDING 11
COMMENTS ON THE SIGNUM FUNCTION, C. Williamson ... 11
EXTENDED PRECISION MULTIPLICATION, L.Leeds, R.Prins. 12
AITKEN'S Δ^2 CONVERGENCE METHOD, J. Valencia 13
MORE PPX PROGRAM AVAILABILITY 14
EVALUATION OF POLY CURVE FITTING, G.Thomson,P.Hanson 16
ITERATIVE REGRESSION WITH THE TI-74 MATH MODULE 20
ITERATIVE REGRESSION WITH ROW REDUCTION ON TI-74 ... 21
KAPREKAR'S CONSTANT, W. Widmer 22
BOOK REVIEW, P. Hanson, W. Bodenmuller 23
EXTENDED PRECISION SQUARE ROOTS ON TI-95, H. Ladd .. 24
HARDWARE AND SUPPLIES 7, 8, 9, 13, 15, 23

Magnetic card service will continue to be available for TI-59 programs in this issue, and for programs in the 1983 through 1986 issues.

One dollar per card plus a stamped and self-addressed envelope, please.

HAPPY HOLIDAYS!

Palmer

ERRATA

Extended Precision Division (V11N4P25) - Larry Leeds and Gene Friel have reported that step 287 of Peter Messer's extended precision division program should be changed from EQ (code 67) to GE (code 77). Both Larry and Gene provided sample problems to illustrate the difficulty. The results are shown below with either EQ or GE at step 287:

Larry Leeds

EQ	GE
444444.	444444.
444444.	444444.
444444.	444444.
888888.	888888.
888888.	888888.
888888.	888888.
888888.	888888.
888888.	888888.
888888.	888888.
1999999.	2000000.
999999.	0.
1000002.	2.
-1.	-1.

Gene Friel

EQ	GE
265371.	265371.
653000.	653000.
100000.	100000.
376829.	376830.
1000000.	0.
37682.	37683.
1000000.	0.
3768.	3768.
300000.	300000.
376.	376.
829999.	830000.
1000037.	37.
682999.	683000.
1000003.	3.
768299.	768300.
1000000.	0.
376829.	376830.
1000000.	0.
37682.	37683.

The exact answer to Larry's example is 2,000,000,000,000,000,002. Gene's problem yields a repeating decimal.

Editor's Note: Peter's submission contained a GE at step 287. I improperly transcribed the program, and my test problems failed to expose the deficiency. Examination of both examples will show that the printout with EQ at step 287 is correct if the seventh digit is interpreted as a carry up to the next line; however, the printout with GE at step 287 is certainly easier to read.

Curve Fitting with Orthogonal Polynomials (V12N1P24) - George Thomson found that the following changes are needed in the TI-74 version of William Hood's program. First, add a new line at 1778. Then, change the <'s to >'s in line 1070, and change line 1910 to set MF = 1. The revised lines will be:

```
1070 IF MF>0 AND M>MM THEN J1=MM+1:MM=M:GOTO 1130
1778 IF PN=0 THEN PAUSE 1
1910 IF A$="Y" OR A$="y" THEN MF=1:GOTO 1680
```

The change at line 1778 flashes "Coeff of Det (R^2) = " before stopping with the value in the display when a printer is not used. The changes at lines 1070 and 1910 provide a faster solution for a polynomial of a higher degree if a solution for a polynomial of lower degree has already been completed. Without those changes the fitting a third degree polynomial to the ten point problem from BYTE requires about 13 seconds and the fitting a fourth degree polynomial to the problem requires an additional 17 seconds. With the changes the third degree solution would still require about 13 seconds if it were the first solution, but the fourth degree solution can be obtained in an additional eight seconds.

MAILBAG

"I greatly appreciate your coverage of Casio's fx-7000G, as in the Ladder Problem. ... ...It's non-TI, but it's a gem and a blessing with its plotting screen that lets you see what functions and phenomena look like, and finds roots to boot." J.V.

"... I bought a TI-66 a while back on the basis of your info--a marvel, but no program storage. I look forward to hearing about the TI-74 and TI-95." R.B.

"Sorry about the TI-59. I have hundreds of programs which I use almost every day. I assume the TI-95 programming is not compatible with the TI-59. Some discussion in the Notes would be helpful. I look forward to each new issue." A.M.

> There is compatibility between the TI-59 and TI-95 programming structure, sort of. In each issue we will have at least one illustration of some of the pit-falls to watch for. In this issue it is Hewlett Ladd's conversion of Robert Prins' square root program.

"Although I don't understand much of the information in the newsletter I do enjoy the challenge and have learned some new things from it." L.K.

"I'd like to see some challenges, puzzles, and other little things to do to challenge the thinking of your readers. The last issue was terrific." M.B.

"Keep up the good work! I have a TI-74, TI-95 and fx-7000G so information on them is of most interest." G.G.

"Regarding the TI-95 I would like to see (1) TI produce a cartridge that only does symbolic math -- in direct competition with the HP-28C, (2) more consumer advertisement, and (3) an elegant program for solving quartic equations using the built-in cubic case." P.M.

"Please try to include more on the TI-59 for this year." W.G.

"Glad to see more BASIC programs." R.B.

"How about that wonderful TI bunch in the Netherlands. I had to get my PC-100 repaired there. I had to wait a whole 30 minutes for it back in 1979." M.K.

"Would like to see more utility programs, M.E., HVAC, and photography." C.C.

"With more and more calculators coming into the marketplace some guidance is needed as to their quirks and individual characteristics. Your newsletter is certainly helping to fill that need for all the membership." P.W.

"Would like to see information on how to use the TI-59 programs on an IBM." F.S.

"...I have the TI-95 and Casio fx-8000G so you can see that I need all the help I can get." B.R.

"The day of the TI-95 is here ...". A.L.

"There is a very interesting laptop released in the UK by Sir Clive Sinclair called a '288'. Z80 based, it weighs under 2 pounds with 32K expandable to 4M. Three EPROM ports up to 1M each. Eight line, 80 column display (supertwist). Integrated in ROM software are a word processor, filer, spreadsheet, diary, calendar and calculator. ... If it gets to the states it's supposed to retail for $300 by Cambridge Computer Ltd, Freepost, Cambridge, CB4 1BR." C.F.

THE TI-59 IS NOT AN ORPHAN - P. Hanson. Based on information from readers and a confirming call to 1-800-TI-CARES I reported that TI was no longer repairing the TI-59. A letter from Tom Shields, Manager Consumer Relations at TI, indicates that my interpretation was not entirely correct. TI has discontinued the exchange policy, whereby a reconditioned unit could be obtained by turning in a broken TI-59 and about $65.00. TI will continue to repair TI-59's under certain conditions. Mr. Shields' letter states:

"When a customer requests that his TI-59 be repaired, rather than replaced, we attempt to repair it. However, due to the unavailability of some internal replacement parts, we cannot repair all TI-59's. ... ... If you want to send your unit for service, it should be sent prepaid and insured to the following address:

Texas Instruments Incorporated
2305 North University Avenue
Lubbock TX 79408

Please enclose only those items in need of service, along with a note describing the particular problem you have observed. Do not send carrying cases, manuals or original display boxes, as these items may not be returned."

The current out-of-warranty repair charge is $90.00 plus local sales tax plus $6.00 for shipping and handling. To be sure you are using up-to-date information, call 1-(806)-747-1882 or 1-800-TI-CARES for the latest repair information.

The notice in V11N4P25 also stated that the replacement fees for obtaining upgraded calculators and printers were

TI-74 for a TI-59 + $99.00 + $6.00 shipping + local sales tax.
TI=95 for a TI-59 + $144.00 + $6.00 shipping + local sales tax.
PC-324 for a PC-100 + $64.00 + $6.00 shipping + local sales tax.

and reported that, in the case of the TI-59 replacements, a user can get a better price from Elek-Tek. That would be true if the exchange was simply one of a broken TI-59 for a new TI-74 or TI-95 plus the replacement fee , shipping fee and sales tax; however, the TI exchange price includes a FREE software cartridge of your choice. The following table compares the exchange price list (without sales tax) with that of some leading discount houses:

Product		TI	Elek-Tek	EduCalc	Service
TI-74	(BASICALC)	$105	$ 94	$ 99.95	$ 99.97
TI-95	(PROCALC)	150	139	145.95	157.93
8K RAM	(Programmable RAM)	32	34	39.95	33.74
PC-324	(Thermal Printer)	70	69	89.95	
AC9201	(AC Adapter)	15	14	16.95	
CI-7	(Cassette Interface)	25	22	26.95	
TI-74 ROM's	(Software Cartridges)	32	29	35.95	
TI-95 ROM's	(Software Cartridges)	32	33	39.95	

where the shipping fee has been included in the TI and Service Merchandise prices. For Elek-Tek add $4.00 shipping/handling for the first item, and $1.00 for each additional item. For EduCalc add $1.00 shipping/handling per order. Thus, an exchange and upgrade with TI provides a TI-74 and one ROM Software Cartridge for $105 plus sales tax; the equivalent would be $128 from Elek-Tek, or $140.90 from EduCalc. Similarly, the exchange with TI provides a TI-95 and one ROM Software Cartridge for $150 plus sales tax; the equivalent would be $177 from Elek-Tek, or $186.90 from EduCalc. So, if you intend to acquire either a TI-74 or TI-95 and a software module, and if you have a defective TI-59 to turn in, you will save money over most discount house equivalents.

MORE ON TI-59 DIAGNOSTICS - P. Hanson. I received a number of favorable comments on the discussion of diagnostics for the TI-59 which appeared in V12N1P4-7. I should note that the EE problem described on V12N1P1 cannot be found with any of the diagnostics discussed. It was found while I was exploring programs which could be used to test additional TI-59 functions. I was trying the print code printer program from V10N3P18-19 which has some hierarchy (HIR) manipulations and a lot of output to the printer. Even though that program happened to have the sequence which identified a problem in a particular TI-59 I am not happy with it as an addition to our diagnostic library. I would prefer a program that thoroughly exercises the HIR commands. The calendar printing program in V9N2P7/8 may be a better test. Any other suggestions? Will someone write an HIR diagnostic?

If you have never tried to use a defective programmable you may wonder what all the fuss is about. Consider the example which appears in Jerry Pournelle's column "Computing at Chaos Manner" in the August 1987 issue of BYTE. On pages 250-251 under the heading "Check Your Math Chip" he describes a series of problems encountered when trying to run some benchmark programs with a new compiler on an "old reliable" computer which he calls Lucy. After finding the problem he wrote:

> "... By then I knew what the situation was. The 8087 was working just well enough to let the programs believe there was one in the system; but not well enough to let the Borland program run properly. The Microsoft code generator was more tolerant and let the program run--but give the wrong answer.
>
> Once I understood that, I removed the math chip. Everything worked fine.
>
> The moral of the story is that if you have a system with a math chip, you want to check it against a known problem like my benchmark every now and then. That 8087 had been in Lucy for at least five years. We've never had any indication that anything was wrong until I tried my benchmark program. Fortunately, Lucy Van Pelt is mostly used as a data-entry device for Q&A files; she's too slow for spreadsheets. I don't think we've got any significant wrong answers--but we easily could have."

I agree with Pournelle's comment completely. If you are doing anything with your programmable calculator or computer which is even moderately important, then you should run some benchmarks or diagnostics now and then. What we need are some programs for the TI-74 and TI-95.

EXPRESSIONS AS THE RESPONSE THE INPUT STATEMENT WITH THE CC-40 AND TI-74 - P. Hanson

During evaluation of solutions for cubic equations we used a test problem proposed by Peter Messer: $x - 2x + (4/3)x - 2/9 = 0$. The fractional coefficients did not cause any difficulty when we were using calculator programs, but when Larry Leeds developed a BASIC program on his Model 100 he found that it was not so easy to enter the fractional coefficients, and he used program statements to enter the coefficients. When I converted his Model 100 program for use on the CC-40 and TI-74 I received a pleasant surprise. The Input statement with those machines permits a response with a numeric expression. An example will take the place of many words. Consider the program at the right. Press RUN and see a question mark in the display. Place 2/9 in the display and press ENTER. The value .2222222222 is printed. A question mark appears in the display indicating that a value for B will be accepted. Place SIN(PI/6) in the display and press ENTER. The value .5 is printed. A question mark in the display indicates that a value for C will be accepted. Place 2 + SQR(5) in the display and press ENTER. The value 4.236067978 is printed. A question mark indicates that a value for D will be accepted. Place A/B + C in the display and press ENTER. The value 4.680512422 is printed.

```
10 OPEN #1,"12",OUTPUT
20 RAD:INPUT A
30 PRINT #1,A
40 INPUT B
50 PRINT #1,B
60 INPUT C
70 PRINT #1,C
80 INPUT D
90 PRINT #1,D
99 END

 .2222222222
 .5
 4.236067978
 4.680512422
```

CALENDAR PROGRAM FOR THE TI-74 - P. Hanson.

Long time members know that I am fascinated by calendar printing programs, and so will not be surprised that I have written one for the TI-74. In this case the program also serves to illustrate the differences in the printing capability of the TI-74/PC-324 and the CC-40/HX-1000.

A calendar program for the CC-40/HX-1000 appeared in V9N5P8. Allowing two characters for each date and one space between dates yields 20 characters as the minimum number needed per line. That was the format used on the TI-59/PC-100 programs where the PC-100 provided exactly 20 characters per line. The HX-1000 provided two print modes, either 18 characters per line or 36 characters per line. One solution would have been to use the 18 character per line mode and the European format as in Dave Leising's calendar program for the TI-66/PC-200. I chose another alternative: the program in V9N5P8 used the 18 character mode for the month and year, and switched to the 36 character mode for the days of the week and the dates. Three-letter abbreviations were used for the days of the week. The actual program was a minimum change version of an earlier program I had written to display a calendar on the screen of my Model 100. A sample printout from the CC-40/HX-1000 program appears below. The use of two character sizes provides an attractive format. Printout of a single month required about 21 seconds, agonizingly slow by TI-59 standards where Patrick Acosta's program in V9N2P7 prints out a a full year in only 83 seconds when running in fast mode.

The program at the right is a modification of the CC-40 program for use with the TI-74/PC-324. The necessary modification accommodates the 24 character print width of the PC-324 by returning to a format similar to that used with the TI-59/PC-100, that is each column separated from the next by only one space, and two-letter abbreviations for the days of the week. I also cleaned up the program by removing the subscript notation for the individual lines of the calendar. I can't recall why I used subscripts in the first place. A representative printout appears below. A month can be printed in nine seconds, about the same speed as the TI-59/PC-100 in fast mode, and much faster than the CC-40/HX-1000.

```
100 DIM Q(12)
105 DATA 31,28,31,30,31,
30,31,31,30,31,30,31
110 FOR I=1 TO 12:READ Q
(I):NEXT I
115 DATA "JANUARY  ","FE
BRUARY ","MARCH    ","AP
RIL    ","MAY      ","JU
NE     "
120 DATA "JULY     ","AU
GUST   ","SEPTEMBER","OC
TOBER  ","NOVEMBER ","DE
CEMBER "
130 INPUT "Enter Month (
1-12): ";M
135 IF M<1 OR M>12 THEN
130
140 INPUT "Enter Year (>
1582): ";R
145 IF R<1583 THEN 140
150 IF R-4*INT(R/4)=0 TH
EN Q(2)=29
155 IF R-100*INT(R/100)=
0 THEN Q(2)=28
160 IF R-400*INT(R/400)=
0 THEN Q(2)=29
165 R1=R-1:R2=R+INT(R1/4
)-INT(R1/100)+INT(R1/400
)
170 FOR I=0 TO M-1:R2=R2
+Q(I):NEXT I
175 D1=R2-7*INT(R2/7)
180 RESTORE 115:FOR I=1
TO M:READ M$:NEXT I
185 OPEN #1,"12",OUTPUT
190 PRINT #1,TAB(3);M$;"
     ";R
210 PRINT #1,"  Su Mo Tu
 We Th Fr Sa"
225 C$=" "&RPT$("   ",D1
)
230 FOR I=1 TO 7-D1:C$=C
$&"  "&STR$(I):NEXT I
235 PRINT #1,C$
245 I=8-D1
250 C$=" "
255 FOR L=1 TO 7
260 IF I>9 THEN B$=" " E
LSE B$="  "
265 C$=C$&B$&STR$(I)
270 I=I+1:IF I>Q(M)THEN
PRINT #1,C$:GOTO 300
275 NEXT L
280 PRINT #1,C$
285 GOTO 250
300 PRINT #1
305 CLOSE #1:GOTO 130
```

CC-40/HX-1000

```
 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
```

TI-74/PC-324

```
 FEBRUARY      2000
 Su Mo Tu We Th Fr Sa
        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 29
```

REPAIR OF A MAGNETIC CARD READER - L. Leeds. My TI-59 was gradually refusing to read magnetic cards. Finally the day came when it gave up entirely. I tried the Drive Roller card, the Head Cleaning card, Hanson's Universal Solvent, the CCL144 Cleaning Card right out of the envelope, a second run through with the CCL144 Cleaning Card, and the CCL144 Cleaning Card dampened with isopropyl alcohol--all with no improvement. Then, with power off, and the CCL144 Cleaning Card dry, I inserted the card in the exit slot and carefully worked it in until it was more than half way in. Then I slowly worked it back and forth a distance of about 3/8 inch, doing this eight or ten times. The machine now reads and writes cards every time, just as though it was brand new!!! It is possible that running the dry card through many times under power drive would have accomplished the same result. At the same time I disassembled the machine and put in a new plastic pad under the keys. No trouble with bounce or double entry. Also it passed all of your tests. Additionally, it gives correct answers. Could I ask for more?

Editor's Note: The CCL144 Cleaning Cards continue to be available for two dollars.

A PROGRAMMING CHALLENGE - While faster algorithms have been know to be available, (e.g., see The Art of Computer Programming by D. Knuth, volume 2, pp 364-398), the factor finders in previous issues of TI PPC Notes have all been versions of the sieve of Eratosthenes. The notice at the right, which is from Page 59 of issue #35 of the EduCALC catalog, offers discussions and examples for one of the faster algorithms. Write to Algorithms Dept., EduCALC Mail Store, 27953 Cabot Road, Laguna Niguel, CA 92677 for a copy. Remember to include the self-addressed envelope with 39 cents in stamps.

Of course, the challenge is to show that more than five or six digits can be obtained from TI Programmables.

Factoring Large Numbers on the HP-16C

Factoring large numbers intrigues both amateur and serious number theorists and factoring now gets increasing attention with recent applications to cryptography. You may be interested in the factoring algorithms that are offered in a recent article by Blair/Lacampagne/Selfridge.

In addition to a short discussion and a step-by-step description of each algorithm, they include programs that factor numbers up to 19 digits, fast, using the HP-16C! Interestingly, these algorithms would only work up to about 5 or 6 digit numbers on other calculators.

As a public service, EduCALC will send you a copy of this article if you send us a self-addressed envelope with 39 cents in postage on it—please send it to our "Algorithms Dept".

TI-59 MODULE CHANGER WANTED - If you have one for sale, write to:

Q. Jose M. Gallego, 2302 'D" Ave., Apt 104, P.O.B. 2746, National City, CA 92050.

CC-40 SOLID STATE CARTRIDGES FOR SALE - The cartridges which are available include a mathematics cartridge, a memo processor/data communications cartridge, a 16K RAM cartridge, a battery backed-up RAM cartridge, and an editor/assembler cartridge. Write to David R. Hertling, 4546 Cherie Glen Trail, Stone Mountain, GA 30083.

USED TI-59 FOR SALE - The Master Library module is installed. Neither the battery charger nor the manuals are available. The battery is weak. The calculator has passed all of the tests described in pages 4 through 7 of V12N1. A friend offers this "bare-bones" device on an "as-is" basis for fifty dollars. You pay the shipping. If you want this device send sixty dollars to me at the news-letter address. I will give my friend fifty dollars, ship the calculator to you, and return any part of the ten dollars shippiing fee which is not used. P.O.H.

THE TI-65 - Bob Stucker writes: The TI-65 Technical Analyst is available at the BEST discount store for about sixty dollars. The size and layout are similar to the TI-66, but in a nice hard case similar to the TI-95. Functionally, the TI-65 is similar to a TI-60. It can be partitioned from 100 program steps and one memory to no program steps and sixteen memories. It has some merged commands and F1 and F2 keys that function like the user defined keys on a TI-66. It has Dsz, tests against a register, and subroutines. The speed is like a TI-66 but accuracy is like a TI-60. Entering an integer and taking the square root five times followed by squaring five times and subtracting the input integer yields -8 E-11 for an input of 2, and -7.9 E-11 for an input of 3. All the TI-60 functions are there except Δ%, nCr, and nPr. The ∫dx function is available. It has constants such as gravity, the speed of light, and Plank's constant. The display has ten digits at all times and has many annunciations such as 2nd, 3rd, INV, Deg, GRAD, etc. It programs in the Insert mode like the TI-66, but when CP is pressed in Lrn mode all steps from the next one on are erased.

A new feature is a timer which counts up to 40 hours or down from an entered time, but there is no beeper. The timer function can be used in a program such as

```
Lbl  F1  TUp  CE  R/S                   Starts timer.
Lbl  F2  TUp  CE  D.d  1/x  R/S         Stops timer.
```

Press F1 at a mile marker on the highway and hold a steady speed. Press F2 at the next mile marker and the displayed value is your speed. With Dsz, the tests against a register, and 100 steps we may finally have a nice replacement which will run many old TI-57 programs.

Editor's Note: Issue #36 of the EduCalc catalog notes that the TI-65 has 16 English-metric conversions, 8 physical constants, 148 math functions, 2-variable statistics, 100 program steps, and can calculate in octal, decimal or hexidecimal bases. EduCalc offers the TI-65 for $49.95. The "EE's Tools and Toys" column in the August 1987 issue of IEEE Spectrum notes that the resolution of the stopwatch-timer is one second and reports the list price as $75.00.

USED HARDWARE FOR SALE

TI-59/PC-100C with four rolls of paper. Six modules--Master Library, Statistics, Electrical Engineering, Real Estate/Investment, Leisure Library, and Math/Utilities with manuals. The 59 Fun and Printer Utility Pakettes. Several dozen blank magnetic cards and several dozen more with programs and documentation. A Service Manual, AC charger, and a TI 12V auto adapter. Will sell the package for $200.

TI-66/BA-55/PC-200 with ten rolls of paper. $70 for the package.

Casio fx-7000G -- $35.

Write to Robert Stucker, 7606 E. 90th Terrace, Kansas City MO 64138 or call (816)-765-4032.

THE CASIO fx-8000G - Stanley Becker reports that the random number generator in the fx-8000G does NOT have the problem reported for the fx-7000G. He also reports that the instructions for the FA-80 interface accessory for the fx-8000G say to connect the printer with the Casio SB-51 cable, but several people at Casio have told him over the telephone that Casio will not make the cable, and that a male-to-male Centronics cable should be used instead.

Editor's Note: I found an interesting observation on the generation of random numbers on page 1 of Volume 2 of Knuth's The Art of Computer Programming:

> "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin. John Von Neumann (1951)"

MORE USEFUL FUNCTIONS ON THE CC-40 AND TI-74 - P. Hanson. Page 5 describes my discovery that the CC-40 and TI-74 permit algebraic expressions as the response to an input statement. I decided to look through the manuals for additional capabilities that I may have missed. I found the PAUSE ALL statement and the subprogram capabilities, both of which provide easier solutions for certain programming requirements.

The PAUSE ALL statement suspends program execution each time a complete output line has been sent to the display, and execution continues when the CLR or ENTER key is pressed. When used with PRINT #PN statements where PN is set to zero if the printer is not connected, or set to one if the printer is connected, the PAUSE ALL statement provides an easy way to stop execution with a result in the display if a printer is not used, but to continue without stoppping if a printer is used. The alternative which I used in earlier programs was a statement such as IF PN = 0 THEN PAUSE each time the option was desired.

The subprogram capability is useful when the programmer would like to use the same mathematical routine several places in the same program, but with different definitions for the variables. The argument list in the CALL statement passes a variable list from the main program to the subprogram. The SUB statement which marks the beginning of a subprogram can redefine the variable list as desired. The subprogram capability is particularly useful in providing portability of a routine from one program to another.

Both the PAUSE ALL and the subprogram capability are demonstrated in the cubic program by Larry Leeds on page 10. The subprogram capability is also demonstrated in an iterative least squares program elsewhere in this issue, and in the DMS to decimal degrees program on page 7 of the first issue of Programmable Calculator News.

ERROR IN PROGRAMMABLE CALCULATOR NEWS - A copy of the Volume 1 Number 1 issue was included with V12N1. A TI representative writes that there were errors in the listing of Patrick Hicks' program "Can I Really Afford It" on page 8 of the newsletter. Lines 190 and 230 should have read as follows:

```
190  PV=PMT*(1-(1+I)^(-N))/I

230  PMT=PV*I/(1-(1+I)^(-N))
```

CCL-144 CLEANING STRIPS - I ordered another box of ten cleaning strips in mid-September. The invoice accompanying the shipment stated "These cleaning strips are discontinued and will no longer be available". I will continue to fill orders as long as these strips last. $2.00 each, no checks.

CUBIC SOLUTION - Larry Leeds. V11N4P16/17 examined the results for a variety of cubic solutions, and invited members to submit more accurate solutions. The algorithm for the program presented here is as follows: Zero is used as the first approximation. An Exact Newton (the program does the differentiation) finds one of the real roots, or the only one. This value is truncated to a six digit number which is used as the approximation. Newton then produces an exceptionally close answer and prints the root. Synthetic division derives the residual quadratic. A test is made to determine if the roots are complex or real, and the quadratic is solved. If complex, the roots are printed. If real, the two roots are truncated to 6 digit numbers which are considered as approximations. Each root, in turn, is then presented to the original cubic and solved by the exact Newton.

```
Ax^3 + Bx^2 + Cx + D = 0
A =   1
B =  -2
C =   1.333333333
D =  -.2222222222

Real Root =
 2.466929833686E-001
Complex Roots =
Real Part
 8.766535083155E-001
Imaginary Part
+/- 3.637078786574E-001

Ax^3 + Bx^2 + Cx + D = 0
A =   1
B =  -6.975
C =   16.216874
D =  -12.5680758

Real Root =
 2.324000000000E+000
Real Root =
 2.326000000000E+000
Real Root =
 2.325000000000E+000
```

The improvement obtained from truncation is presumed to be associated with the Newton iteration process and the overflow truncation which occurs in the computer. Each iteration produces a 14 digit number which will be both squared and cubed, hence the value of the function is not exact. As the iteration approaches the correct value of X you may obtain a value which is in error in both the 13th and 14th digit positions. Because of overflow truncation, the program would conclude that the erroneous value of X is the exact root. I found by experiment that if this value of X is then truncated and is put through the Newton solution again, the result will be very close to the exact value. The examples at the right show the effect of the truncation. The solutions at the top are with the program as on page 11. The solutions at the bottom are with the truncations removed. Note that the program finds the exact roots for the second example when the truncation is in place.

```
Ax^3 + Bx^2 + Cx + D = 0
A =   1
B =  -2
C =   1.333333333
D =  -.2222222222

Real Root =
 2.466929833685E-001
Complex Roots =
Real Part
 8.766535083160E-001
Imaginary Part
+/- 3.637078786559E-001

Ax^3 + Bx^2 + Cx + D = 0
A =   1
B =  -6.975
C =   16.216874
D =  -12.5680758

Real Root =
 2.324000817341E+000
Real Root =
 2.325999823508E+000
Real Root =
 2.324998366829E+000
```

I would appreciate hearing from the club members of any difficulties encountered with this algorithm.

Editor's Note: Larry writes his BASIC programs in Microsoft BASIC for the Radio Shack Model 100. I convert them for use with the CC-40 and TI-74. The truncation function is very easy to implement on the Model 100 by using the single precision command, e.g., A = CSNG(A) truncates the precision from 14 digits to 7 digits. The subprogram in lines 1001-1004 solves the problem nicely for the TI-74 or CC-40. The calls at lines 240, 250 and 800 pass the appropriate value to and from the subprogram, and the subprogram uses its own variables S and K. Note that it is permissible to use K for one purpose inside the subprogram even though it is defined differently outside the subprogram and in the call to the subprogram. One alternative was to repeat lines 1002 and 1003 three times with appropriately chosen variables. Another was to call a subroutine -- that would require a change of variables prior to entry to the subroutine and after exit from the subroutine.

Line 100 of this program illustrates the use of the PAUSE ALL command to stop the computer for display if a printer is not used.

Cubic Solution - (cont)

```
10 REM Cubic Solution by
 L. Leeds
12 INPUT "Use Printer (Y
/N) ? ";A$
14 IF A$="Y"OR A$="y"THE
N PN=1 ELSE 20
16 INPUT "Device Code ?
";P$
18 OPEN #1,P$,OUTPUT
20 PRINT #PN:PRINT #PN,"
Ax^3 + Bx^2 + Cx + D = 0
"
25 IF PN=0 THEN PAUSE
30 IMAGE "##.###########
#^^^^^";N
40 R$="Real Root = "
60 A$="A =  ":INPUT A$:A
65 PRINT #PN,A$;A
70 B$="B =  ":INPUT B$:B
75 PRINT #PN,B$;B
80 C$="C =  ":INPUT C$:C
85 PRINT #PN,C$;C
90 D$="D =  ":INPUT D$:D
95 PRINT #PN,D$;D
100 PRINT #PN:PAUSE ALL
110 P=B/A:Q=C/A:R=D/A
120 A=P:B=Q:C=R
130 F=1:X=0:E=1.E-08
140 GOSUB 600
150 Y=X-U/V
160 IF E>ABS(X-Y)THEN 18
0
170 X=Y:GOTO 140
180 IF F=1 THEN 800
190 PRINT #PN,R$:PRINT #
PN,USING 30,Y
200 G=A+Y:H=G*Y+B
210 M=G*G/4:J=M-H:G=-G/2
220 IF J<0 THEN J=-J:Z=S
QR(J):GOTO 700
230 Z=SQR(J):K=G+Z:N=G-Z
240 CALL TRUNC(K)
250 CALL TRUNC(N)
260 X=K
270 GOSUB 600
280 Y=X-U/V
290 IF E>ABS(X-Y)THEN P=
Y:PRINT #PN,R$:PRINT #PN
,USING 30;P:GOTO 310
300 X=Y:GOTO 270
310 X=N
320 GOSUB 600
330 Y=X-U/V
340 IF E>ABS(X-Y)THEN Q=
Y:PRINT #PN,R$:PRINT #PN
,USING 30;Q:GOTO 900
350 X=Y:GOTO 320
600 U=X*X*(X+A)+B*X+C
610 V=X*(3*X+2*A)+B
620 RETURN
700 IF ABS(G)<9.E-13 THE
N G=0
710 PRINT #PN,"Complex R
oots = "
720 PRINT #PN,"Real Part
":PRINT #PN,USING 30;G
730 PRINT #PN,"Imaginary
 Part"
740 PRINT #PN,"+/-";
750 PRINT #PN,USING 30,Z
760 GOTO 900
800 CALL TRUNC(Y)
820 F=0:X=Y:GOTO 140
900 INPUT "Another Probl
em (Y/N) ? ";A$
910 IF A$="Y"OR A$="y"TH
EN PAUSE 0:GOTO 20
920 END
1000 REM Subprograms
1001 SUB TRUNC(S)
1002 K=-INT(LOG(ABS(S)))
+5
1003 S=INT(S*10^K)*10^(-
K)
1004 SUBEND
```

NOTES ON PI-FINDING - Page 11A of the 16 October 1987 issue of the St. Petersburg Times includes an article by Ed Lion of UPI on the latest edition of the Guinness Book of Records. The article stated that "the world's most boring book" was found to be "a bound version in 19,900 pages of pi calculated to more than 33 million decimal places with a computer programmed by Japanese mathematician Yasumasa Kanada." The same article notes that "In the greatest feat of memory category, Hideaki Tomoyori, 55, of Japan extended 'pi' -- the geometric constant -- to a record 40,000 decimal places in 17 hours, 21 minutes."

The May 1985 issue of BYTE included an article "Computing Pi" by David J. Crawford. Subsequent "Letters" columns in BYTE have included a series of submissions on the calculation of pi for the past several months. Most of the routines are in BASIC.

SIGN(UM) IN PLEASE - C. Williamson. Robert Prins reports in V11N4P5 that the TI-95 SGN function returns a one for a zero input. Of course this is a corruption of the signum function (x/abs x) which returns a zero. At least the HP-41 calls its equivalent function SIGN; it returns zero for alpha input; it's still misleading.

Programming challenge: Consider only numeric input. Write a program for the TI-95 which simulates the TI-59's Op 10 (true signum function). Now do the reverse on the TI-59, that is simulate the TI-95 SGN function. To make things more interesting, use no direct comparisons. Explain your approach.

Editor's Note: In later correspondence Charlie reports that he didn't find any IEEE standard on the SGN or SIGN functions.

EXTENDED PRECISION MULTIPLICATION - V12N1P16 presented Larry Leeds' program for extended precision multiplication which was expected to handle problems with multipliers of up to 120 digits. Using multipliers of all nines Robert Prins found that the program would not yield correct answers for multipliers of more than 60 digits. Larry confirms that the program as listed on V12N1P16 yields incorrect answers for multipliers of more than 60 digits. That program can be modified to operate with 5 digit words by changing program steps 066 and 076 from 6 to 5. The resulting program will then yield correct results for multipliers of 20 words, or for multipliers of 100x100 digits. The execution times for a given number of input words will be the same as reported previously. The listing below includes those changes, and adds the capability to print the multipliers and the product. For operation without a printer change program step 155 from PRT to R/S.

User Instructions:

1. Partition the factors into 5 digit words starting at the right. The first word in each factor may have less than 5 digits.

2. Press A to initialize for entry of the first factor. Enter each 5 digit word starting at the left, and press R/S. Each word is printed as it is entered.

3. Press B to initialize for entry of the second factor. Enter each 5 digit word starting from the left, and press R/S. Each word is printed as it is entered.

4. To solve press C. See a flashing "1." in the display. Press 7 and then EE to enter fast mode. When the solution is complete the calculator will stop with a zero in the display.

5. Press D to obtain a printout of the product in 5 digit words, starting at the left.

Program Listing:

000	91	R/S	040	22	INV	080	69	OP	120	25	CLR	160	01	01	200	00	0
001	25	CLR	041	44	SUM	081	31	31	121	91	R/S	161	53	53	201	00	0
002	09	9	042	03	03	082	69	OP	122	72	ST*	162	98	ADV	202	00	0
003	42	STO	043	69	OP	083	32	32	123	00	00	163	25	CLR	203	00	0
004	01	01	044	21	21	084	97	DSZ	124	99	PRT	164	69	OP	204	00	0
005	03	3	045	97	DSZ	085	05	05	125	69	OP	165	99	99	205	00	0
006	00	0	046	04	04	086	00	00	126	20	20	166	91	R/S	206	00	0
007	42	STO	047	00	00	087	61	61	127	69	OP	167	00	0	207	00	0
008	02	02	048	13	13	088	25	CLR	128	25	25	168	00	0	208	00	0
009	05	5	049	43	RCL	089	66	PAU	129	69	OP	169	00	0	209	00	0
010	01	1	050	06	06	090	81	RST	130	27	27	170	00	0	210	00	0
011	42	STO	051	42	STO	091	76	LBL	131	61	GTO	171	00	0	211	00	0
012	03	03	052	01	01	092	11	A	132	01	01	172	00	0	212	00	0
013	73	RC*	053	42	STO	093	98	ADV	133	21	21	173	00	0	213	00	0
014	01	01	054	02	02	094	09	9	134	76	LBL	174	00	0	214	00	0
015	65	×	055	69	OP	095	69	OP	135	14	D	175	00	0	215	76	LBL
016	73	RC*	056	32	32	096	17	17	136	98	ADV	176	00	0	216	13	C
017	02	02	057	43	RCL	097	47	CMS	137	05	5	177	00	0	217	43	RCL
018	95	=	058	08	08	098	09	9	138	00	0	178	00	0	218	04	04
019	74	SM*	059	42	STO	099	42	STO	139	42	STO	179	00	0	219	85	+
020	03	03	060	05	05	100	00	00	140	03	03	180	00	0	220	43	RCL
021	69	OP	061	73	RC*	101	25	CLR	141	43	RCL	181	00	0	221	05	05
022	22	22	062	01	01	102	91	R/S	142	50	50	182	00	0	222	95	=
023	69	OP	063	55	÷	103	72	ST*	143	22	INV	183	00	0	223	42	STO
024	23	23	064	01	1	104	00	00	144	67	EQ	184	00	0	224	08	08
025	97	DSZ	065	52	EE	105	99	PRT	145	01	01	185	00	0	225	42	STO
026	05	05	066	05	5	106	69	OP	146	51	51	186	00	0	226	00	00
027	00	00	067	95	=	107	20	20	147	69	OP	187	00	0	227	85	+
028	13	13	068	75	-	108	69	OP	148	23	23	188	00	0	228	04	4
029	03	3	069	59	INT	109	24	24	149	69	OP	189	00	0	229	09	9
030	00	0	070	74	SM*	110	61	GTO	150	30	30	190	00	0	230	95	=
031	42	STO	071	02	02	111	01	01	151	22	INV	191	00	0	231	42	STO
032	02	02	072	95	=	112	02	02	152	52	EE	192	00	0	232	06	06
033	43	RCL	073	65	×	113	76	LBL	153	73	RC*	193	00	0	233	69	OP
034	07	07	074	01	1	114	12	B	154	03	03	194	00	0	234	38	38
035	42	STO	075	52	EE	115	98	ADV	155	99	PRT	195	00	0	235	04	4
036	05	05	076	05	5	116	03	3	156	69	OP	196	00	0	236	05	5
037	75	-	077	95	=	117	00	0	157	23	23	197	00	0	237	30	TAN
038	01	1	078	72	ST*	118	42	STO	158	97	DSZ	198	00	0	238	33	X^2
039	95	=	079	01	01	119	00	00	159	00	00	199	00	0	239	86	STF

AITKEN'S Δ^2 CONVERGENCE METHOD

- Jorge Valencia of Lima, Peru. When a first order iterative process for finding roots of a function is slow to converge, convergence can be accelerated by Aitken's second finite difference method, which finds a root for every guess of the solution. The results with the program below are fast, and may be surprisingly accurate, even with a guess which is far removed from the true root.

User Instructions

1. Rearrange the function, if necessary, in the form $x = \phi(x)$.

2. Enter the program and press GTO 072 LRN.

3. Enter $\phi(x)$ as a series of keystrokes. The latest value of x will be in the display register at step 072 and may be stored for further use in data register R00. Do not use data registers 01 through 06. End with RTN.

4. Leave LRN mode. Set any angular mode as required by the function o(x).

5. Enter the degree of accuracy and press B.

6. Enter a guess for the root. Press A to find the true root.

7. For a new root, go back to steps 5 and 6.

Program Listing

000	76	LBL	018	00	00	036	05	05	054	43	RCL	072	53	(	090	34	√X
001	11	A	019	72	72	037	33	X²	055	01	01	073	42	STO	091	35	1/X
002	42	STO	020	42	STO	038	95	=	056	61	GTO	074	00	00	092	85	+
003	01	01	021	03	03	039	35	1/X	057	00	00	075	65	×	093	53	(
004	71	SBR	022	75	-	040	94	+/-	058	04	04	076	01	1	094	01	1
005	00	00	023	43	RCL	041	85	+	059	43	RCL	077	00	0	095	02	2
006	72	72	024	02	02	042	43	RCL	060	01	01	078	65	×	096	02	2
007	42	STO	025	95	=	043	03	03	061	91	R/S	079	53	(	097	05	5
008	02	02	026	42	STO	044	95	=	062	76	LBL	080	53	(	098	75	-
009	75	-	027	05	05	045	42	STO	063	12	B	081	02	2	099	43	RCL
010	43	RCL	028	75	-	046	01	01	064	45	Yˣ	082	00	0	100	00	00
011	01	01	029	43	RCL	047	15	E	065	32	X:T	083	02	2	101	33	X²
012	95	=	030	04	04	048	50	I×I	066	01	1	084	05	5	102	54	)
013	42	STO	031	95	=	049	32	X:T	067	95	=	085	75	-	103	34	√X
014	04	04	032	42	STO	050	77	GE	068	91	R/S	086	43	RCL	104	35	1/X
015	43	RCL	033	06	06	051	00	00	069	76	LBL	087	00	00	105	54	)
016	02	02	034	55	÷	052	59	59	070	15	E	088	33	X²	106	54	)
017	71	SBR	035	43	RCL	053	32	X:T	071	75	-	089	54	)	107	95	=
															108	92	RTN

Sample Problems: Steps 072 through 108 in the listing are for a solution of the ladder problem from V12N1 (equation 7 from V12N1P11). If you set the accuracy at 1e-10 and start with a guess of 32 you will get 31.8174591 (31.81745909539 in the display register) in 65 seconds.

The steps at the right are for Peter Messer's benchmark cubic test of $x^3 - 2x^2 + (4/3)x - 2/9 = 0$ from V11N4P16. If you set the accuracy at 1e-10 and start with a guess of zero you will get the real root, 0.2466929834 (.2466929833616 in the display register), in 25 seconds.

072	53	(	084	02	2
073	42	STO	085	55	÷
074	00	00	086	09	9
075	65	×	087	54	)
076	33	X²	088	94	+/-
077	75	-	089	65	×
078	02	2	090	03	3
079	65	×	091	55	÷
080	43	RCL	092	04	4
081	00	00	093	95	=
082	33	X²	094	92	RTN
083	75	-			

TI-59 PLUS EE MODULE FOR SALE

MORE PPX PROGRAM AVAILABILITY - V9N5P19 set up an informal program exchange to provide access to programs which were formerly available from the PPX Exchange. Additional programs were listed in V10N1P19 and in V10N3P20. The following list shows 56 additional programs which have been made available by members.

1F	038001	- Inventory Report
12A	058003	- Small Business General Payroll
1	098009	- Purchase Agreement Analysis
1	098010	- Sheet Metal Material Pricing
12F	108005	- Small Business Accounting
P	308011	- Cholesky Matrix Solver
P	308060	- Matrix Multiply
3P	308093	- Eigenvalues
P	308094	- Matrix Multiply
P	338038	- Simultaneous Equations
P	348011	- Double Integral
3P	348017	- Triple Integral
P	348019	- Double Integral by Riemann Sums
P	398013	- A/B = C/D
GP	398019	- Spherical Triangles
P	398068	- Ordinary and Partial Derivatives
P	398098	- Coordinate Transformations
P	398190	- Arc Length, Surface of Revolution
1P	398205	- Simultaneous Equations
P	398228	- Partial Derivatives Notation
P	398247	- Circle Tangent to Three Other Random Circles
P	398256	- Finite Differences
P	398264	- F(x) Any Delta-x with List and PLot
P	408051	- Electric Field Potentials
P	628051	- Flexibility and Stiffness Coefficients
MP	628052	- Curved Structures
MP	628065	- Mohr's Circle from Strain Gage
EP	628159	- Vibration - Three Degrees of Freedom
P	628164	- Vibration - Single Degree of Freedom
EP	628172	- Moment of Inertia
P	628186	- Continuous Beams (4 Spans)
P	648114	- RLC Circuits
P	648115	- Resistance Networks
P	658027	- Resistor Triangle Measurements
FN	658030	- Distance by Geographical Coordinates
N	658045	- Ground Station G/T
N	658115	- Digital Circuit Truth Table Generation
P	658129	- Heat Sinks for Semi-conductors
N	658136	- Antenna Heights Required for L.O.S. Radio Path
N	658140	- Microwave Path Profile Calculations
N	658150	- Azimuths, Path Length & Coordinates of Intermediate Points for Radio Path
P	658159	- Inverse LaPlace Transforms
N	658160	- Microwave Path and Parabolic Antenna Parameters
N	658161	- Microwave Beam Clearance & Parabolic Antenna Parameters
N	658162	- Over the Water Microwave Path Reflection Points
N	658165	- F.M. Microwave System Noise
N	658166	- 4 and 6 GHZ Frequency Diversity Combinations
N	658167	- Unavailability of Microwave Paths due to Multipath ...
N	658168	- Microwave Path Calculations by Inverse Position
N	658171	- Earth Station "Lookarc" Parameters to Geo Satellites

More PPX Programs - (cont)

P 668020 - Rectangular/Delta Strain Gages
P 668031 - One Dimension Heat Conduction
18P 668075 - 3-D Mohr's Circle
P 668076 - 2-D Mohr's Circle
P 669079 - Section Properties
P 668149 - Rectangular Fin Thermal Analysis
P 668174 - Maximum Stress - Rectangular Flat Panel
P 688007 - Second Order Underdamped System Response
P 698014 - 45 Degree Rosette Analysis- (Stress)
P 698036 - Loads due to Pitch and Roll of Ship
P 798047 - Developed Length of Tubes
P 798057 - Center Finder for CMM
1D 908018 - Universal Plotter
1F 908050 - Store and Recall Programs
19 908105 - Improved Cartesian Graph
1 908116 - Alpha Printout, System Order Status
P 908182 - Advanced Use of Registers
P 908183 - Address Labels
P 908211 - Use of "T" Registers
P 908215 - Simplified Triangles
P 968014 - Music Transposer
P 988058 - Sailboat Race Navigation

Code 1 means the programs are available on a loan basis from TI PPC NOTES. Send one dollar (two dollars for overseas) to cover shipping costs for each program you wish to borrow. It is understood that the programs will be returned promptly to be available for other members.

Codes 2 through M were defined in earlier issues.

Code N means the programs are available from Val Barron, 90 Cedar Point Drive, Williams Bay, WI 53191. Send a stamped and self addressed envelope for details.

Code P means the programs are available from James Taylor, P. O. Box 174, MA 01945. Send a stamped and self addressed envelope for details.

USED HARDWARE FOR SALE - One hundred dollars for the complete package.

TI-59 Calculator
PC-100C Printer
Master Library Module
Real Estate/Investment Module
Business Decisions Module
Leisure Library Module
Math/Utilities Module
12 Blank Magnetic Cards
Personal Programming Manual
TI-59 Workbook

The owner reports that the calculator passes all the tests in V12N1P4-7. Write to Thomas P. Nyenhuis, Milwaukee Metal Products, Inc., 8000 West Florist Avenue, Milwaukee WI 53218 or call (414)-463-2090.

USED HARDWARE FOR SALE - $125 for the complete package.

TI-59 Calculator
PC-100C Printer
40 Magnetic Cards
12 Rolls Printer Paper
Leisure Library Module
EE Specialty Pakette
59 Fun Pakette
Programmable Calculator Sourcebook

The calculator passes the diagnostics in V12N1P4-7. The battery pack is dead. Write to John Trask, 10800 Hesby Street, North Hollywood, CA 91601.

<u>EVALUATION OF POLYNOMIAL CURVE FITTING PROGRAMS</u> - V12N1P24/25 presented a program by William Hood for curve fitting using orthogonal polynomial techniques. A primary reason for using such techniques is reduced susceptibility to ill-conditioning; however, the sample provided with Hood's article in BYTE did not illustrate that effect. V12N2P25 reported that for the sample problem the sums of the squares of the residuals were equal to nine decimal places for the orthogonal polynomial method, for the row reduction program on V12N1P14 and for the V11N4P12 solution using the Math module of the TI-74.

One question is what are the correct coefficients of the solution. Richard Spurrior calculated the solution in quadratic precision using a version of Hood's program. The first twenty digits of his results for solutions with second through fourth degree polynomials are (constant terms first):

Degree	2	3	4
A0	5.3993800361687283825	0.49716955875172593O4	-0.57002471695575463 53
A1	1.5004168045476485057	5.7929781587696064511	7.1999665801335072608
A2	-0.0326693617568390723	-0.7139797262710357798	-1.1131337063816293808
A3		0.0281457952959845654	0.0669397883174221636
A4			-0.0012073187307998324
Variance	7.5798137801289391680	1.0206238865340448785	0.9917402194039777401
R-squared	0.8307672547295986789	0.9804680860539975219	0.9841840328617498167

George Thomson proposed that the so-called "Wampler quintics" would provide better insight into the relative capabilities of various curve fitting programs. The method was described in Roy H. Wampler's paper "An Evaluation of Linear Least Squares Computer Programs" in the April-June 1969 issue of the <u>Journal of Research</u> of the National Bureau of Standards. Two test polynomials were proposed:

$$y = 1 + x + x^2 + x^3 + x^4 + x^5$$

$$y = 1 + x/10 + (x/10)^2 + (x/10)^3 + (x/10)^4 + (x/10)^5$$

where 21 input data pairs are defined for values of x from 0 to 20. Thus, for the first quintic the y values range from 1 to 3368421, and for the second quintic the y values range from 1 to 63. The paper also defined a measure (C) of the number of correct decimal digits in a coefficient of the fitted polynomial as the negative log of the magnitude of the relative error. If a coefficient is exact the number of digits with which the machine computes is used. The paper uses the average of the C values for the six coefficients as the figure of merit of the solution. George Thomson prefers the use of the minimum C value as a figure of merit. Others like to consider the standard error of the residuals as well.

The following table compares various figures of merit for the first quintic for four programs which have been published in previous issues of TI PPC Notes:

Machine	TI-59	TI-74	TI-74	TI-74
Program	V9N2P20	V11N4P12	V12N1P14	V12N1P24
Method	Mat Inv	Mat Inv	Row Red	Orth Poly
C(ave)	3.65	4.36	5.62	7.41
C(min)	2.16	2.62	4.15	6.05
S.E.	1.95E-03	2.54E-01	2.26E-05	2.95E-06

Evaluation of Polynomial Curve Fitting - (cont)

Preliminary tests showed that the second quintic, the one with tens in the denominators, did not provide as much differentiation between the results from the different programs. Accordingly, the results for the second quintic have not been presented here.

Examination of the figures of merit on page 16 reveals one anomaly: the number of correct digits in the coefficients is slightly higher for the TI-74 matrix inversion routine than for the TI-59 matrix inversion program; however, the standard error of the residuals for the TI-59 program is two orders of magnitude smaller than for the TI-74 program. Examination of the details of the solutions for the four programs shows why:

```
A1 =  1.003396496
A2 =  0.9932158788
A3 =  1.002457193
A4 =  0.9996749308
A5 =  1.000017923
A6 =  0.9999996501

d1 =   .0033964957
d2 =  -.0012379293
d3 =  -.002667963
d4 =  -.0022512867
d5 =  -.000999407
d6 =   .000380358
d7 =   .001442883
d8 =   .00196327
d9 =   .00189487
d10 =  .00132724
d11 =  .000444
d12 = -.0005187
d13 = -.0013138
d14 = -.0017271
d15 = -.0016179
d16 = -.0009602
d17 =  .000112
d18 =  .00127
d19 =  .001936
d20 =  .001256
d21 = -.001952

Mean = .000008425

S.E. = .001952144
```

V9N2P20

```
A1 =  .9989
A2 =  1.0024
A3 =  .9996
A4 =  1.000067
A5 =  .999999
A6 =  1.0000001

d1 =  .0011
d2 = -.0009661
d3 = -.0026232
d4 = -.0042523
d5 = -.0062344
d6 = -.0089625
d7 = -.0128536
d8 = -.0183607
d9 = -.0259848
d10 = -.0362869
d11 = -.0499
d12 = -.0675411
d13 = -.0900232
d14 = -.1182673
d15 = -.1533144
d16 = -.1963375
d17 = -.248654
d18 = -.311736
d19 = -.387225
d20 = -.476942
d21 = -.5829

Mean Error =
-.1271938636

S.E. =  .2543009854
```

V11N4P12

```
A1 =  1.000030632
A2 =  .99992933
A3 =  1.000027049
A4 =  .9999963001
A5 =  1.000000209
A6 =  .9999999959

d1 = -.000030632
d2 =  .0000164843
d3 =  2.890386E-05
d4 =  .0000219328
d5 =  .0000068624
d6 = -.0000085344
d7 = -.0000195047
d8 = -.00002381
d9 = -.00002128
d10 = -.00001319
d11 = -.00000192
d12 =  .00000965
d13 =  .00001857
d14 =  .00002241
d15 =  .00001966
d16 =  .00001026
d17 = -.000004
d18 = -.000019
d19 = -.000026
d20 = -.000016
d21 =  .000027

Mean = -1.017975E-07

S.E. =  2.264843E-05
```

V12N1P14

```
C(0) =  1.000000207
C(1) =  .99999912
C(2) =  1.000000449
C(3) =  .99999992
C(4) =  1.000000005
C(5) =  .9999999999

Residual Variance =
 8.723924E-12

Coeff of Det (R^2) =
1.000000

D(1) =  .0000002074
D(2) = -2.98076E-07
D(3) = -3.13736E-07
D(4) = -.0000001451
D(5) =  .0000000014
D(6) =  .0000000049
D(7) = -.0000001838
D(8) = -.00000056
D(9) = -.00000108
D(10) = -.00000166
D(11) = -.00000229
D(12) = -.00000265
D(13) = -.00000303
D(14) = -.00000291
D(15) = -.00000273
D(16) = -.00000264
D(17) = -.000002
D(18) = -.000002
D(19) = -.000003
D(20) = -.000004
D(21) = -.000007
```

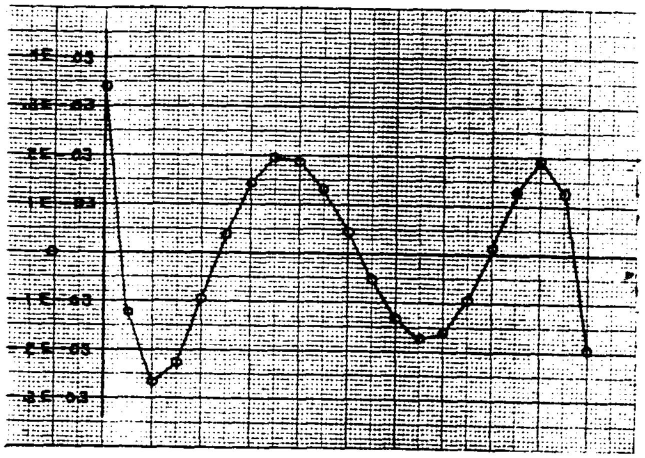
V12N1P24

The residuals for the V11N4P12 solution (the TI-74 using the Mathematics module matrix inversion routine) are all negative except for the first residual which is very small. Clearly, the mean of the residuals, which should be zero for a least squares polynomial curve fit, can not be zero. The residuals for the V12N1P24 solution (the TI-74 using Hood's orthogonal polynomial routine) show a similar negative bias, albeit much smaller in magnitude. For both programs the mean of the residuals was of the same order of magnitude as the standard error for the residuals. George Thomson had found a similar effect with a version of Hood's program on his PC where he observed a well defined ramp in the residuals even though the mean of the residuals was very small.

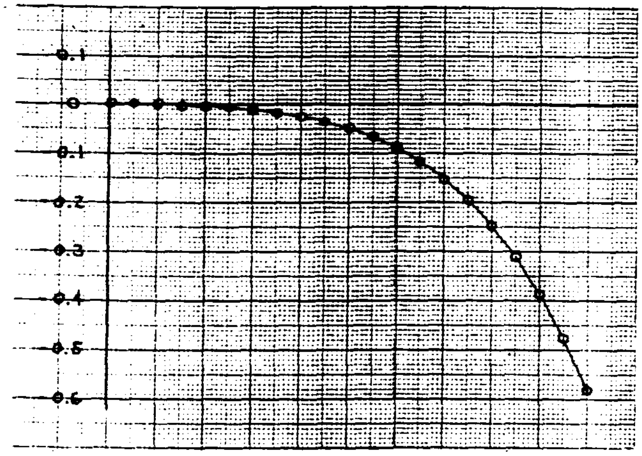
In contrast, the residuals for the V9N2P20 solution (the TI-59 using the ML-02 matrix inversion routine) and the V12N1P14 solution (the TI-74 using a row reduction routine) have both positive and negative signs such that the mean of the residuals is much smaller than the standard error for the residuals.

What is not evident from the tables is that the residuals for all four programs seem to be on smooth error curves. See the figures on the next page, but remember that the vertical scales vary considerably.

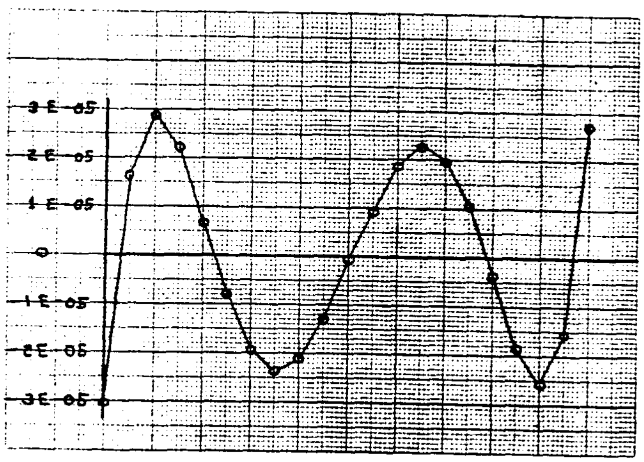
Evaluation of Polynomial Curve Fitting - (cont)



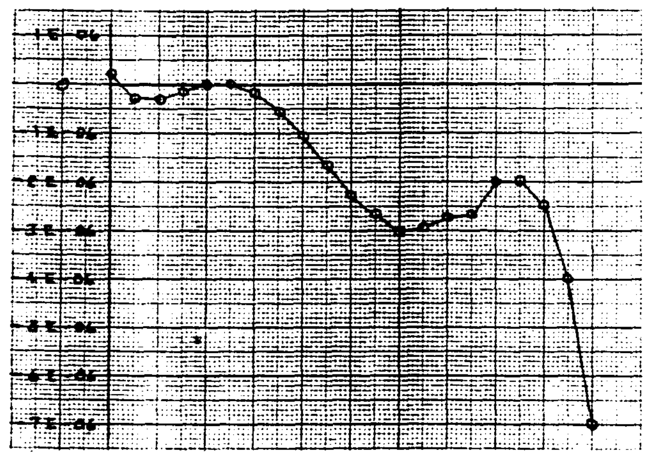
V9N2P20 - TI-59 Matrix Inversion



V11N4P12 - TI-74 Matrix Inversion



V12N1P14 - TI-74 Row Reduction



V12N1P24 - TI-74 Orthogonal Polynomial

The nature of these residual curves suggests that one might be able to find another set of polynomial coefficients which would fit the residual curves. Then, if the coefficients from the solution on the residuals were combined with the original coefficients, the resulting coefficients might provide a better fit to the original data. The table at the right shows the results after such an exercise with the polynomial regression program for the TI-59 (V9N2P20). The new coefficients have a C(ave) of 8.53 digits and a C(min) of 6.54 digits, a four digit improvement over the solution on page 17. The mean of the residuals is reduced by a factor of seven, and the standard error is reduced by a factor of over 1600. That kind of improvement in performance comes very hard with the TI-59 program where

1. The original set of 21 data pairs must be entered and each data pair takes about 20 seconds to be accepted.
2. The regression must be completed.
3. The original set of data pairs must be re-entered and the residuals calculated.
4. The set of data pairs for the residual errors must be entered. Again, each data pair requires about 20 seconds to be accepted.
5. The regression on the residuals must be completed, and the results combined with the original coefficients by hand.
6. The original data pairs must be re-entered and the new residuals calculated. This step can be easy this time if the user remembered to store the data entered in step 3 above.

A1 =	1.000000108	
A2 =	0.9999997117	
A3 =	1.000000121	
A4 =	0.9999999824	
A5 =	1.000000001	
A6 =	1.	
d1 =	1.08488-07	
d2 =	-7.6282	-08
d3 =	-1.1218	-07
d4 =	-7.04	-08
d5 =	0	
d6 =	6.6	-08
d7 =	1.02	-07
d8 =	1.1	-07
d9 =	1.	-07
d10 =	1.	-07
d11 =	-1.	-07
d12 =	2.	-07
d13 =	1.	-07
d15 =	5.	-07
d16 =	1.1	-06
d17 =	1.	-06
d18 =	1.	-06
d19 =	1.	-06
d20 =	1.	-06
d21 =	4.	-06
Mean =	5.49-07	
S.E. =	1.20-06	

Evaluation of Polynomial Curve Fitting - (cont)

Setting up an interative solution is much easier on the TI-74 where sufficient memory is available to hold the intermediate results. One question is how many iterations are needed. The following table helps answer that question for the solution which uses the matrix inversion routine from the Math module (V11N4P12).

A1 = .9989000000000E+00	A1 = .9999999833080E+00	A1 = .9999999830544E+00	A1 = .9999999549789E+00
A2 = .1002400000000E+01	A2 = .1000000061400E+01	A2 = .1000000052284E+01	A2 = .1000000127748E+01
A3 = .9996000000000E+00	A3 = .9999999685980E+00	A3 = .9999999762683E+00	A3 = .9999999445710E+00
A4 = .1000067000000E+01	A4 = .1000000005442E+01	A4 = .1000000003639E+01	A4 = .1000000008296E+01
A5 = .9999990000000E+00	A5 = .9999999996273E+00	A5 = .9999999997836E+00	A5 = .9999999995032E+00
A6 = .1000000100000E+01	A6 = .1000000000009E+01	A6 = .1000000000004E+01	A6 = .1000000000010E+01
d1 = 1.10000000000E-03	d1 = 1.66920000000E-08	d1 = 1.69455600000E-08	d1 = 4.50210600000E-08
d2 = -9.66100000000E-04	d2 = -1.83840000000E-08	d2 = -1.50330000000E-08	d2 = -3.51070000000E-08
d3 = -2.62320000000E-03	d3 = -1.83600000000E-08	d3 = -1.84730000000E-08	d3 = -4.74980000000E-08
d4 = -4.25230000000E-03	d4 = -3.80000000000E-09	d4 = -8.00000000000E-09	d4 = -2.55000000000E-08
d5 = -6.23440000000E-03	d5 = 1.14000000000E-08	d5 = 5.90000000000E-09	d5 = 6.90000000000E-09
d6 = -8.96250000000E-03	d6 = 1.93000000000E-08	d6 = 1.66000000000E-08	d6 = 3.42000000000E-08
d7 = -1.28536000000E-02	d7 = 1.64000000000E-08	d7 = 2.09000000000E-08	d7 = 4.82000000000E-08
d8 = -1.83607000000E-02	d8 = .00000000000E+00	d8 = 2.00000000000E-08	d8 = 4.00000000000E-08
d9 = -2.59848000000E-02	d9 = -1.00000000000E-08	d9 = 1.00000000000E-08	d9 = 3.00000000000E-08
d10 = -3.62869000000E-02	d10 = -4.00000000000E-08	d10 = .00000000000E+00	d10 = 1.00000000000E-08
d11 = -4.99000000000E-02	d11 = -7.00000000000E-08	d11 = -1.00000000000E-08	d11 = -2.00000000000E-08
d12 = -6.75411000000E-02	d12 = -9.00000000000E-08	d12 = .00000000000E+00	d12 = -4.00000000000E-08
d13 = -9.00232000000E-02	d13 = -1.10000000000E-07	d13 = 1.00000000000E-08	d13 = -3.00000000000E-08
d14 = -1.18267300000E-01	d14 = -1.20000000000E-07	d14 = 4.00000000000E-08	d14 = 1.00000000000E-08
d15 = -1.53314400000E-01	d15 = -1.40000000000E-07	d15 = 1.10000000000E-07	d15 = 7.00000000000E-08
d16 = -1.96337500000E-01	d16 = -1.70000000000E-07	d16 = 2.00000000000E-07	d16 = 1.60000000000E-07
d17 = -2.48654000000E-01	d17 = .00000000000E+00	d17 = 1.00000000000E-06	d17 = 1.00000000000E-06
d18 = -3.11736000000E-01	d18 = -1.00000000000E-06	d18 = .00000000000E+00	d18 = 1.00000000000E-06
d19 = -3.87225000000E-01	d19 = -1.00000000000E-06	d19 = .00000000000E+00	d19 = .00000000000E+00
d20 = -4.76942000000E-01	d20 = -1.00000000000E-06	d20 = 1.00000000000E-06	d20 = .00000000000E+00
d21 = -5.82900000000E-01	d21 = -2.00000000000E-06	d21 = 1.00000000000E-06	d21 = 1.00000000000E-06
Mean Error = -.1332507143	Mean Error = -2.727025E-07	Mean Error = 1.618495E-07	Mean Error = 1.550579E-07
S.E. = .2543009854	S.E. = 6.875774E-07	S.E. = 4.513731E-07	S.E. = 4.506149E-07
S2 = .9700348675	S2 = 7.09144E-12	S2 = 3.056066E-12	S2 = 3.045806E-12
C = 4.35837688	C = 8.538493149	C = 8.696504536	C = 8.313573852

The baseline program from V11N4P12 was modified to print out the solution and the residuals in exponential format, and to add solutions for the sum of the squares of the residuals (S2) and the average figure of merit (C). The left-hand column is the baseline solution, the same as in the second column on page 17. The next three columns reflect the result of additional iterative solutions on the residuals. The table shows that there is substantial improvement after the first iteration, but little improvement after additional iterations. The coefficients after the first iteration show an improvement of 4.2 digits and the standard error is reduced by a factor of 370,000.

Tests of the iterative technique with the solution based on the row reduction routine (V12N1P14) also showed substantial improvement after one iteration but little improvement from additional iterations. The coefficients after the first iteration show an improvement of 2.6 digits and the standard error is reduced by a factor of 60.

Programs which provide a single iteration for both the matrix inversion routine and the row reduction routine appear on pages 20 and 21. Of course execution time is increased. The matrix inversion program requires 135 seconds to solve the quintic. The row reduction program requires 110 seconds.

In a future issue we will examine the effect of iterative techniques on the orthogonal polynomial routine, and will address polynomial regression and iterative techniques for the TI-95.

<u>ITERATIVE REGRESSION USING THE TI-74 MATH MODULE</u> - This program is an iterative version of the program which appeared in V11N4P12/13. The least squares solution was moved to a subroutine (lines 900 to 995), an array for accumulation of the coefficients was added, and a working array, Z(50), to hold the dependent variables for use in the solution was added. Lines 130 and 140 provide for data input. Lines 150-170 select the order of the solution, clear the coefficient array, and transfer the input independent variables to the working array for use in the first pass least squares solution. The GOSUB 900 command at line 200 provides the first pass. The GOSUB 900 command at line 210 provides the second pass using the residuals from the first pass as the dependent variables. Additional iterations could be added by inserting more GOSUB 900 commands between lines 210 and 300. Lines 300-650 provide output of the solution to a printer or to the display. The PRINT USING commands at lines 340 and 550 define exponential format for maximum resolution in the output. Lines 700-740 provide options for different solutions without re-entry of the input data.

<u>User Instructions</u>

1. The Mathematics software module must be installed.

2. A full set of prompts are available.

3. The user defined functions must be defined by F(1) through F(N) in subroutine 800. To obtain a constant in the solution define one of the functions as one as in line 810 of the program below. Each pair of X,Z values are available in turn at entry to this subroutine. The user defined functions in the program listing below provides for a polynomial solution.

```
100 DIM A(8,8),B(8),C(8,8),C1(8),F(8),X(50),Y(50),Z(50)
110 CALL UP("Least Squares Fit",PN)
120 PRINT "Are the functions correct?":PAUSE 2
130 INPUT "Number of Data Pairs? ";K
140 CALL AU("X","Y",X(),Y(),1,K,PN)
150 INPUT "Order of the solution? ";N
160 FOR I=1 TO N:C1(I)=0:NEXT I
170 FOR I=1 TO K:Z(I)=Y(I):NEXT I
200 GOSUB 900
210 GOSUB 900
300 PRINT #PN
310 FOR I=1 TO N
320 X$="A"&STR$(I)&" = "
330 PRINT #PN,X$;
340 PRINT #PN,USING"##.##########^^^^",C1(I)
350 IF PN=0 THEN PAUSE
360 NEXT I
370 PRINT #PN
500 INPUT "Display Residuals <Y/N>? ";A$
510 IF A$="N"OR A$="n"THEN 600
520 FOR L=1 TO K
530 P$="d"&STR$(L)&" = "
540 PRINT #PN,P$;
550 PRINT #PN,USING"##.##########^^^^",Z(L)
560 IF PN=0 THEN PAUSE
570 NEXT L
580 PRINT #PN
600 PRINT #PN,"Mean Error = ";S1/K
610 IF PN=0 THEN PAUSE
620 PRINT #PN
630 PRINT #PN,"S.E. = ";SQR(S2/(K-N))
640 IF PN=0 THEN PAUSE
650 PRINT #PN
700 INPUT "Edit Input Data <Y/N>? ";A$
710 IF A$="N"OR A$="n"THEN 730
720 CALL AU("X","Y",X(),Y(),1,K,PN)
730 INPUT "Different order <Y/N>? ";A$
740 IF A$="Y"OR A$="y"THEN 150
799 STOP
800 REM User Defined Functions
810 F(1)=1
820 FOR W=2 TO N
830 F(W)=F(W-1)*X(L)
840 NEXT W
890 RETURN
900 REM Normal Equation Subroutine
905 FOR I=1 TO N:FOR J=1 TO N
910 A(I,J)=0:NEXT J
915 B(I)=0:NEXT I
920 FOR L=1 TO K:GOSUB 800
925 FOR I=1 TO N:FOR J=1 TO N
930 A(I,J)=A(I,J)+F(I)*F(J):NEXT J
935 B(I)=B(I)+F(I)*Z(L):NEXT I
940 NEXT L
945 CALL MATS(A(,),C(,),B(),1,1,5,1,N,1,R)
950 IF R=-1 THEN PRINT "Matrix is singular":PAUSE
955 FOR I=1 TO N:C1(I)=C1(I)+A(1,I):NEXT I
960 S1=0:S2=0
965 FOR L=1 TO K:GOSUB 800
970 YF=0:FOR J=1 TO N
975 YF=YF+C1(J)*F(J):NEXT J
980 Z(L)=Y(L)-YF
985 S1=S1+Z(L):S2=S2+Z(L)*Z(L)
990 NEXT L
995 RETURN
```

<u>ITERATIVE</u> <u>REGRESSION</u> <u>USING</u> <u>A</u> <u>ROW</u> <u>REDUCTION</u> <u>ROUTINE</u> - This program is an iterative version of the program which appeared in V12N1P14. In this case the least squares solution was moved to a subprogram SOLVE (lines 1000 to 1400) to provide another demonstration of the subprogram capability, including the calling of a subroutine within the subprogram. A subroutine mechanization such as that used on page 20 would actually yield a somewhat more efficient program. An array for accumulation of the coefficients was added, and two working arrays, R(50) and Z(50), were added.

Lines 105 to 190 provide for setup and data input. Lines 200-210 select the order of the solution, and clear the coefficient array. The CALL SOLVE command at line 300 provides the first pass. The CALL SOLVE command at line 310 provides the second pass using the residuals from the first pass as the dependent variables. Additional iterations could be added by inserting more CALL SOLVE commands between lines 310 and 400. Lines 400-650 provide output of the solution to a printer or to the display. The PRINT USING commands at lines 425 and 545 define exponential format for maximum resolution in the output. Lines 700-790 provide options for different solutions without re-entry of the input data. The user defined functions inside the subprogram at steps 1300-1350 are for a polynomial regression.

```
100 DIM A(8,8),B(8),C(8)
,F(8),X(50),Y(50),Z(50),
R(50)
105 INPUT "Use Printer <
Y/N>? ";A$
110 IF A$="Y"OR A$="y"TH
EN PN=1 ELSE 125
115 INPUT "Device Code ?
 ";P$
120 OPEN #1,P$,OUTPUT
125 PRINT "Are the funct
ions correct?":PAUSE 2
130 INPUT "Number of Dat
a Pairs? ";K
140 FOR I=1 TO K
150 A$="X"&STR$(I)&" = "
:INPUT A$;X(I)
160 IF PN<>0 THEN PRINT
#PN,A$,X(I)
170 A$="Y"&STR$(I)&" = "
:INPUT A$;Y(I)
180 IF PN<>0 THEN PRINT
#PN,A$,Y(I)
185 PRINT #PN:IF E$<>""T
HEN 700
190 NEXT I
200 INPUT "Order of the
solution? ";N
210 FOR I=1 TO N:C(I)=0:
NEXT I
300 CALL SOLVE(X(),Y(),Y
(),R(),C(),N,K,S1,S2)
310 CALL SOLVE(X(),Y(),R
(),R(),C(),N,K,S1,S2)
400 FOR I=1 TO N
410 X$="A"&STR$(I)&" = "
420 PRINT #PN,X$;
425 PRINT #PN,USING"##.#
##########^^^^",C(I)
430 IF PN=0 THEN PAUSE
440 NEXT I
450 PRINT #PN
500 INPUT "Display Resid
uals <Y/N>? ";A$
510 IF A$="N"OR A$="n"TH
EN 600
520 FOR L=1 TO K
530 P$="d"&STR$(L)&" = "
540 PRINT #PN,P$;
545 PRINT #PN,USING"##.#
##########^^^^",R(L)
550 IF PN=0 THEN PAUSE
560 NEXT L
570 PRINT #PN
600 PRINT #PN,"Mean = ";
S1/K
610 IF PN=0 THEN PAUSE
620 PRINT #PN
630 PRINT #PN,"S.E. = ";
SQR(S2/(K-N))
640 IF PN=0 THEN PAUSE
650 PRINT #PN
700 INPUT "Edit Input Da
ta <Y/N>? ";E$
710 IF E$="N"OR E$="n"TH
EN 780
720 INPUT "Which Data Pa
ir to Edit? ";I
730 IF I<1 OR I>K THEN 7
00
740 GOTO 150
780 INPUT "New Solution
<Y/N>? ";A$
790 IF A$="Y"OR A$="y"TH
EN 200
799 STOP
1000 SUB SOLVE(X(),Y(),Z
(),R(),C(),N,K,S1,S2)
1010 FOR I=1 TO N:FOR J=
1 TO N
1015 A(I,J)=0:NEXT J
1020 B(I)=0:NEXT I
1025 FOR L=1 TO K
1030 GOSUB 1300
1035 FOR I=1 TO N:FOR J=
1 TO N
1040 A(I,J)=A(I,J)+F(I)*
F(J):NEXT J
1045 B(I)=B(I)+F(I)*Z(L)
:NEXT I
1050 NEXT L
1100 FOR L=1 TO N
1105 P=A(L,L)
1110 FOR J=L TO N
1115 A(L,J)=A(L,J)/P:NEX
T J
1120 B(L)=B(L)/P
1125 FOR I=1 TO N
1130 IF I=L THEN 1155
1135 G=A(I,L)
1140 FOR J=L TO N
1145 A(I,J)=A(I,J)-G*A(L
,J):NEXT J
1150 B(I)=B(I)-G*B(L)
1155 NEXT I
1160 NEXT L
1200 FOR I=1 TO N:C(I)=C
(I)+B(I):NEXT I
1205 S1=0:S2=0
1210 FOR L=1 TO K:GOSUB
1300
1215 YF=0:FOR J=1 TO N
1220 YF=YF+C(J)*F(J):NEX
T J
1225 R(L)=Y(L)-YF
1230 S1=S1+R(L):S2=S2+R(
L)*R(L)
1235 NEXT L
1240 SUBEXIT
1300 REM USER DEFINED FU
NCTIONS
1310 F(1)=1
1320 FOR W=2 TO N
1330 F(W)=F(W-1)*X(L)
1340 NEXT W
1350 RETURN
1400 SUBEND
```

KAPREKAR'S CONSTANT, 6174: A PERSISTENT DIFFERENCE - W. Widmer. D. R. Kaprekar, of Devali, India, has discovered many curious properties of integer numbers. Among these is his Kaprekar Constant, 6174. Kaprekar also notes (Journal of Recreational Mathematics, V13N1P2, 1980-1981) that this "constant" is also a Harshad Number, or one integrably divisible by the sum of its digits (i.e., 6174 mod 18 = 0). He states that "Harshad" is a Sanskrit word meaning "giving joy." Perhaps the following discussion and program will provide amusement, if not, indeed, joy!

Israel Cohen briefly notes, without proof (Journal of Recreational Mathematics, V12N1P78) an interesting property of the number 6174 which he calls "the persistent difference." Take any positive four-digit number, N(0)<10000, whose digits are not all equal. Permute the four digits, including zeroes, to form the maximum integer combination, M; also reverse this order to give the minimum (least), L. Then take the difference, (M - L) = N(1). Continue this operation on N(1), and then repeatedly to form the integer sequence N(1), N(2), ... N(i). It is conjectured that this sequence will always yield the number 6174 = N(i) for any N(0) as above specified. If any of our PPC members can prove this conjecture, I should be grateful to learn of this. It would also be interesting to know if there are higher orders of N, say of 5 or more digits, which would yield such a "persistent difference" (not of course, 6174). I have tried this for isolated samples of 5, 6, 7, 8, 9, 10, 12 and 16 digit integers and obtain cyclically recurring sequences rather than a single persistent (constant difference). Three-digit numbers yield a constant difference of 495, not a Harshad number. Two digit numbers cycle into 9, 81, 63, 27, 45, 9, ... The 10-digit number 9876543210 does give a constant 9753086421 and the 9 digit number 876543210 produces a constant 864197532. But not all 10-digit and 9-digit numbers do this. Again, the constants are not Harshad numbers.

The TI-58/59 program below is for the four digit case which yields 6174. It pauses for the successive differences N(1), N(2), ... N(i) = 6174 for a given N(0) abcd. To use the program, input the four digits in sequence and press A after each input. Then press R/S. The run will stop with N(i) = 6174 in the t register and i in the display.

Example: For N(0) = 2584 press 2 A 5 A 8 A 4 A R/S and after about seventy seconds see a 7 in the display.

```
000 76 LBL   032 01 01    064 61 GTO   096 44 SUM   128 82 82    160 00 0
001 11 A     033 77 GE    065 00 00    097 07 07    129 66 PAU   161 49 PRD
002 48 EXC   034 00 00    066 23 23    098 65 x     130 42 STO   162 08 08
003 04 04    035 40 40    067 43 RCL   099 01 1     131 08 08    163 43 RCL
004 48 EXC   036 48 EXC   068 05 05    100 00 0     132 01 1     164 08 08
005 03 03    037 02 02    069 32 X:T   101 54 )     133 00 0     165 59 INT
006 48 EXC   038 42 STO   070 43 RCL   102 44 SUM   134 00 0     166 42 STO
007 02 02    039 01 01    071 04 04    103 06 06    135 00 0     167 03 03
008 48 EXC   040 43 RCL   072 42 STO   104 43 RCL   136 22 INV   168 22 INV
009 01 01    041 03 03    073 07 07    105 02 02    137 49 PRD   169 44 SUM
010 91 R/S   042 32 X:T   074 42 STO   106 65 x     138 08 08    170 08 08
011 06 6     043 43 RCL   075 06 06    107 01 1     139 43 RCL   171 43 RCL
012 01 1     044 02 02    076 01 1     108 00 0     140 08 08    172 08 08
013 07 7     045 77 GE    077 00 0     109 54 )     141 59 INT   173 65 x
014 04 4     046 00 00    078 00 0     110 44 SUM   142 42 STO   174 01 1
015 42 STO   047 52 52    079 00 0     111 06 06    143 01 01    175 00 0
016 05 05    048 48 EXC   080 49 PRD   112 65 x     144 22 INV   176 54 )
017 00 0     049 03 03    081 06 06    113 01 1     145 44 SUM   177 42 STO
018 42 STO   050 42 STO   082 65 x     114 00 0     146 08 08    178 04 04
019 09 09    051 02 02    083 43 RCL   115 54 )     147 01 1     179 61 GTO
020 05 5     052 43 RCL   084 01 01    116 44 SUM   148 00 0     180 00 00
021 42 STO   053 04 04    085 44 SUM   117 07 07    149 49 PRD   181 20 20
022 00 00    054 32 X:T   086 06 06    118 43 RCL   150 08 08    182 66 PAU
023 22 INV   055 43 RCL   087 54 )     119 07 07    151 43 RCL   183 32 X:T
024 97 DSZ   056 03 03    088 44 SUM   120 75 -     152 08 08    184 43 RCL
025 00 00    057 77 GE    089 07 07    121 43 RCL   153 59 INT   185 09 09
026 00 00    058 00 00    090 43 RCL   122 06 06    154 42 STO   186 91 R/S
027 67 67    059 23 23    091 03 03    123 54 )     155 02 02    187 00 0
028 43 RCL   060 48 EXC   092 65 x     124 69 OP    156 22 INV   188 00 0
029 02 02    061 04 04    093 01 1     125 29 29    157 44 SUM   189 00 0
030 32 X:T   062 42 STO   094 00 0     126 67 EQ    158 08 08    190 00 0
031 43 RCL   063 03 03    095 54 )     127 01 01    159 01 1     191 00 0
```

Kaprekar's Constant, 6174 - (cont)

The program as shown flashes the intermediate N values. To print the N's change the PAU commands at steps 129 and 182 to PRT and change steps 186-187 to PRT R/S. N(0) will not be printed. For the example the printed sequence will be as at the right.

```
6084.
8172.
7443.
3996.
6264.
4176.
6174.
   7.
```

Editor's Note: I tried ten other values for N(0). The highest value for i that I found was six. Is there any N(0) for which i>7 ? Will someone provide a program which permits entry of the N(0) as a single four digit number and which runs in fast mode?

SUPPLEMENTARY BOOK REVIEW - Engineering Statistics With a Programmable Calculator by William Volk (1982, McGraw-Hill). Old-timers in the club may recall that the format and text were favorably reviewed by W. J. Widmer in V7N7/8P15. It now appears that the programs and examples are suspect.

Walter Bodenmuller of Calgary could not run the Regression-2 Program for "Linear Correlations Other than Straight Lines" from Table B.22 on pages 334 and 335 of the book, and asked for help. A quick scan of the listing showed that the commands GTO CLR at steps 189/190 were non-sensical since they made a large segment of the code inaccessible. I jumped to the conclusion that the GTO had merely been substituted for SBR and made a correction, but the program still wouldn't yield correct results. Eventually, I found that steps 189/190 should be SBR CE. Working through the program error was also hindered by a misprint in the table of results for the sample problem accompanying the program. The slope for the linear regression solution was listed as 115.71. It should be 155.71.

Have other readers had difficulty with the programs in this book? Was there an errata sheet which somehow got misplaced in the book that Walter borrowed from a public library? The book was listed for $19.95 in earlier EduCalc catalogs as stock #E-110. It is not listed in the current catalog.

One final comment: The Regression-2 program is simply not a good illustration of using the capability of the TI-59. The sums for the linear regression are implemented in a "brute force" manner rather than using the built-in Σ+ function. Similarly, the calculations of slope, intercept and correlation coefficient are implemented in a "brute force" manner rather than using the built-in Op-12 and Op-13 functions. Even worse, the program requires the user to enter data pairs by first entering the y value, pressing x⇌t, entering the x value, and then pressing A. That sequence is the reverse of the convention used for data pair entry on all TI programmables. The reversed entry sequence, with the y value first, is typical of the HP machines. P. Hanson.

USED HARDWARE FOR SALE - $200 for the complete package.

TI-59 Calculator	Applied Statistics Module
PC-100C Printer	Mathematics Specialty Pakette
80 Magnetic Cards	Fluid Dynamics Specialty Pakette
3 Rolls Printer Paper	3-D Graphics Specialty Pakette
AC Adapter	Printer Utility Specialty Pakette

Will perform the tests in V12N1P4-7 before negotiating a final price. Write to Robert Ericson, 32 Ferncrest Blvd., North Providence, RI 02911 or call (401)-353-5870.

EXTENDED PRECISION SQUARE ROOTS ON THE TI-95 - Hewlett Ladd. Earlier issues have noted that conversion of most programs from the TI-59 for the TI-95 can be easily accomplished by a direct conversion with any absolute addresing of the TI-59 changed to label addressing for the TI-95. Problems will occur, however, when the TI-59 solution relied on a "quirk" which was unique to that machine. An example occurred in a conversion of a many digits of e program due to differences in how the two machines handle the EE function (see V11N4P4).

A recent attempt to convert Robert Prins' 800 digit square root solution from V10N4P23 revealed another difference between the two machines: in the TI-59 a label acts as a true Nop, while in the TI-59 a label will change a "soft" display a "hard" display. Consider the following examples:

1. On both machines the sequence 1 2 3 Nop . 4 5 6 will yield 123.456 in the display.

2. On the TI-59 the sequence 1 2 3 LBL A . 4 5 6 will also yield 123.456 in the display. LBL A acts just like a Nop.

3. On the the TI-95 the sequence 1 2 3 LBL AB . 4 5 6 will yield 0.456 in the display. Thus, the LBL AB "hardened" the display such that the decimal point was recognized as the start of a new input. Page 4-5 of the TI-95 Programming Guide states:

- "You can label any part of a program; the presence of the label does not interfere with program execution or any calculations in progress in the program."

In view of the example that statement is not strictly true.

A condition such as example 3 occurred after conversion of Robert Prins' program. The solution to the problem was to use absolute addressing in that area of the conversion. That is why the listing for the converted program includes the absolute address instruction GTO 0180 at line 0166. A printout of 400 digits of the square root of 3 using the converted program appears at the right.

```
SQUARE ROOT N

          3.

 1732050807.
 5688772935.
 2744634150.
 5872366942.
 8052538103.
 8062805580.
 6979451933.
  169088000.
 3708114618.
 6757248575.
 6756261414.
 1540670302.
 9969945094.
 9989524788.
 1165551209.
 4373648528.
  932319023.
  558206797.
 4820101084.
 6749232650.
 1531234326.
 6903322886.
 6506722546.
 6892183797.
 1227047131.
 6603678615.
 8801904998.
 6537379859.
 3894676503.
 4750657605.
  756618348.
 1296061009.
 4760218719.
  325083145.
 8295239598.
 3299778982.
 4508288714.
 4638329173.
 4722416398.
 1128740615.
```

```
0000 CE `SQUARE ROOT N`
0014 ADV PRT CMS CE⌐1<`
0020 `N<100 ?` BRK ADV
0029 PRT STO 049 CE `# `
0036 `BLOCKS? <41:` BRK
0049 ADV STO 002 10
0055 STO 000 INV LOG
0060 STO 001 1 STO 008
0067 STO 089
0070 LBL 01 90 STO 003
0078 STO 005 50 STO 004
0086 STO 006 RCL 008
0092 STO 007 +/- ST+ 003
0099 ST+ 004
0102 LBL 02 RCL IND 003
0109 INC 003 STO 010
0115 RCL IND 004 INC 004
0122 IF= 010 GTL 02
0128 IF< 010 GTL 05 0
0135 STO 010
0138 LBL 03 INV INC 005
0145 INV INC 006
0149 ST- IND 006
0153 RCL IND 005
0157 ST- IND 006
0161 RCL IND 006 INV
0166 IF< 010 GTO 0180
0172 RCL 001 ST+ IND 006
0179 1. DSZ 007 GTL 03
0187 INC 009 2
0191 LBL 04 ST+ 089
0197 GTL 01
0200 LBL 05 DSZ 000
0206 GTL 06 10 EXC 000
0214 EXC 009 PRT INC 008
0221 DSZ 002 GTL 06 CE
0228 `MORE BLOCKS?` Y/N
0241 GTL 11 GTO 0000
0247 LBL 06 RCL 007
0253 STO 003 10 ST* 009
0261 LBL 07 INV INC 005
0268 ST* IND 005 DSZ 003
0275 GTL 07 x^2
0279 LBL 08 INV INC 006
0286 ST* IND 006 DSZ 007
0293 GTL 08 2 STO 007 50
0302 LBL 09 STO 006
0308 RCL 008 EXC 003 x~t
0315 0 STO 010 x~t
0320 LBL 10 ST- IND 006
0327 INV INC 006 x~t
0332 ST+ IND 006 (
0337 RCL IND 006 /
0342 RCL 001 )( INT *
0349 x~t RCL 001 )
0354 DSZ 003 GTL 10 90
0362 DSZ 007 GTL 09 9
0369 +/- GTL 04
0373 LBL 11 CE `# BLOCK`
0384 `S?` BRK STO 002
0390 GTL 06
```

All TI-95 Firmware Revealed - Robert AH Prins

If you are an oldtime member, these words may seem familiar, and they are. Have a look at V3N10 of 52 Notes [October 1978]. On page 4 of that issue you will get across an article by Steffen Seitz, which carried the title "Some 58/59 Firmware Revealed". In it you could read how to "download" the programs behind the DMS, P-R and statistical functions into normal memory.

This article describes two methods to do something similar on the TI-95. Using these methods, you will be able to find entry points in cartridges and you can even have a look at the internal ROMs of the TI-95. The first method is rather simple, but notwithstanding its simplicity, it allows you to have a look at the contents of cartridges and even at some of the internal ROMs of the TI-95. This last feature may sound very exiting, but you'll find out that it is useless, as it will drop you into a ROM-page with is programmed in assembler.
The other method is a bit more complicated, as it requires a short program in assembler. However, once you have this program, you can download ROMs to RAM, in chunks of 4Kbyte.

Getting into ROM: The easy way

NOTE: To save space I will abbreviate SystemRegister with SR in the rest of this article.

To get into ROM using this method, you must first get into SYSTEM and UNFORMATTED modes. To do so you press,

CONV F5 <BAS> F5 <UNF> FUNC F3 <SYS> F1 <YES>

Next you should recall SR 038 [RCL 2038] and write down its contents. In stead of writing it down, you may also store it in SR 015, provided you don't use any programs from cartridge, as they use this register. Next you will have to change the partitioning, but instead of using the PAR function, we are going to use SR 038.

So enter 00400040FFFF0000 and store it in SR 038 [STO 2038]. As can be read on page C-19 of the TI-95 Programming Guide, SR 038 contains information about the current partitioning of the machine. With the above number we have created an artificial partitioning, with a programmemory starting at address 4000h. If you have a look at page C-17, you will see that this is exactly the address at which the cartridge port starts. So if you have a cartridge installed, you can now press LEARN F1 <1st>, and use -> to single step all the way to address FFFFh. If you do this with the MATH rom, you will find that TI took the liberty of filling almost 8 (EIGHT!) Kb with NOP's. If you do not have any cartridges, you can put 00C000C0FFFF0000 in SR 038, to get directly into the system ROMs.

After you have seen everything you wanted to see, you must restore SR 038 to it's original value, otherwise you will certainly destroy your internal File Space.

Method 2: Downloading ROM to RAM

This method is a bit more elaborate, and to use it you must first create a small program, which I have called ASM. With help of it you can enter codes in memory, without having to enter LEARN mode. (Program on next page)

Once you have created ASM, you can create ROM, but before doing so, you should change the partitioning to P4096,R000,F3104. Next clear program memory and start ASM. The display will show 3000: 00. At this moment you can enter the numbers below to create ROM. To help you a little, the address of the first byte of each group of 16 bytes is placed before them.

Instructions for ASM

1. Enter the program, and store it in File Space.
2. Clear program memory.
3. Start ASM. The display will show, in HEX, the current address and its contents.
4. Enter the new contents, and press NXT. If you don't want to enter a new code, just press NXT, and the old code will remain unaltered.
5. To correct a mistake, press DEL, and re-enter the correct code with NXT.

NOTE: DEL(eting) to an address before the first address displayed after the start of ASM, is possible, but it will bring you into the directory of the File Space, which may cause loss of data! Should it happen, don't panic, just press NXT to get you out of the danger zone.

Listing of ASM

```
0000 HEX ( RCB 20B4 *
0006 100 + RCB 20B5 )
0014 STO 2015 DFN CLR
0019 DFN F1:NXT@01
0026 DFN F2:DEL@02
0033 GTL 03
0036 LBL 01 STB IND 2015
0043 X~T INC 2015
0047 LBL 03 COL 05
0052 MRG 2015 COL 05 ':'
0058 RCB IND 2015 OLD
0063 COL 09 MRG =
0067 COL 09 DEL RTN
0071 LBL 02 INV INC 2015
0078 GTL 03
```

```
3000: 88 0F FF 39 88 30 00 3B 80 06 B8 A3 FC 06 12 3F
3010: 84 06 42 3E 3C 9A 3D 9B 3B D3 3D 79 00 3C D3 3B
3020: 79 00 3A DB 39 E3 EE 42 3C 3E D5 38 D5 39 D5 3A
3030: D5 3B D5 3C B9 82 06 0A
```

After having entered the last 0A, the display should show 3038: 00

Once you have create ROM, you may like to have a look at it. As you might have expected, it looks rather strange, to say the least. However, don't forget that it is assembler!

```
0000   8 ENG LBL 9? '0' NOP ';' HYP PI
0011   LN SBA 612 '?' 4 HYP 'B><' ASM
0022   '=' PAR ';' NOP '=y' NOP '<'
0030   NOP ';y' NOP ':' FIX 9 ST+ EE42
0040   '<>' SF 38 SF 39 SF 3A SF 3B
0050   SF 3C OLD 2 HYP BRK
```

Should you be interested in a commented assembler listing, drop me a line, include two dollar, and I will send you a copy.

Now store this "garbage" in File Space. Once you are ready, you can start using it, following these instructions.

Instructions for ROM:

Command	Press	Display
1. Set the partitioning to P4096,R000,F3104	PAR FIL 3104 ENT REG 0 ENT SET	P4096,R000,F3104
2. Put MEMROM in the alpha register.	ALPHA MEMROM ALPHA	MEMROM
3. Select UNFORMATTED mode	CONV BAS UNF	0000000000000000 DEC HEX OCT 2sC UNF
4. Select SYSTEM mode	FUNC SYS YES	SYSTEM FUNCTIONS STB RCB SBA
5. Enter the number of the ROM page you want to download	00 / 01 / 02 / 03	xx STB RCB SBA
6. Enter the address at which you want to begin downloading	hhhh	xxhhhh STB RCB SBA

Command	Press	Display
7. Download 4096 bytes into RAM (hhhh will be incremented)	SBA 600	xxhhhh0000000000 STB RCB SBA
8 Get into LEARN mode to see the data you have downloaded	LEARN 1st	??? ??? ?? PC= 0 000

I wish you lots of success in decoding the programs. By the way, if you don't have the patience to look for useful sequences yourself, you may enter 01CC00 and press SBA 600 LEARN 1st. You'll be in for a formidable surprise!

[Please don't be offended by the above copyright notice, I only want to prevent commercial use of the material, as I spent large amounts of time to get familiar with the instruction set of the TMS 70C46, the processor used in the TI-95]

TI-95 MEMORY DUMPS - Robert AH Prins and Ruud J Paap

What is a memory dump? In it simplest terms it is the entire contents of the memory of a computer, and it mostly shows up as half a foot of paper after a program has terminated abnormally... However, it can also be a carbon copy of the contents of the memory saved for later use.

It would be nice if the latter of these possebilities could be used with the memory of the TI-95. That way you could not only recover from "MEMORY CLEARED", but it would also give you the opportunity so keep some entirely different TI-95's at your disposal. This article will provide a method to do just that.

Storing the information in the TI-95 is a matter of changing the partitioning, but unfortunately you cannot change used File Space directly into programsteps or dataregisters. However, TI has been so kind to give us an (almost!) complete map of the memory of the TI-95 and of the systemregisters. Using these, and more specifically using systemregister 038, we can change the partitioning without having to resort to the PAR command. This method has the advantage that we can change even used File Space into programsteps or dataregisters, without loosing the original contents!!!

This principle is used in the method described below, and it allows you to store the complete memory of your TI-95 on cassette or in a cartridge. However, take into account the warning on page A-3 of the Programming Guide, because you have to unprotect the system!

The method is fairly simple:
If you consider the entire memory as a string of blocks, as shown below, you can find the File Space at the left, directly followed by programstep 0000. However, register A (000) starts at the far right, and the registers run to the left, that is the highest register follows directly after the last programstep.

File Space	<-Step 0000 Programmemory	Dataregisters Register 000->

Normally, when you have some files, you cannot shift step 0000 to the extreme left - you get the message "FILES IN USE" if you try it -, but by storing the partitioning directly in the appropriate systemregister, you can bypass the PAR command, and thereby create a P7200,R000,F0000 partitioning. Such a partitioning can be stored on cassette or in a cartridge, ready for later reloading! Using this method you can create several completely different types of calculators.

I have used the following steps quite a few times with complete success, so I can assure they are valid:

Command	Press	Display
1. Switch the machine on	ON	TI-95 PROCALC
2. Select SYSTEM MODE	FUNC SYS YES	SYSTEM FUNCTIONS STB RCB SBA
3. Select UNFORMATTED MODE	CONV BAS UNF	0000000000000000 DEC HEX OCT 2sC UNF
4. Recall systemregister 038 and store its contents in normal register 000	RCL 2038 STO 000	30383038183C7D00 (you may see another value)
5. Enter E023E02300400000	E023E02300400000 (E is 2nd STO)	E023E02300400000
6. Put this number in system register 038	STO 2038	E023E02300400000
7. Put 7200 programsteps on tape	I/O TAP WRT	SAVE PGM OR REGS PGM REG ESC
8. Select PGM and enter a name, for example M01 (Machine 01)	PGM M01 ENT	ENTER FILE= M01 FILE M01 WRITTEN

At this point I should advise you to VERIFY the contents of the file, and if you want even more safety, you can repeat step 8 a second time.

At a later time, if you want to restore the old status of your TI-95, you use the following method:

1. Switch the machine on, and set the partitioning to 7200 programsteps	2nd PAR FIL 0 ENT REG 0 ENT PS 7200 ENT SET	P7200,R000,F0000
2. Read the program in from cassette	I/O TAP RD PGM M01 ENT	ENTER FILE= M01 FILE M01 READ
3. Change the partitioning, to allow readout of register 000	2nd PART REG 1 ENT SET	P7192,R001,F0000
4. Put the TI-95 in SYSTEM and UNFORMATTED modes	FUNC SYS YES CONV BAS UNF	1000000000000000 DEC HEX OCT 2sC UNF
5. Store the contents of normal register 000 into system register 038	RCL 0000 STO 2038	30383038183C7D00 DEC HEX OCT 2sC UNF