* TI PPC NOTES *

****

NEWSLETTER OF THE TI PROGRAMMABLE CALCULATOR CLUB. V1N1, 1980.

9213 Lanham Severn Road, LANHAM, MD 20801.

Welcome to the all-new TI Programmable Calculator club and it's newsletter, the TI PPC Notes! Just as our illustrious predecessor, the 52-Notes, we are a non-profit loosely organized programmable calculator club. We are in no way sanctioned by, nor do we receive any help from Texas Instruments Inc. Because our members live all over the US, and hopefully all over the world, it is financially prohibitive to meet at regular intervals. Therefore, the activity centers around a newsletter, which is mailed about once every five weeks. This leaves two weeks of vacation time to your editor.

In these pages we will mainly deal with the TI-59. But, if reader interest justifies it, we will also be concerned with good, old SR-52 and with the TI-57 and the SR-56. And, of course, we will enthusiastically embrace any new model TI might throw our way, provided it is programmable.

The first category we will address can loosely be named "Utilities." It concerns anything that will make programming easier. That category will have the highest priority.

Next on the list come "calculator quirks." Any not published behavior trick of any of the programmables will be discussed here.

We will also bring you good programs. How many per issue will depend on reader participation. But with respect to subject matter, and from experience gleened from other clubs, our own Washington DC area club and the many clubs in Europe, game programs seem to be the all-time favored. That is followed by utility programs, math and plotting. Other categories, such as electronics engineering, finance, real estate, statistics, surveying and many others will be dealt with, if readers specifically request it, and if I receive participation in these categories. I am only an electronics engineer and through the use of the SR-52, and later the TI-59, I learned a little about statistics and math. But I am in no way an expert in all those fields. So I have to rely on the experts. Only through active participation of a large portion of our readership can a magazine like ours be kept alive and thriving.

In order to maintain a high quality of published material, I will have each contributed program refereed by at least two good programmers, who are at the same time familiar with the subject matter. They will not only question the validity of formulas and algorithms used, but they will also try as much as possible to improve upon the programming itself and optimize were indicated. We have seen too many horrible examples in many magazines lately. Thus, we will stop cold such sequences as STO 00 CLR RCL 00 , unless you show us a compelling reason to blaspheme this way.

You can save the club many hours of re-typing and copying if you make your contribution copy-ready and in triplicate.(one for me, one for each of the referees) And if you fold your contribution a couple of times before putting it in a too small envelope, coupled to the use of that cheap blue printing paper, you will manage to bring even our copying machine close to tears.

I will refrain from prescribing a set format for the contributions. Variety is the spice of life! Just make it understandable to your fellow programmers, and be short and to the point. I will not engage in inventing a special cryptic language and I hope you will neither. Normal English doesn't require that much more space.

If you write me (and I love to receive mail) and you expect an answer, please send me a SASE. It will guarantee a reply. Thank you.

Maurice E.T. Swinnen

An interesting application of the HIR registers is the following: when using OP 04 OP 06 type printing, we often cause distortion in the alpha printing in the right-hand margin, because the calculator happens to be either in a special mode, such as EE or ENG, or in FIX n. So, we remedy the situation by inserting INV FIX or INV ENG commands at strategic places, while later in the program returning again to the required mode or FIX n. All this can easily be avoided, at the beginning of the program, we write a separate printing routine as follows:

```
LBL PRT  DIV 12  INV LOG  + 1  )  HIR 08  X:T  OP 06  RTN
```

We now write our programs : ...=(result) x:t PRINT CODE SBR PRT ....

This will work in any mode, EE or ENG, and in any FIX n.
The Snow Bros., Richard and Robert, champions of enhancement and optimization, at seeing this routine, said it could be done shorter. So, their version reads:

```
LBL PRT +  12 INV LOG ) HIR 08  X:T OP 06 RTN
```

Which saves two steps. But, you will say, at noticing the + sign, this routine will not work, because 12 INV LOG = 999 999 999 991 or 9 short of 10 EE 12. That is , by means of the firmware in the TI-59 anyway. But the Snows have an ingenious solution to this: write your print code in the normal manner, THEN ADD 9 TO IT. It works. And nobody will be able to figure out your crazy print code !

There are currently several printer sensing routines in use. The latest one to appear uses the HIR registers: 1 P/R OP 00 HIR 18 *which will return with a zero in the display when the printer is connected, and with a "1" when the calculator is used alone.*
It is easy to set a flag for one of the two results:

CP INV EQ B STF 1 LBL B ...*which will set flag 1 when the printer is connected, not when not. Your program routines can now end as:*

....PRT IFF 1 STO R/S LBL STO ...*which will stop at R/S, to allow copying results, when the calculator is used alone. Otherwise, the result is printed and program execution continues with LBL STO.*

It is well known that neither CLR, nor CMS, nor any other key will clear the HIR registers. But, if you want to clear selectively HIR 8, either from the key board or in a program, this sequence will do it: Either enter a zero or any whole number (integer) and press or write D.MS. If you entered a number that contained a fractional part, such as 12.56, the fractional part will be retained in HIR 8, but it will be MULTIPLIED by 100 when you recall it from HIR 8. This brings up a handy routine to produce the number of the highest register available in a given partition: OP 16 D.MS HIR 18 *will, for example, return with 59 if the calculator was in the turn-on partition.*
You might say " big deal, OP 16 INV INT X 100 = will produce the same result." Granted, but it is both longer and slower.

Somebody has seen fit to knock off two steps in the now famous code converter of Robert Snow. The price paid for this enhancement is two seconds slower execution.

Remember that this routine will convert up to five digits in the display into the required print code when SBR PRT is called.

```
LBL PRT  CP  EQ  016  DIV  LOG  INT  STO 08  OP 28  INV  LOG  +  1  +  LOG  INT  X  100
PRD 09  2  -  INT  SUM  09  =  X  10  DSZ 8  015  CLR  EXC 09  RTN
```

<u>Is converting a certain sequence into a SBR worth the trouble ?</u> In other words, how many programs steps can I expect to save (or to lose)? A little research turned up the following inequalities:

With user-defined keys.................... $M + N + 3 < MN$
With labels.............................$2M + N + 3 < MN$
With direct addresses.......................$3M + N + 1 < MN$

In which M = the number of times a particular sequence appears in a program
N = the number of steps in that particular sequence.

So, if we write a short utility program in which we subtract each time the left side of the inequality from MN, we will actually obtain the number of steps either saved or lost, a positive result indicating savings and a negative one denoting loss. To make the whole thing more attractive we can add a little flashing to the negative result.

In the following program, enter N and press A.
enter M and press B.
For user-defined keys press C.
For common labels press D.
For direct addresses press E.

```
LBL A STO 00 RTN LBL B STO 01 RTN LBL C SBR 068 ( RCL 01 + RCL 00 + 3 ) SBR 076  RTN
LBL D SBR 068  ( RCL 01 X 2 + RCL 00 + 3 ) SBR 076  RTN  LBL E SBR 068 ( RCL 01 X 3 +
RCL 00 + 1 ) SBR 076 RTN RCL 01 X RCL 00 = - RTN = CP INV GE 083 RTN  +  +  RTN
```

As can be observed, the program has been checked with itself, and it was found that is was worth to convert two sequences to subroutines !

To which the Snow Bros. (who else ?) sent the following two versions.
The first one is a highly optimized version of the original one, for calculator use only.

```
LBL A STO 00 STO 02 R/S  STO 01 PRD 02  LBL E  RCL 01  -  2 + LBL D 2 X LBL C CE  RCL 01
+ RCL 00 + 3 - RCL 02  =  +/-  + LNX  0  = RTN
```

The second version is intended for printer use and will produce attractive headings.

```
LBL  A  STO 00  X  R/S  STO 01  -  RCL 00  - 3  -  41361735  OP 04  RCL 01  -  OP 06
27142736  OP 04  RCL 01  - OP 06  13161635  OP 04  RCL 01  + 2 =  OP 06  ADV  R/S
```

In both routines, enter N and press A, enter M and press R/S.
For user-defined keys, press C.
For common labels, press D.
For direct addresses, press E.

(Just looking at the code in the second routine, it seems to be superfluous to press C, D or E. Those labels are not there!)

The second version is a shining example of " during a pending operation, that is as long as you don't write =, you can do almost anything, including loading print code in registers."

The TI PROGRAMMABLE CALCULATOR CLUB is a non-profit loosely organized club of people using programmable calculators produced by Texas Instruments, Inc. The club is by no means affilliated with,nor receives any help from Texas Instruments, Inc. The material contained in the pages of this newsletter, the TI PPC Notes, is not copyrighted and maybe reproduced. We ask, as a matter of courtesy, that the source, TI PPC NOTES, be mentioned. Thank you. The editor is Maurice E.T. Swinnen, 9213 Lanham Severn Road, LANHAM, Maryland, 20801. The use of the material contained in this newsletter is entirely at the user's risk. No reponsibility as to accuracy and the consequences due to lack of it, will be borne by the editor. To defray costs of printing and distribution of this newsletter, a contribution of $ 20.00 per annum in the US and $ 24.00 per annum abroad is solicited.

Data entry through one user-defined key. (TI-59)

Instead of the usual LBL A STO 00 R/S STO 01 R/S ... *if you have more data items to enter than your user-defined keys can stand, one could use the so-called stack method. It is an imitation of the stack used in RPN calculators.*
LBL A PRT EXC 01 EXC 02 EXC 03 EXC 04 EXC 05 RTN *This, of course, can have any length, with any number of registers in any order involved. It only requires that the user remember which data has been entered so far, provided he/she doesn't get interrupted by the phone or one of the kids needing attention. So, the method should have some safeguards built into it, to signal the user when the stack is full.*

```
R/S LBL A PRT EXC 01 EXC 02 EXC 03 EXC 04 EXC 05 IFF 0 STO STF 0 5 STO 00 LBL STO INV DSZ

0 RCL GTO 000 LBL RCL 9 1/X ADV PRT ADV R/S
```

Data are entered as nnn A nnn A ...five times.Attempting to enter a sixth data item will make the printer print .111111 and set flag 0, preventing any further data entering. Only RST or INV STF 0 will undo this state.
Your first datum will end up in Reg 05, while the last entry can be found in Reg 01. Just key in INV LIST to check it out.
A further refinement could consist of having no print-out while the data are entered, but when the stack is full, instead of printing .11111, the program will print your data with suitable descriptors in the margin.

```
LBL RCL 42 02 OP 04  RCL 05 OP 06 42 03 OP 04 RCL 04 OP 06 35 02 OP 04 RCL 03 OP 06  35

03 OP 04 RCL 02 OP 06 14 02 OP 04 RCL 01 OP 06 ADV R/S
```

This routine LBL RCL will replace the simple one on line 11 above : LBL RCL 9 1/x ...
You might also delete the PRT after LBL A.
Entry is the same as in the routine on lines 10 and 11: nnn A nnn A... five times.
Then the printer will automatically produce a list of your entered data with the descriptors in the right-hand margin.

Before sending my "creations" into the wide, wide world, I usually give them to close friends for "comment." Thus, Richard Snow optimized it as follows:

```
LBL B' OP 04 RCL IND 00 OP 06 OP 30 RTN LBL A 5 STO 00 0 RTN STO IND 00 DSZ 0 016 A

42 02 B' 42 03 B' 35 02 B' 35 03 B' 14 02 B' ADV RTN
```

In this routine, initialize with A and enter all data through R/S. The digit following LBL A determines how many data items may be entered.

The stack method is used a lot on the SR-56, because of the absence of IND STO or IND RCL, according to Richard. That is, my simple one EXC 01 EXC 02.... The optimized version contains indirect functions, however.

Just when I finished typing the above, Richard sent me still another version, which does not use OP 06. Initialize with A, then enter your data through R/S. When the stack is full the paper advances. If you attempt to enter more data points, the display flashes.

```
LBL A 5 STO 00 0 R/S  STO IND 00 PRT DSZ 0 005 ADV RTN SBR CE RST
```

When the display flashes and you press R/S again, the error is cleared and the program initialized for re-entry of data.

As you can see from the foregoing,I am trying NOT to adopt a shorthand. The only sign causing me some trouble is the DIVIDE BY sign. Rather than risking to end up with something that looks like a + sign, due to bad printing, I will use DIV at all times. The slash, I feel, is too ambiguous and is used in other keys, such as 1/x, P/R, etc.
The SECOND key is so obvious to anybody who has used the calculator for some time, that I feel it is not necessary anymore to have a special sign for it. Use it where it is implied.
I will also refrain from using the asterisk (*) for the INDIRECT key. I will write IND instead. All this to minimize confusion.

MANUAL PLOTTER-SCALER. *There are lots of automatic plotting programs which will plot f(x), provided you define f(x) somehow in program memory. But what if you have a list of data you would like to plot? You could use OP 07 manually, but you would still be faced with the task of manually scaling your data within the zero to twenty range of OP 07. After having experienced this problem a couple of times at the office, I finally settled on this short routine. It is short enough to be used even by TI-58 fans. The instructions are simple:*

Scan your data and spot the lowest point, enter it and press A.MIN printed.
Do the same for your maximum point, enter it and press B. MAX printed.
Now enter in succession all your data points either through key E or R/S.
If you enter a data point out of bounds, either upper or lower, a small e is printed.

```
LBL E  - RCL 00  = DIV  RCL 02  = CP  INV  GE  083  X:T 20  X:T  GE  094  OP 07  R/S
GTO E  LBL A  STO 00 282429  OP 04  RCL 00  OP 06 RTN  LBL B  STO 01  281344  OP 04
RCL 01  OP 06  -  RCL 00  =  DIV  20  +  1  EE  9  +/-  INV  EE  =  STO 02  OP 00  ADV R/S
GTO  E  54  EE  8  INV  EE  OP 01  GTO  098  54  OP 04  OP 05  OP 00  R/S  GTO  E
```

TWO-VARIABLE GRID-PLOT. *As the name implies, this short routine by Bill Skillman, will plot simultaneously two variables in the range 0 to 19. If out of bounds, a "?" is printed at the appropriate edge.The grid spacing is 5 printing spaces. The symbol for x is the asterisk (*) and for the y he used an "8". A cross-over is indicated with an "x". The instructions are: Write the definition of f(x) in user memory. Place x in the t-register, place y in the display, a call to E' will plot both.*

```
000:  LBL E'  INV STF 4  INV  STF 5  STO 04  2 OP 01 OP 02 OP 03  OP 04  X:T  NOP
SBR  137  EXC 04  SBR 137  OP 22  OP  23  X  49  +  RCL 02  X:T RCL 03  EQ  049  2  =
OP IND 02  RCL 04 X  58  X:T  9  =  EQ  064  NOP  +  2 =  OP IND 03  INV IFF 4  093  1
+  71  X  HIR  15  X:T  1  EE  4  +/-  GE  090  1 EE  6  +/-  =  HIR 35  INV IFF 5  134
1 EE  10  +/-  X:T  HIR  18  GE  123  X  50 =  EQ  123  71  X  X:T  GTO  129  69  EE  12
+/-  +  1  =  HIR 38  OP 05  RTN  INV EE CP  INV GE  174  X:T  20  X:T  GE  180  INT
+/-  +  (  CE  +/-  DIV  5  )  INT  X  EXC  02  STO  03  5  + 4  =  INV  LOG  X^2  RTN
STF  4  CLR  GTO  183  STF 5  3 EXC 02  STO 03  CLR RTN
```

To demonstrate its abilities, I wrote this short sin-cos routine. Start with A.

```
LBL A  RCL 11  SIN +  1 = X 9.9  =  INT  X:T  RCL 11  COS  + 1  =  X 9.9  =  E'  18
SUM 11 GTO  A
```

Bill's routine uses registers 2, 3 and 4 and flags 4 and 5.

I received in the mail a special offer from the Alchemy press, Aptos , California, for the Handcalculator Handbook. They offer this $ 5.95 book for only $ 2.99. I sent away for it and I am delighted. At this price it is by far the best offer I have received in a long time. Although the book was not intended for <u>*programmable*</u> *calculators, all the programs, and there are over 500 in the book, can be keyed into a programmable calculator. The programs work for both RPN and AOS system calculators. On V1N1P6 if have reproduced the entire page as I received it in the mail. Judge for yourself!*

SPECIAL OFFER

$2.99 **$2.99**

The HANDCALCULATOR HANDBOOK

by T.R. DICKSON

364 Pages 25 Chapters Paperback

This Handbook is a complete calculator reference book. A reference containing a wealth of algorithms for mathematical applications of typical scientific calculators. Nearly 500 program examples illustrate a wide variety of calculator uses. Programs are given in RPN (Reverse Polish Notation), AOS (Algebraic Operating System) and Algebraic number processing illustrating the use of any typical calculator. Each algorithm is explained and described in a flow chart type of notation. This notation allows the direct and specific implementation of the algorithm on a calculator. Furthermore this flow chart notation can be used as a direct aid in the computer implementations of the algorithms.

Table of Contents by Chapters

Table of Programs ● History of the Handcalculator ● Glossary of Calculator Terms Calculator Functions and Operators ● Types of Calculators ● Computations on the Calculator Conversion Constants ● Programming for a Calculator ● Function Evaluation ● Tabulations Logarithms ● Powers and Roots of Numbers ● Roots of Functions ● Vector and Matrix Operations Solving Linear Simultaneous Equations ● Statistics ● Curve Fitting ● Numerical Integration Trignometric Functions and Triangles ● Conversions ● Complex Arithmetic ● Surveying Navigation ● Business and Finance ● Miscellaneous Applications ● References

Usual list price is $5.95 but the Handbook can be yours for only $2.99 by direct mail.

ALCHEMY PRESS BOX 1057 APTOS, CALIFORNIA 95003

Enclosed please find a check or money order in the amount of $____________

for ____________ copies of the HANDCALCULATOR HANDBOOK.

Send to:

Name ______________________________

Address ______________________________

City ______________________________

State ____________________ **Zip** ____________________

Price includes tax, handling and mailing charges. Add 50¢ per book if UPS shipping is desired.

Money back guarantee if not satisfied.

Alchemy Press 512 Monterey Dr. Aptos, California 95003

PLEASE POST, DUPLICATE or CIRCULATE THIS NOTICE

MASTER LIBRARY PRINTING ROUTINES. *The printing provision in the Master Library module has been variously described as..... well, all the descriptions I have heard or seen in print were all rather derogatory and I don't care to repeat them. During my visit to Lubbock, in the summer of 1979 I was unable to find out which of the applications programmers was responsible for it. Instead it was pointed out how well programmed the newer modules are. Thus, I met Don O'Grady and Garry Morella, who did the EE, the M/U and the RPN modules. All my respects. Those modules are tops. This not only according to my opinion, but I still have to meet a good programmer who knocks them. Also during the summer of 1979 I got into contact with Valentino Ducati, a top programmer who lives in Wintertur, Switzerland. Among the many good routines and programs I received from him I found a pakette of programs to access the PGMs in the Master Library. Valentino has kept each routine under 240 steps long, such that two routines fit on one mag card. You may now throw away the useless plastic cards and use real mag cards instead. Valentino told me he hadn't developed routines for ML-24 and ML-25 yet, but he sent me routines for all the other ones. So, I quickly wrote the missing ones, according to Val's recipe. Those two are reproduced on P8 and P9. If there is enough reader interest, I'll ask Valentino's permission to publish the rest of them. The instructions are extremely simple:*
Read in the card side necessary for a particular program.
Then, instead of doing the first step in the manual, which ususally says SELECT PROGRAM, go to step 2. Use absolutely the same keys as in the manual. The program in user memory will access the correct program and correct SBRs.
Inputs and outputs will now be printed with suitable descriptors in the right-hand margin.
In both routines, you will probably recognize LBL PRT, at the start, as the interesting HIR application discussed on P1.

One of the main advantages of these routines is, that you may put your calculator in any mode (EE, ENG or NORMAL) and any FIX n, without fear of distortion in the descriptors.

TI-59 Firmware. *In V3N10P4 of 52-Notes, Steffen Seitz showed us a routine to reveal the TI-59 firmware. His sequence was:*
OP 09 PGM AB R/S R/S R/S OP 17 GTO FGH OP 17 R/S P/R LRN
In which AB is any of the ML-module programs: 02, 03, 04, 08, 11, 12, 13, 18, 24 or 25.
And in which FGH any step 000 to 575 or 800 to 959.
In a subsequent issue, V3N12P5 of 52-Notes, Bill Skillman identified the constants.
Then, reading GESPRO, a new German-language newsletter, I saw a contribution by one Walter Ulrich. His sequence OP 09 PGM 24 R/S R/S R/S OP 17 GTO 959 OP 17 R/S P/R LRN *produces in the display 959 55. If you now SST you can advance up to 1374 00 ! The author wonders aloud what all this means. Obviously, he had never heard of Steffen Seitz's discovery. If you try to print Walter Ulrich's sequence, you discover that it is still Steffen's software, now at (apparently) different program counter steps. The printer even refuses to print anything above 999 and reverts there to zero.*
Dick Blayney has compilated a nice summary of all this, unfortunately too long to fit in this newsletter. So, everybody might experiment on his own and signal anything unusual.
Dick also notes that, contrary to V3N12P5 of 52-Notes, (Steve Bepko could get a listing with mnemonics up to step 503 only) it is possible to step 583, with the PC100A. Dick says to use the ususal GTO 479 9 OP 17 PGM 12 SBR 444 R/S P/R LRN, than manually SST to step 505, followed by LRN LIST, at which the listing is produced up to step 583 with mnemonics. Only step 576 is missing, which was the same in the listing without mnemonics. Any more ideas ?

Dick Blayney seems to enjoy delving into the inner workings of things and making sense of seemingly too complicated issues. I received a 24-page book he wrote with the title:" THE COMPLETE ANALYSIS OF SELF-PROGRAMMING, PROGRAM REGISTER TRANSFERS AND FRACTURED DISPLAYS FOR THE SR-52." And, although I will have to re-read some of it a couple of times, much of the mystery has been cleared up in my mind. Dick is willing to make it available to our readers at $ 3.00 US, to cover the cost of copying and mailing. Dick's address is Richard D. Blayney, 26632 Guadiana, Mission Viejo, CA 92691. A real bargain to an avid 52 fan!

Special access routine.

```
000   76  LBL
001   99  PRT
002   55   ÷
003   01   1
004   02   2
005   22  INV
006   28  LOG
007   85   +
008   01   1
009   54   )
010   82  HIR
011   08   08
012   32  X:T
013   69  OP
014   06   06
015   92  RTN
016   76  LBL
017   11   A
018   32  X:T
019   02   2
020   04   4
021   02   2
022   09   9
023   71  SBR
024   99  PRT
025   36  PGM
026   24   24
027   11   A
028   32  X:T
029   01   1
030   05   5
031   02   2
032   08   8
033   71  SBR
034   99  PRT
035   98  ADV
036   91  R/S
037   76  LBL
038   16  A'
039   32  X:T
040   01   1
041   05   5
042   02   2
043   08   8
044   71  SBR
045   99  PRT
046   36  PGM
047   24   24
048   16  A'
049   32  X:T
050   02   2
051   04   4
052   02   2
053   09   9
054   71  SBR
055   99  PRT
056   98  ADV
057   91  R/S
058   76  LBL
059   12   B
060   32  X:T
061   02   2
062   01   1
063   03   3
064   07   7
065   71  SBR
066   99  PRT
067   36  PGM
068   24   24
069   12   B
070   32  X:T
071   03   3
072   08   8
073   04   4
074   00   0
075   71  SBR
076   99  PRT
077   98  ADV
078   91  R/S
079   76  LBL
080   17  B'
081   32  X:T
082   03   3
083   08   8
084   04   4
085   00   0
086   71  SBR
087   99  PRT
088   36  PGM
089   24   24
090   17  B'
091   32  X:T
092   02   2
093   01   1
094   03   3
095   07   7
096   71  SBR
097   99  PRT
098   98  ADV
099   91  R/S
100   76  LBL
101   13   C
102   32  X:T
103   04   4
104   05   5
105   01   1
106   06   6
107   71  SBR
108   99  PRT
109   36  PGM
110   24   24
111   13   C
112   32  X:T
113   03   3
114   08   8
115   04   4
116   00   0
117   71  SBR
118   99  PRT
119   98  ADV
120   91  R/S
121   76  LBL
122   18  C'
123   32  X:T
124   03   3
125   08   8
126   04   4
127   00   0
128   71  SBR
129   99  PRT
130   36  PGM
131   24   24
132   18  C'
133   32  X:T
134   04   4
135   05   5
136   01   1
137   06   6
138   71  SBR
139   99  PRT
140   98  ADV
141   91  R/S
142   76  LBL
143   14   D
144   32  X:T
145   03   3
146   08   8
147   02   2
148   04   4
149   71  SBR
150   99  PRT
151   36  PGM
152   24   24
153   14   D
154   32  X:T
155   02   2
156   06   6
157   02   2
158   08   8
159   71  SBR
160   99  PRT
161   98  ADV
162   91  R/S
163   76  LBL
164   19  D'
165   32  X:T
166   02   2
167   06   6
168   02   2
169   08   8
170   71  SBR
171   99  PRT
172   36  PGM
173   24   24
174   19  D'
175   32  X:T
176   03   3
177   08   8
178   02   2
179   04   4
180   71  SBR
181   99  PRT
182   98  ADV
183   91  R/S
184   76  LBL
185   15   E
186   32  X:T
187   03   3
188   01   1
189   03   3
190   08   8
191   02   2
192   04   4
193   71  SBR
194   99  PRT
195   36  PGM
196   24   24
197   15   E
198   32  X:T
199   03   3
200   08   8
201   02   2
202   04   4
203   04   4
204   00   0
205   71  SBR
206   99  PRT
207   98  ADV
208   91  R/S
209   76  LBL
210   10  E'
211   32  X:T
212   03   3
213   08   8
214   02   2
215   04   4
216   04   4
217   00   0
218   71  SBR
219   99  PRT
220   36  PGM
221   24   24
222   10  E'
223   32  X:T
224   03   3
225   01   1
226   03   3
227   08   8
228   02   2
229   04   4
230   71  SBR
231   99  PRT
232   98  ADV
233   91  R/S
234   00   0
235   00   0
236   00   0
237   00   0
238   00   0
239   00   0
```

Special access routine.

```
000  76 LBL
001  99 PRT
002  55  ÷
003  01  1
004  02  2
005  22 INV
006  28 LOG
007  85  +
008  01  1
009  54  )
010  82 HIR
011  08  08
012  32 X:T
013  69 OP
014  06  06
015  92 RTN
016  76 LBL
017  11  A
018  32 X:T
019  06  6
020  05  5
021  02  2
022  01  1
023  71 SBR
024  99 PRT
025  36 PGM
026  25  25
027  11  A
028  32 X:T
029  06  6
030  05  5
031  01  1
032  05  5
033  71 SBR
034  99 PRT
035  98 ADV
036  91 R/S
037  76 LBL
038  16 A'
039  32 X:T

040  06  6
041  05  5
042  01  1
043  05  5
044  71 SBR
045  99 PRT
046  36 PGM
047  25  25
048  16 A'
049  32 X:T
050  06  6
051  05  5
052  02  2
053  01  1
054  71 SBR
055  99 PRT
056  98 ADV
057  91 R/S
058  76 LBL
059  12  B
060  71 SBR
061  42 STD
062  36 PGM
063  25  25
064  12  B
065  71 SBR
066  43 RCL
067  98 ADV
068  91 R/S
069  76 LBL
070  17 B'
071  71 SBR
072  43 RCL
073  36 PGM
074  25  25
075  17 B'
076  71 SBR
077  42 STD
078  98 ADV
079  91 R/S

080  76 LBL
081  42 STD
082  32 X:T
083  02  2
084  01  1
085  02  2
086  07  7
087  03  3
088  02  2
089  04  4
090  06  6
091  71 SBR
092  99 PRT
093  92 RTN
094  76 LBL
095  43 RCL
096  32 X:T
097  02  2
098  07  7
099  04  4
100  00  0
101  00  0
102  00  0
103  71 SBR
104  99 PRT
105  92 RTN
106  76 LBL
107  13  C
108  71 SBR
109  52 EE
110  36 PGM
111  25  25
112  13  C
113  71 SBR
114  44 SUM
115  98 ADV
116  91 R/S
117  76 LBL
118  18 C'
119  71 SBR

120  44 SUM
121  36 PGM
122  25  25
123  18 C'
124  71 SBR
125  52 EE
126  98 ADV
127  91 R/S
128  76 LBL
129  52 EE
130  32 X:T
131  02  2
132  02  2
133  01  1
134  03  3
135  02  2
136  07  7
137  71 SBR
138  99 PRT
139  92 RTN
140  76 LBL
141  44 SUM
142  32 X:T
143  02  2
144  07  7
145  04  4
146  00  0
147  00  0
148  00  0
149  71 SBR
150  99 PRT
151  92 RTN
152  76 LBL
153  14  D
154  32 X:T
155  03  3
156  02  2
157  04  4
158  06  6
159  71 SBR

160  99 PRT
161  36 PGM
162  25  25
163  14  D
164  32 X:T
165  02  2
166  02  2
167  04  4
168  00  0
169  71 SBR
170  99 PRT
171  98 ADV
172  91 R/S
173  76 LBL
174  19 D'
175  32 X:T
176  02  2
177  02  2
178  04  4
179  00  0
180  71 SBR
181  99 PRT
182  36 PGM
183  25  25
184  19 D'
185  32 X:T
186  03  3
187  02  2
188  04  4
189  06  6
190  71 SBR
191  99 PRT
192  98 ADV
193  91 R/S
194  76 LBL
195  15  E
196  71 SBR
197  22 INV
198  36 PGM
199  25  25

200  15  E
201  71 SBR
202  24 CE
203  98 ADV
204  91 R/S
205  76 LBL
206  10 E'
207  71 SBR
208  24 CE
209  36 PGM
210  25  25
211  10 E'
212  71 SBR
213  22 INV
214  98 ADV
215  91 R/S
216  76 LBL
217  22 INV
218  32 X:T
219  02  2
220  07  7
221  01  1
222  04  4
223  03  3
224  06  6
225  71 SBR
226  99 PRT
227  92 RTN
228  76 LBL
229  24 CE
230  32 X:T
231  02  2
232  06  6
233  02  2
234  02  2
235  04  4
236  00  0
237  71 SBR
238  99 PRT
239  92 RTN
```

TRACE? Don Laughery discovered a curious "new state." If, from the key board, you key in the sequence PGM 10 SBR 500 R/S CLR LRN LRN LRN the calculator seems to be in TRACE mode. PGM 10 can be 10 through 17. SBR 500 can be any high number. CLR can also be CE, bu' then the calculator stays in EE mode, otherwise the mode is NORMAL. As you guessed, the ML module has to be in place for all this. Resetting flag 9, nor pressing the TRACE key on the printer, will influence this state. But pressing RST or CP will. Some unknown flag? Or maybe the module itself?
When you go into LRN mode, the only unusal thing you will see is an x:t step at 000. But if you attempt to key in a program, the calculator insists on reducing all the codes by 10. Thus, LBL becomes PAU, etc. If you take this into account and key in a code 10 higher than the normal one, your keyed-in program will run. If it contains an RST, however, it will reset this "new state." To access your program in user memory you will have to treat it as PGM 00 and call it as such, because you seem to still have access to PGM 10. Don't press RST to get rid of the access to PGM 10, because it will also "kill" the "new state." Anybody has any ideas what is happening?

PRINTING DOTS CLEANING ROUTINE. TI recommends to clean the printing dots every time you put in a fresh roll of paper. To this end, you insert the strip of bond paper that comes with the rolls of paper and run the program from page VI-12 of the manual. The idea is, to excercise all the dots, 100 in total. But, one should not only excercise all dots, but also produce the maximum possible heat in them, to burn off the tiny shreds of paper sticking to them. Thus, the OP 05 loop should be as short as possible. What could be shorter than OP 05 RST, reasoned Don Laughery. He proposes the following routine:

OP 05 RST LBL A 11 11 11 11 11 OP 01 OP 02 OP 03 OP 04 RST Press A to start.

This could be even shorter, although not as easy to key in:

OP 05 RST LBL A 9 1/X HIR 05 HIR 06 HIR 07 HIR 08 RST also started by pressing A.

I thought I had seen it all! Not necessarily so. Part of my job description at the Walter Reed Army Institute of Research reads, in typical governmentese, as follows: "Incumbent scrutinizes existent and current literature for articles with medical, biological and medico-biological engineering relevance and transmits same to qualified personnel." In normal English, that means that I am required to read all kinds of "strange" magazines and journals. Thus, I recently saw in "The Journal of Extra-corporeal Technology" (I kid you not, this one is for real!) an article called" Programmable Calculators and Cardiopulmonary Bypass." The three authors, two medical doctors and one registered nurse, have developed several programs for the TI-59/PC100A, to be used right in the operating room! One program, nicely interactive, computes the required amount of heparin, a blood thinner, needed by the patient during bypass surgery. Another program calculates protamine reversal, still another one body surface and from it the needed blood pump volume during surgery. Then there are several programs to be used in post-op, most of them to compute the correct concentration of drugs for intravenous administration, such as dopamine, epinephrine, lidocaine and the antihypertensive drug nitroprusside. As one of the lab technicians said the other day:" Far out, man!"

This newsletter wouldn't be complete without a game program. This one concerns the riddle how to bring a man, a fox, a chicken and a sack of corn across the river. In Europe a simular one concerns a farmer, a wolf, a goat and a cabbage.The first program I saw appeared in Display, written in German by R. Loehler. After I translated it for our local club, we soon had several versions, each one more optimized than the former one. The one by Richard Snow stood out of the pack, however, because it was the first one to have a "real river." The one on page P11 is the latest version.(by Bill Skillman, after he saw Richard's version) In my opinion, it is an ideal game program, in that it requires the user to press just one single key. All instructions are subsequently printed. So, just press E and sit back.

PROGRAM DERIVED (REMOTELY!) FROM "HOL' UEBER" BY R.LOEHLER (IN DISPLAY), MODIFIED BY RICH. SNOW

```
000  42 STO
001  05  05
002  73 RC*
003  05  05
004  32 X:T
005  43 RCL
006  01  01
007  22 INV
008  67  EQ
009  01  01
010  55  55 → (155)
011  55  ÷
012  04  4
013  95  =
014  35 1/X
015  72 ST*
016  05  05
017  61 GTO
018  00  00
019  79  79
020  76 LBL
021  18 C'
022  42 STO
023  05  05
024  76 LBL
025  16 A'
026  04  4
027  93  .
028  76 LBL
029  17 B'
030  03  3
031  59 INT
032  42 STO
033  00  00
034  73 RC*
035  05  05
036  69 OP
037  35  35
038  84 OP*
039  00  00
040  97 DSZ
041  00  00
042  00  00
043  34  34
044  69 OP
045  05  05
046  69 OP
047  00  00
048  92 RTN
049  76 LBL
050  15  E   START
051  08  8
052  69 OP
053  17  17
054  07  7
055  09  9
056  18 C'
057  98 ADV
058  16 A'
059  16 A'
```

MOVE

PRINT A LINE

```
060  16 A'
061  16 A'
062  16 A'
063  16 A'
064  98 ADV
065  17 B'
066  17 B'
067  17 B'
068  17 B'
069  17 B'
070  71 SBR
071  01  01
072  60  60 → (160)
073  42 STO
074  04  04
075  42 STO
076  03  03
077  42 STO
078  02  02
079  42 STO
080  01  01
081  43 RCL
082  21  21
083  84 OP*
084  01  01
085  69 OP
086  05  05
087  00  0
088  84 OP*
089  01  01
090  43 RCL
091  38  38
092  84 OP*
093  02  02
094  69 OP
095  05  05
096  00  0
097  84 OP*
098  02  02
099  43 RCL
100  44  44
101  84 OP*
102  03  03
103  69 OP
104  05  05
105  00  0
106  84 OP*
107  03  03
108  43 RCL
109  41  41
110  84 OP*
111  04  04
112  69 OP
113  05  05
114  00  0
115  84 OP*
116  04  04
117  69 OP
118  05  05
119  43 RCL
```

PRINT RIVER AND POSITIONS

```
120  03  03
121  32 X:T
122  43 RCL
123  01  01
124  67  EQ
125  01  01
126  70  70
127  43 RCL
128  02  02
129  67  EQ
130  01  01
131  94  94
132  43 RCL
133  04  04
134  67  EQ
135  02  02
136  01  01
137  00  0
138  91 R/S
139  76 LBL
140  11  A    MAN
141  01  1
142  81 RST
143  76 LBL
144  12  B    HEN
145  03  3
146  81 RST
147  76 LBL
148  13  C    CORN
149  04  4
150  81 RST
151  76 LBL
152  14  D    FOX
153  02  2
154  81 RST
155  02  2
156  00  0
157  42 STO
158  05  05
159  17 B'
160  43 RCL
161  24  24
162  69 OP
163  03  03
164  43 RCL
165  23  23
166  69 OP
167  02  02
168  01  1
169  92 RTN
170  01  1
171  67  EQ
172  01  01
173  37  37
174  43 RCL
175  02  02
176  22 INV
177  67  EQ
178  01  01
179  37  37
```

WRONG ENTRY

```
180  43 RCL
181  04  04
182  22 INV
183  67  EQ
184  01  01
185  37  37
186  01  1
187  07  7
188  18 C'
189  02  2
190  91 R/S
191  61 GTO
192  00  00
193  70  70   (70)
194  03  3
195  06  6
196  18 C'
197  16 A'
198  61 GTO
199  02  02
200  07  07
201  03  3
202  06  6
203  18 C'
204  02  2
205  08  8
206  18 C'
207  69 OP
208  69  69
209  98 ADV
210  61 GTO
211  00  00
212  70  70   (70)
```

HAPPY ENDING

FOX EATS HEN

HEN EATS CORN

LABELS

```
021  18 C'
025  16 A'
029  17 B'
049  15  E
140  11  A
144  12  B
148  13  C
152  14  D
```

REGISTERS

0 COUNTER
1 MAN LOCATION
2 FOX "
3 HEN "
4 CORN "
5 POINTER

14-79 PRESTORED

```
***RIVER CROSSING***

CAN YOU FIGURE HOW
TO GET A MAN AND A
FOX, HEN AND A SACK
OF CORN ACROSS A RI-
VER IN A BOAT HOLD-
ING MAN + ONE OTHER?

 TO MOVE   PRESS
MAN ALONE    A
MAN + HEN    B
MAN +CORN    C
MAN + FOX    D
   MAN:((((((((:
  FOX :((((((((:
  HEN :((((((((:
 CORN :((((((((:
      :((((((((:
      :((((((((:  MAN
  FOX :((((((((:
      :((((((((: HEN
 CORN :((((((((:
      :((((((((:
   MAN:((((((((:
  FOX :((((((((:
      :((((((((: HEN
 CORN :((((((((:
      :((((((((:
      :((((((((:  MAN
      :((((((((: FOX
      :((((((((: HEN
 CORN :((((((((:
      :((((((((:
   MAN:((((((((:
      :((((((((: FOX
  HEN :((((((((:
 CORN :((((((((:
      :((((((((:
      :((((((((:  MAN
      :((((((((: FOX
  HEN :((((((((:
      :((((((((:CORN
      :((((((((:
   MAN:((((((((:
      :((((((((: FOX
  HEN :((((((((:
      :((((((((:CORN
      :((((((((:
      :((((((((:  MAN
      :((((((((: FOX
      :((((((((: HEN
      :((((((((:CORN
      :((((((((:
```

```
CONGRATS, YOU DID IT
   MAN:((((((((:
  FOX :((((((((:
  HEN :((((((((:
 CORN :((((((((:
      :((((((((:
      :((((((((:  MAN
      :((((((((: FOX
  HEN :((((((((:
 CORN :((((((((:
      :((((((((:
****YOU BLEW IT!!***
THE HEN ATE THE CORN

   MAN:((((((((:
  FOX :((((((((:
  HEN :((((((((:
 CORN :((((((((:
      :((((((((:
      :((((((((:  MAN
  FOX :((((((((:
  HEN :((((((((:
      :((((((((:CORN
      :((((((((:
****YOU BLEW IT!!***
THE FOX ATE THE  HEN

   MAN:((((((((:
  FOX :((((((((:
  HEN :((((((((:
 CORN :((((((((:
      :((((((((:

ALPHA REG LIST:

  NUMERIC   REG ALPHA
1532312235.  14 CONGR
1337365700.  15 ATS,
4532410016.  16 YOU D
2416002437.  17 ID IT
4335323122.  18 WRONG
  17313735.  19  ENTR
4500737373.  20 Y !!!
  30133100.  21  MAN
        0.   22
6255555555.  23 :((((
5555555562.  24 ((((:
3723170023.  25 THE H
1731001337.  26 EN AT
1700372317.  27 E THE
  15323531.  28  CORN
3723170021.  29 THE F
3244001337.  30 OX AT
1700372317.  31 E THE
    231731.  32   HEN
5151515145.  33 ****Y
3241001427.  34 OU BL
1743002437.  35 EW IT
7373515151.  36 !!***
3013310047.  37 MAN +
  21324400.  38  FOX
    160000.  39   D
3013310047.  40 MAN +
1532353100.  41 CORN
    150000.  42   C
3013310047.  43 MAN +
  23173100.  44  HEN
    140000.  45   B
3013310013.  46 MAN A
2732311700.  47 LONE
    130000.  48   A
  37320030.  49  TO M
3242170000.  50 OVE
3335173636.  51 PRESS
2431220030.  52 ING M
1331004700.  53 AN +
3231170032.  54 ONE O
3723173571.  55 THER?
4217350024.  56 VER I
3100130014.  57 N A B
3213370023.  58 OAT H
3227162000.  59 OLD-
3221001532.  60 OF CO
3531001315.  61 RN AC
3532363600.  62 ROSS
1300352420.  63 A RI-
2132445700.  64 FOX,
2317310013.  65 HEN A
3116001300.  66 ND A
3613152600.  67 SACK
3732002217.  68 TO GE
3700130030.  69 T A M
1331001331.  70 AN AN
1600130000.  71 D A
1513310045.  72 CAN Y
3241002124.  73 OU FI
2241351700.  74 GURE
2332430000.  75 HOW
5151513524.  76 ***RI
4217350015.  77 VER C
3532363624.  78 ROSSI
3122515151.  79 NG***
```

USER INSTRUCTIONS

1. READ CARDS, BANKS 1-4
2. PRESS E, FOLLOW INSTRUCTIONS

(REQUIRES PC-100)

NOTE: Key in the program and load the registers with 8 OP 17. Then repartition to 6 OP 17 and record 4 card sides.

NEWCOMER'S CORNER.- Although some of my friends adviced me to call this the beginner's corner, I hesitate to use that word. Beginner, in my mind, conjures "rank amateur." Most of our members have had the calculator now for some time and are fairly familiar with its operation. But they might be newcomers to the world of calculator fanatics, which most of the old 52-Notes members are. And they might not know all the tricks and quirks we all learned during the last couple of years. In this corner then, we will try to rehash those tricks and discoveries that have proven to be the most useful. This way we hope to bring the newcomers up to par in a rather short time. Once this is done we will abandon this part of the newsletter and use the space for some other useful purpose.

THE HIERARCHIAL INTERNAL REGISTERS.- Besides the 100 data registers, the TI-59 has eight registers normally used to temporarily store operands during execution of arithmetic. Each time a calculation requires opening up one more set of parenthesis, one more HIR register is called upon to perform. Normally, those registers are not accessible from the keyboard. TI did not mention them in the manual. I first read about them in Display whose editor, Heinrich Schnepf, reported on them and on their respective functions in June 1977, just one month after the TI-59 arrived on the market.

As I said, there are eight HIRs, 1 to and including register 8. All the normal register arithmetic, such as store, recall, product, sum, and the inverses of the last two, can be performed on them. This can only be done under program control. That is, we have to write the necessary commands into a program. The two-digit code that represents HIR is 82. This we can easily synthesize by keying STO 82, followed by a deletion of STO. So, key STO 82 BST BST DEL SST. Granted, it is a little cumbersome, but, as you will soon realize, the HIRs allow neat tricks to be performed. Next we have to follow the HIR command by a two-digit code that will tell the calculator 1) which function is required and 2) on what register to execute it. The first digit then means: 0 = STO, 1 = RCL, 3 = SUM, 4 = PRD, 5 = INV SUM, and 6, 7, 8 and 9 all mean INV PROD.

Thus, if you want to write "store into HIR 8" you write 82 08, which will be listed as HIR 08. If you now want to recall from that same register you write 82 18 and it will be listed as HIR 18.

So, to recapitulate, if you want to write HIR 35, meaning SUM into HIR 5, you key: STO 82 BST BST DEL SST STO 35 BST BST DEL SST.

Some of the HIR registers are shared by other internal functions of the calculator, which, because of its rather complex rules, was the reason TI did not mention their existence. But that did not deter the real fanatics to learn how to use the HIRs with great success. For example HIR 5 to HIR 8 are identical to the print.code registers OP 01 through OP 04. In a later article we will go deeper into the rules that govern those special uses. For now, it suffices that, if you see in a program HIR 45, for example, you know what it means and how you can key it in.

DSZ ON REGISTERS HIGHER THAN 9.- The manual says that DSZ will work only on registers 0 through 9. But soon after the calculator was on the market, a lot of people announced that they had found a way to DSZ all the registers. It is done the same way we synthesized the HIRs. Suppose you want to DSZ register 35. So, you key DSZ STO 35 BST BST DEL SST. That will leave the code 97 35 in program memory. As you can see, it is not possible to DSZ a register higher than 9 FROM THE KEYBOARD, but is is possible to do so under program control.If that DSZ 35 is followed by a direct address, such as 255, you key GTO 255 and go back to delete the GTO. The end result will be 97 35 02 55.

PROGRAMMED PARTITIONING.- Although the manual advices to mark the correct partitioning on each recorded magnetic card, that should hardly be necessary. You should try to write all your programs, such that they contain within the first 160 program steps, a sequence that tells the calculator to re-partition automatcially. For example, you are using a total of 712 steps and 27 data registers. Obviously, the partitioning should be 719.29. So, your initialization routine, usually Label E, should contain: LBL E 3 OP 17 ..... To key in your program you press from the keyboard 3 OP 17, go into LRN and enter all the needed program steps and load the required registers with data.Then you go out of LRN, press 6 OP 17 and record the two mag cards, all four sides in that partitioning. No need to write the partitioning on the cards: It will always be the turn-on one. And it reduces the amount of button pushing,as an added bonus.

I hope you enjoyed this first issue of our newsletter. See you next month. Maurice