LRN

* TI PPC NOTES * V6N8, 1981

9213 Lanham Severn Road, Lanham MD 20706
or P.O. Box 710, Lanham MD 20706

Newsletter of
the TI Programmable Calculator Club.

Recently, in v6n2p4, US Patent # 4,153,937, I hinted at possible hidden gold in the TI-59. John Worthington has found some of it. John has a 59 and found at least two 58's at the office that list key code 21 as "HI*" (yes, that is HIR IND) and keycode 26 as "X!". (yes, X-factorial) On run-of-the-mill calculators both list "2ND".

So, put in user memory the following short test program: LBL A STO 00 CLR X:T "21" 0 RTN LBL B "26" RTN (Remember to put in "21" as STO 21 and subsequently to delete STO) User instructions: 1. Enter the number to be stored (or summed, subtracted or producted, or divided) into the HIR and press X:T. 2. Specify the HIR operation (e.g. to product HIR 8, write 48) and press A.

To use the X! routine, key in any integer and press B. For example, enter 5 and press B. If the display says 120, you got a winner. Otherwise, forget it. If you have a winner, though, please drop me a line and tell me what the manufacturing date is, such as ATA.... I will publish the results of this survey.

As treatened in v6n6/7, I went on vacation to see the little village in Belgium I was born in some fifty-five years ago. I met all the friends from TI-SOFT, the fantastic sister publication in Kapellen, Belgium: the dynamic Thomas Coppens and his friends Jean Verswijvelen, Annie Debaere and A. Broeckx. Because PPX in Lubbock has not provided them with any programs as yet (save the ones sent by me directly) TI-SOFT keeps an extensive library of programs at hand for its European members. I brought with me a selection of their best ones. Copies of them have been sent already to reviewers in the States and we will publish as many as we can.

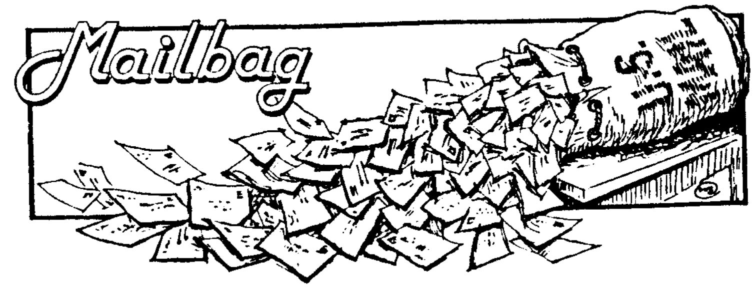
I also met Lars Hedlund of our Swedish sister publication PROGRAMBITEN. I agreed with him that I will publish as soon as possible a listing and instructions for all the routines that supposedly will go into our own module. This to give members a chance to comment on them. However, writing the newsletter itself eats up a large chunk of my free time. I would like to delegate the authority on the module to somebody (or to some group) preferrrably here in the States.(I am not prejudiced with repect to foreign members. Coordination is easier and mail goes faster here in the States). Please write me if you or your group feels they can handle this job. I would be very grateful.

Tom Coppens brought to my attention a fantastic book published in Belgium and written in English, Astronomical Formulae for Calculators, by Jean Meeus. As I saw the enormous interest this book would attract among our US members I bought one hundred of them and Tom arranged to send them over by surface mail. The full description of this book can be found on page 11 of this issue. Because our dollar made a rather strong showing in Europe, I was able to buy those books at a favorable price. You may send for your copy at $ 10.00 each. Please add $ 2.00 for first class mailing costs.

Several members have complained about the keys on their calculator. Some of them miss often, while others require herculean force to make them perform. Others still are so sensitive that merely touching them produces a string of identical digits across the display. The simple solution is, of course, to exchange the calculator. But that costs you $ 53.00 each time. Does anyone know how to disassemble and clean those keys?

Maurice E.T. Swinnen

IN THIS ISSUE:



Dear Maurice,

I was sorry to hear about your Annie passing away in June.

T.J. & 718 other members.

I was really touched by your outpouring of sympathy. I am unable to thank each of you individually. But let this short note serve as such. It feels good to have so many kind friends, especially at a time like this. I thank you all from the bottom of my heart. Maurice.

Dear Mr. Swinnen,

In v5n1p2 you mention the existence of several printer sensing routines and in v6n2p16 of several peel off routines. Could you publish some of each in the newsletter.

Also, in v5n8p3 you list some devices produced by American Microproducts, Inc. What are exactly CROM simulators and emulators? Are the disk drives and line printers (which should not be confused with a PC100?) compter-like peripherals, but for the TI-59? If these accessories are being marketed, where can I write for further information?

A.C. ROchester, NY.

The printer sensing routine that readily comes to mind is : 20 OP 07 OP 19 CLR. If you put that at the beginning of your initialization routine, it will set flag 7 with the printer attached and not when the calculator alone is used. It is a simple trick to check flag 7 to go to a print sequence and to stop if flag 7 is not set. I also talked about printer checking routines in v5n1p2 and the PPX newsleter had something about it in vol.2, issue 3, page 1.

American Microproducts is situated in Richardson, TX. About a year ago they announced with lots of fanfare that they were going to produce (and market) a four-module selector. They even had a manual and an automatic one that could be called from a program. I used their prototype units for two weeks and found them OK. So, I put my findings in the newsletter, which prompted lots of members to inquire at American Microproducts or at their marketing division System-7. Many members complained later to me that they received no answer whatsoever. The only conclusion I can come to is, that I have been tricked in permitting the TI PPC Club to be used as a marketing survey guinea pig. I apologize and I vow it will never happen again. After almost a year I am still waiting for word if they are going to market the four-module switcher. I will not publish the address of American Microproducts here until I am assured that they are serious about the matter. In the mean time, if you want to inquire at System-7, please call Mr. Harvey Sperber at 806-792-2657. He should be in a position to give you information. As to the other peripheral TI-59 devices, I have seen pictures of them but I am not at all sure if they are available on the market. I have never seen one work. ED.

Dear Mr. Swinnen,

First of all, let me thank you for your prompt appraisal of the program I had sent you several weeks ago. I am doubly appreciative because you so graciously took so much time in the early days of your recent.....

While I was pleased to be able to prepare a program that filled a need, I realize that it falls short of the sophisticated models that I see in LRN and PPX. At the same time, I am happy that you suggested publication of my effort, as amended with your help. I hope that this step will motivate others to share their experiences with routine problems that crop up in (their respective professions).

Thank you again for your help and encouragement.

J.W. Lancaster, PA.

Published programs should be either practical (that is, help out somebody and make his daily grind a little easier) or it should teach us how to make good programs ourselves. Yours is a fine example of the first category. ED.

Dear Maurice,

I was glad you finally found Fred Fish. Please send me a copy of the Survival Manual.

B.E.Z. & 259 other members.

The first 200 copies have been mailed and a third reprinting has been ordered. When I came back from vacation my neighbor, who had dutifully emptied the mail box each day, handed me a bag full of mail you wouldn't believe. The guys at the local post office had another surprise for me, as my P.O. Box had suffered a slight indigestion. ED.

Dear Maurice,

Among many other items, I liked very much the "Poem Machine" program by Michael Malik. It does get a bit boring after a while, though, as the random generator tends to produce the same results with a lot of different seeds. And then, of course, in your comments, you did something what we call in German "den Speck durch den Mund ziehen," *(To make somebody's mouth water.ED)* by mentioning the Snow-Swinnen-underground versions of the saucier kind. Why not publish them to allow the club members serious comparative studies? If they are really of the no-no kind, you might consider to make them available by special request orders only.

If you have not mailed yet my (requested article) at the time you receive this letter, no hurry at all, you might perhaps throw in a copy of your infamous works...

Dr. P.W. Switzerland.

Dear P.W.,

To paraphrase the English shool boy who, when asked to write something about the Virgin Queen, commented "As a queen she was a great success", I am compelled to remark that the "proper English-speaking and-writing Swiss doctor" has a remarkable command of the English language.

Sure, I will publish the print code register contents of the infamous version in a future issue. The more that I received at least twenty five additional requests, of which more than twenty four came from members living in California. ED.

FAST MODE,- Patrick Acosta. Up to now, the only generally known way of getting into Fast Mode was the PGM 02 SBR 239 *method using the ML module. There are two other ways that work:*

1. STF IND 7 INV *which works only from the keyboard, and*
2. *a programmable version using a hex keycode* STF IND *h*12.

Both of these methods have certain advantages over the CROM method. All three have in common that they load the calculator's flag register, the CROM method loading the flag register from the following octet, probably after first altering that octet, and the two other methods loading the flag register from the display. This flag register (shown as register 0 in figure 19 of U.S. Patent # 4,153,937) has 16 digits. The digits relevant to fast mode are used as explained below:

User flags	*9*	*8*	*7*	*6*	*5*
User flags	*4*	*3*	*2*	*1*	*0*
First five digits of flag register	*d1*	*d2*	*d3*	*d4*	*d5*

If the digit is.....

0, 4 or 8	*Both corresponding flags are reset.*
1, 5 or 9	*The corresponding bottom flag is set and the corresponding top flag is reset.*
2 or 6	*The corresponding top flag is set and the corresponding bottom flag is reset.*
3 or 7	*Both corresponding flags are set.*

The ninth through twelfth digits store the address used in any kind of jump (except RTS) are encoded as follows:

W X Y Z — *WXY*: *octet*; *Z*: *byte*

Thus, the address above would normally be 8(WXY)+Z, except when accessing Fast Mode when the calculator will jump to the address 8*(WXY)+Z+1.*

The thirteenth digit is called the "Program Source Flag". This digit must be 2, 4 or 6 in order to enter Fast Mode.

Digit sixteen holds the fixed-point display mode. This digit will be zero for an INV FIX display; otherwise it will be n+2 in a FIX-n display.

Now to get into the STF IND 7 INV *type of Fast Mode, enter a 13-digit number into the display, with the 9th through the 12th digit encoding the address you want the Fast Mode to start at, and the 13th digit always equal to 2, 4 or 6. Then press:*
DEG FIX 0 RST STF IND 7 *(or any other digit)* INV *(or B, X+T, STO, EE).*

For example, pressing .00007032+44444= DEG FIX 0 RST STF IND 7 INV *will clear all flags and begin execution in Fast Mode at 8*(070)+3+1 = step 564 in an INV FIX display mode. Or:* 2 EE 12 +/- + 3 = RST DEG STF IND 7 INV *sets flags 9 and 4 and begins Fast Mode execution ar step 001.*

A few things you should watch out for:

1. *You must be in DEG mode to enter Fast Mode. If necessary, you may later in your Fast Mode program change to RAD or GRAD mode.*
2. *You may be in FIX 9 mode when entering Fast Mode only if the number in the display shows no fractional part. Otherwise you MUST be in FIX 0.*
3. *An EE or ENG mode is alright as long as rule 2 is also followed.*
4. *The above procedure does not automatically reload the "command buffer register". So you must press RST (unless the calculator has stopped at the 7th step of an octet) to ensure that this register is loaded immediately with the keycodes you want to execute. Otherwise, the calculator will execute the instructions in the buffer register intil reloaded at the end of the octet.*

5. *You canNOT begin execution at step 000.*
6. *You must have some memory in the current partition. So, do not use 0 OP 17.*
7. *Changing the signs of the 13-digit number and /or it's exponent will change the FIX n display mode.*
8. *Using 9 as the byte within the octet (digit 12 of the flag register) sends the calculator to step (WXY)*8+8. (Note to members unfamiliar with computer notation: the asterisk is used to denote "times or multiplication".)*

As a practical matter, it is most convenient to store the 13-digit number in an available register and recall it when needed. Thus, in program memory you could write: LBL E FIX 0 DEG RCL NN R/S. *Then from the keyboard you must press* E RST *(This last step could be omitted if the R/S is at a step congruent to 7 modulo 8)* STF IND 7 INV.

For those who think this is still too much key punching, put LBL E FIX 0 DEG RCL NN STF IND *in your program such that the IND is at the last step of the partition. Then you will only need to press* E 7 INV. *In that case, just put a CE instruction as the first step of your Fast Mode program, to clear the error condition caused by the aforementioned procedure.*

This method does not erase your program or data or change the partitioning, as the CROM method does. But most of the ususal rules of Fast Mode still apply.

The hexadecimal method is similar to the keyboard method, with this difference that it can be programmed. Assuming the 13-digit number is stored in register NN, the sequence is LBL A FIX 0 DEG RCL NN STF IND *h12. Remember that the hex key code can only be created at the first step of an octet. Since this method also doesn't immediately load the "command buffer register", only some of the next seven keycodes will be executed before the buffer is reloaded. How many and which ones are executed depends on the twelfth digit of the 13-digit number. So, to simplify matters, the instructions in the buffer register, following h12 are used to make the jump to the beginning of the Fast Mode sequence.The sequence then is*

LBL A FIX 0 DEG RCL NN STF IND *h12* NOP GTO NNN.

The address now loaded into the flag register is unimportant. You must only be sure that the GTO NNN is allowed to execute completely, which means that digit 12 must be $\leq$ *3. And, of course, digit 13 must be 2, 4 or 6. Note that the step following h12 is always ignored. Also, any keycode ending in C (2 in display) may be substituted for h12.*

The nice thing about this method is, that you may use library programs or statistics and conversion functions, then switch to Fast Mode under program control. Another nice feature , for calculators-alone programs, is that you may call the above LBL A as a subroutine in your Normal Mode program. Then, when your Fast Mode segment arrives at the RTN instruction, you return to Normal Mode at the point from which you called subroutine A. However, with the printer attached, the calculator seems to return in trace mode. (At least itdid for Palmer Hansen in one of the programs in which this was tried.) Note that the Fast Mode segment must be the lowest level subroutine. You still canNOT call subroutines in Fast Mode.

For the 58C , the STF h12 works just as well, but Palmer Hansen found found STF IND h12 necessary on his 59. Also notice that, if you single-step past the STF IND h12, none of the usual methods of stopping Fast Mode will work, except RST.

You can use sequences such as SBR NNN, SBR LBL, X=T LBL (but not GTO LBL) in Fast Mode. But this will cause a return to Normal Mode and if you arrive at a RTN instruction, the calculator begins execution somewhere in the CROM, usually causing a crash.

Other methods of causing a jump in the program counter might also be sudied for possible Fast Mode access. For instance, John Mairs' keycode translation in v5n4/5p18. Also, De Mees' quirk in v6n3p6 might be tried with various keycodes in the first octet. My 58C, besides not entering Fast Mode with PGM 02 SBR 239, doesn't have any of the CROM quirks either.

DIMENSIONS OPERATIONS. This is the fancy title we selected for the program I requested in v6n6/7p17: a program to do all four arithmetic operations to two dimensions entered in mixed feet, inches and fractions of inches.

From the many submissions we selected four that, in the opinion of the reviewers, had the most plus factors: speed, user-friendliness and good programming. The selected programs are by Jules Bell from Baltimore, Ralph Snyder from Indianapolis, Mike Shanock from Hamden, Connecticut and the duo John Worthington and Emil Regelman, from respectively Bowie, Maryland and Eagles Landing, Virginia. (Some local members have suggested that I use some sort of short-hand when mentionong the duo, such as "Worthman" or "Regelton", as their names appear so often in these pages.)

I know I will bore my European friends to death with four programs that deal with feet, inches and their fractions. They are used to and use exclusively centimeters, meters and kilometers ever since Napoleon did away with the el, the rod, the mile , a few of the inhabitants of France's neighbors and other, in the Marshal's view, inconsistencies. But again, as with the famous (and maybe infamous) calendar programs, the feet, inches and their fractions are of no real consequence to the majority of American and Canadian members either. Only to some civil engineers, mechanical engineers and surveyors will these programs offer any practical value. To the others it is nothing but a vehicle understood by almost everybody to learn new programming techniques.

And if you think there is nothing new in these programs, find out how Jules Bell used Program 20 of the M/U module, routine B, from steps 123 to 146, to work out his print-out logic. Or try to incorporate in your next program the t-register entry technique, so characteristic of a Worthington-Regelman program, but so seldomly encountered anywhere else.

You will find the four programs in question on the next four pages. TI-58 users will be gratified to find that each of these programs fits on their machines too.

SPEEDY FACTOR FINDER.- Björn Gustavsson. Björn didn't like the beginning of Patrick Acosta's SFF program in v6n4/5p13. It had too much button pushing. So he reworked the first 107 lines to reduce the manual key punching to an absolute minimum. The new instructions are:

1. Load bank 1.
2. Press A.
3. Re-load bank 1. Display will show a 2.
4. Load bank 2. Display will show a 3.
5. Load bank 3. Display will show a 4.
6. Load bank 4. Display will show a 0.
7. Enter an integer and press R/S. Factors will be printed.
   If you don't use a printer, write R/S at steps 090 and 101.
8. For a new number, go to step 7.

```
000  00  0       016  02  2       032  42  STO     048  61  GTO
001  00  0       017  91  R/S     033  02  02      049  00  00
002  00  0       018  03  3       034  29  CP      050  82  82
003  76  LBL     019  91  R/S     035  61  GTO     051  01  1
004  11  A       020  04  4       036  01  01      052  85  +
005  36  PGM     021  91  R/S     037  07  07      053  01  1
006  02  02      022  01  1       038  68  NOP     054  85  +
007  71  SBR     023  69  OP      039  08  8       055  01  1
008  02  02      024  17  17      040  05  5       056  85  +
009  39  39      025  25  CLR     041  00  0       057  01  1
010  09  9       026  91  R/S     042  61  GTO     058  85  +
011  00  0       027  42  STO     043  00  00      059  53  (
012  22  INV     028  01  01      044  82  82      060  43  RCL
013  58  FIX     029  98  ADV     045  08  8       061  02  02
014  22  INV     030  99  PRT     046  06  6       062  55  ÷
015  57  ENG     031  02  2       047  04  4       063  02  2

064  01  1       080  07  7       096  01  01
065  00  0       081  95  =       097  67  EQ
066  54  )       082  42  STO     098  01  01
067  22  INV     083  04  04      099  01  01
068  59  INT     084  43  RCL     100  99  PRT
069  65  ×       085  02  02      101  68  NOP
070  05  5       086  22  INV     102  61  GTO
071  04  4       087  49  PRD     103  00  00
072  95  =       088  01  01      104  25  25
073  59  INT     089  99  PRT     105  68  NOP
074  65  ×       090  68  NOP     106  68  NOP
075  01  1       091  83  GO*
076  04  4       092  04  04
077  85  +       093  01  1
078  01  1       094  32  X:T
079  00  0       095  43  RCL
```

SPEEDY FACTOR FINDER ON THE TI-57.- Björn Gustavsson. Björn also enhanced this one, originally written by Bill Skillman and published in v5n8p15. He used Peter Van Roy's trick in v5n6p12 to speed up Bill's program further. To list this program I used Björn's "TI-57 LISTNING PÅ TI-59" program in Programbiten, 80-3, pp 26-27 and 29. (By the way, Peter Van Roy has dropped out as an active TI-57 programmer. But a new giant has appeared in the form of an Ingmar Magnusson. What I have seen so far is impressive: A MIN/MAX program based on the Newton-Raphson method and a calendar program for the TI-57)

SFF program on the TI-57, B. Gustavsson, listing.

```
00  32 1 STD 1
01  15   CLR
02  32 0 STD 0
03  61 0 SBR 0
04  01     1
05  61 9 SBR 9
06  61 0 SBR 0
07  86 7 LBL 7
08  19   CT
09  61 8 SBR 8

10  61 8 SBR 8
11  04     4
12  61 1 SBR 1
13  06     6
14  61 9 SBR 9
15  06     6
16  61 1 SBR 1
17  33 0 RCL 0
18  23   X²

19  32 7 STD 7
20  33 1 RCL 1
21  76   GE
22  51 7 GTD 7
23  84   +/-
24  81   R/S
25  86 8 LBL 8
26  04     4
27  86 9 LBL 9

28  61 1 SBR 1
29  86 0 LBL 0
30  02     2
31  86 1 LBL 1
32  34 0 SUM 0
33  86 5 LBL 5
34  89     x̄
35 -49   INV INT
36 -66   INV EQ

37 -61   INV SBR
38  33 1 RCL 1
39  32 7 STD 7
40  33 0 RCL 0
41 -39 1 INV PRD 1
42  66   EQ
43  84   +/-
44  19   CT
45  81   R/S
46  51 5 GTD 5
```

PROGRAM LOADER, - Björn Gustavsson. This unique program could be a possible candidate for our own module. It makes it possible to load programs without going into LRN mode !!!

Instructions:

1. *Enter a register number and press or call E.*

2a. *Enter a single code and press or call A.*

OR

2b. *Enter one to five codes and call/press B. Codes will not be separated.*

3. *When all codes are entered, press/call C.*
You may now re-partition and call the entered program with SBR nnn.

Example:

Let's enter the following program: LBL A STO 00 LBL CE RCL 00 X DSZ 0 CE 1 = RTN
We want the program to begin at step 480, so we enter 59 as the register number and we press E. Then we enter the key codes as follows:

7611 B 4200 B 7624 B 4300 B 65 A 970024 B 1 A 95 A 92 A C

To run the entered program, first press 2nd CP to clear the loading program. Re-partition by pressing 5 OP 17. Then enter, say, 5 and press A. See 120 displayed. (you might have noticed by now that this is a factorial program)

You can list the program by pressing GTO 480 2nd LIST. As you can see, there are a lot of "2nd" steps in it. They will not bother the program, as they have the effect of a NOP instruction.(with an additional advantage that it can be created at any and every step in memory)

The "A" routine is intended for entering single codes. The "B" routine for those that cannot be separated, such as 4201 (STO 01) or 970024 (DSZ 0 CE). This routine will fill part of a register with "2nd" (code 26) instructions and place the codes in the next register, in order to avoid separation.

This program could be used by a TI-57 compiler, which translates TI-57 key codes into a program for the TI-59 and loads it ! (and runs it, of course !)

```
000 76 LBL
001 13  C
002 43 RCL
003 01  01
004 29 CP
005 67  EQ
006 00  00
007 39  39
008 02  2
009 06  6
010 11  A
011 61 GTD
012 13  C
013 76 LBL
014 11  A
015 69 DP
016 21  21
017 48 EXC
018 01  01
019 32 X:T
020 01  1
021 67  EQ
022 00  00
023 60  60
024 07  7
025 67  EQ
026 00  00
027 40  40

028 32 X:T
029 48 EXC
030 01  01
031 44 SUM
U32 02  02
033 93  .
034 00  0
035 01  1
036 49 PRD
037 02  02
038 00  0
039 92 RTN
040 01  1
041 00  0
042 32 X:T
043 43 RCL
044 01  01
045 77  GE
046 00  00
047 91  91
048 02  2
049 06  6
050 44 SUM
051 02  02
052 43 RCL
053 02  02
054 94 +/-
055 64 PD*

056 00  00
057 69 DP
058 30  30
059 01  1
060 53  (
061 48 EXC
062 01  01
063 65  ×
064 93  .
065 01  1
066 42 STD
067 02  02
068 75  -
069 59 INT
070 49 PRD
071 02  02
072 54  )
073 53  (
074 94 +/-
075 65  ×
076 01  1
077 00  0
078 00  0
079 75  -
080 03  3
081 54  )
082 22 INV
083 28 LOG

084 52 EE
085 22 INV
086 52 EE
087 72 ST*
088 00  00
089 00  0
090 92 RTN
091 44 SUM
092 02  02
093 43 RCL
094 02  02
095 94 +/-
096 64 PD*
097 00  00
098 69 DP
099 30  30
100 00  0
101 42 STD
102 01  01
103 92 RTN
104 76 LBL
105 12  B
106 53  (
107 29 CP
108 55  ÷
109 67  EQ
110 01  01
111 15  15

112 34 ΓX
113 28 LOG
114 59 INT
115 42 STD
116 03  03
117 22 INV
118 28 LOG
119 33 X²
120 52 EE
121 22 INV
122 52 EE
123 54  )
124 82 HIR
125 04  04
126 69 DP
127 23  23
128 43 RCL
129 03  03
130 32 X:T
131 53  (
132 06  6
133 75  -
134 43 RCL
135 01  01
136 54  )
137 77  GE
138 01  01
139 41  41

140 13  C
141 82 HIR
142 14  14
143 59 INT
144 82 HIR
145 54  54
146 11  A
147 01  1
148 00  0
149 00  0
150 82 HIR
151 44  44
152 97 DSZ
153 03  03
154 01  01
155 41  41
156 00  0
157 92 RTN
158 76 LBL
159 15  E
160 42 STD
161 00  00
162 00  0
163 42 STD
164 01  01
165 92 RTN

480 26 2ND
481 76 LBL
482 11  A
483 42 STD
484 00  00
485 76 LBL
486 24 CE
487 26 2ND
488 26 2ND
489 43 RCL
490 00  00
491 65  ×
492 97 DSZ
493 00  00
494 24 CE
495 26 2ND
496 26 2ND
497 01  1
498 95  =
499 92 RTN
500 26 2ND
501 26 2ND
502 26 2ND
503 26 2ND
```

DIMENSIONS (feet, inches, fractions) OPERATIONS, - Ralph W. Snyder. This program adds, subtracts, multiplies or divides two (and more) dimensions expressed in feet, inches and fractions of inches.
The entry format is FF.IISS in which FF=feet, II=inches and SS=16ths of an inch.
Printout: Each dimension as entered is printed in FF.IISS format, followed by an F.f printout, in which F=feet and .f=decimal fraction of a foot. Answer also given in F.f.
Optional printout: after answer in F.f,it's possible to obtain an answer in FF.IISS.
Run Time: about 1 sec per dimension, with printout. 1 Sec for reconversion to FF.IISS.

User Instructions:
1. Enter first dimension in FF.IISS format and press A.
2a Enter second dimension in FF.IISS format.
 b To add, press B; to subtract press +/- B; repeat 2a, 2b as often as needed.
 c For sq.ft, press C.
   For cu.ft, after step 2c, enter third dimension and press C.
 d To divide, press D.
3 Optional after steps 2b, 2c or 2d, press R/S for reconversion to FF.IISS format.

Example:

Suppose you want to add 2' 6 3/4" to 3' 2 7/16". (remember to mentally convert fractions of inches to 16ths of inches. So 3/4" becomes 12/16")
Enter 2.0612 and press A. Print outs: 2.0612 and 2.5625 (the last one is F.f format)
Enter 3.0207 and press B. Print outs: 3.0207 and 5.765625 (sum in F.f format)
To reconvert the result, press R/S. Print out: 5.0903 or 5'9 3/16"
To subtract say 3' ½" for this result:
Enter 3.0008 +/- B. Print out: -3.0008, 3.041666667 (input in F.f) and 2.723958333 (the answer in F.f format)
To reconvert the answer, press R/S. Print out: 2.0811 or 2' 8 11/16".

Suppose you want to multiply 4' 8 5/8" with 3' 2 7/16".
Enter 4.0810 and press A. See sample print out.
Enter 3.0207 and press C. Press R/S to reconvert.

Suppose you now want to multiply this result by 3' 6 3/8".
Enter 3.0606 and press C. To reconvert, press R/S.

Suppose you now want to devide this result by 3' 6 3/8".
Enter 3.0606 and press D. To reconvert, press R/S.

All print-outs of the examples are shown below with the program listing.

Derivations as follows:

$$F.f = FF + \frac{II}{12} + \frac{SS}{12 \times 16} = FF + \frac{II}{12} + \frac{\frac{SS}{100} \times \frac{25}{4}}{12} = FF + \frac{II}{12} + \frac{.SS\left[1 + \frac{21}{4}\right]}{12} = FF + \frac{II.SS + .SS \times 21/4}{12}$$

$$FF.IISS = F.f - .f + \frac{.f \times 12 - .i}{100} + \frac{.i \times 16}{100 \times 100} = F + \frac{.f \times 12}{100} - \frac{.i}{100} + \frac{.i \times .16}{100}$$

$$FF.IISS = F + \frac{.f \times 12}{100} - \frac{.i(1 - .16)}{100} = F + \frac{.f \times 12 - .i \times .84}{100},$$ where .f = decimal foot and .i = decimal inch generated by .f x 12.

Note that conversion and reconversion is programmed in 16ths mode. That is, limit on fractions is 16ths, and all fractions with denominators less than 16 are changed mentally to 16ths before entering. However, if your particular discipline requires you to work more with either 8ths or 4ths, it is possible to reprogram for those alternatives.

Mode.	Steps 018 through 021 (conversion)	Steps 096 through 098.(Reconv.)
16ths	2 1 DIV 4	. 8 4
8ths	2 3 DIV 2	. 9 2
4ths	2 4 NOP NOP	. 9 6

DIM.OPS. Ralph Snyder, program listing.

```
        2.0612
        2.5625
        3.0207
      3.203125
      5.765625
        5.0903

       -3.0008
  -3.041666667
   2.723958333
        2.0811

         4.081
       4.71875
        3.0207
      3.203125
   15.11474609
   15.01060313

        3.0606
       3.53125
   53.37394714
   53.04077979
```

```
000  76 LBL      019  01  1       037  76 LBL      056  43 RCL      075  99 PRT      093  22 INV
001  16 A'       020  55  ÷       038  12  B       057  01  01      076  61 GTO      094  59 INT
002  99 PRT      021  04  4       039  53  (       058  54  )       077  00  00      095  65  ×
003  75  -       022  54  )       040  16 A'       059  99 PRT      078  48  48      096  93  .
004  53  (       023  55  ÷       041  85  +       060  42 STO      079  91 R/S      097  08  8
005  22 INV      024  01  1       042  43 RCL      061  01  01      080  76 LBL      098  04  4
006  59 INT      025  02  2       043  01  01      062  61 GTO      081  17 B'       099  54  )
007  75  -       026  54  )       044  54  )       063  00  00      082  75  -       100  65  ×
008  53  (       027  54  )       045  99 PRT      064  48  48      083  53  (       101  93  .
009  24 CE       028  99 PRT      046  42 STO      065  91 R/S      084  22 INV      102  00  0
010  65  ×       029  92 RTN      047  01  01      066  76 LBL      085  59 INT      103  01  1
011  01  1       030  76 LBL      048  91 R/S      067  14  D       086  75  -       104  54  )
012  00  0       031  11  A       049  61 GTO      068  53  (       087  53  (       105  54  )
013  00  0       032  53  (       050  17 B'       069  16 A'       088  24 CE       106  99 PRT
014  85  +       033  16 A'       051  76 LBL      070  35 1/X      089  65  ×       107  69 OP
015  22 INV      034  42 STO      052  13  C       071  65  ×       090  01  1       108  05  05
016  59 INT      035  01  01      053  53  (       072  43 RCL      091  02  2       109  91 R/S
017  65  ×       036  91 R/S      054  16 A'       073  01  01      092  75  -
018  02  2                        055  65  ×       074  54  )
```

DIMENSIONS OPERATIONS,(feet, inches, fractions)-Michael E. Shanock. Allows the user to quickly manipulate string addition, subtraction, multiplication or division of mixed feet, inches and fractions of inches, or convert feet in fractional form. The program works with or without the printer and does not require any module.

User Instructions:

1. Enter feet and press A.
2. Enter inches and press B.
3. Enter numerator of the fraction and press C.
4. Enter denominator of the fraction and press D.
5. Call for conversion to feet, inches and fraction to nearest 16ths, press E.
6. To convert feet in fractions to feet/inches/fraction: Enter feet in fractional form and press 2nd E'.

Note that answers may be added, subtracted, multiplied or divided by using the four operations keys and the equal sign. For example: The sum of 7' 11 ¼" and 5' 3/8" is to divided by 3 and expressed in feet-inches to the nearest 16th of an inch:

ENTER	PRESS	DISPLAY
7	A	7
11	B	7.9166
1	C	disregard
4	D	7.9375
	+	
5	A	5
0	B	0
3	C	disregard
8	D	12.96875 (total feet)
	DIV 3 =	4.3229
	2nd E'	4 (feet)
		3 (inches)
		14 (16ths)= 7/8"

The user may, on occasion, get an answer which includes 16/16, for example, 8.4978 feet converts to 8' 5 16/16". This indicates that the dimension is less than, but rounds out to 8' 6".

To change fractional roundoff, replace the digits 16 (steps 065-066) with whatever roundoff denominator is desired.

One could elaborate on this program some more, adding bells and whistles such as descriptors, user entry of roundoff factor before starting the program (store that factor and recall it at steps 065-066. But store it in a register that gets recorded on the mag card, together with the program.) etc. etc.

DIM.OPS., Mike Shanok,- program listing.

11.
11.
5.
8.
11.96875
10.
0.
3.
8.
22.
5.
5.
16.

000	76	LBL
001	16	A'
002	65	×
003	32	X:T
004	01	1
005	95	=
006	59	INT
007	66	PAU
008	66	PAU
009	66	PAU
010	66	PAU
011	99	PRT
012	32	X:T
013	22	INV
014	59	INT
015	92	RTN
016	76	LBL
017	11	A
018	32	X:T
019	53	(
020	32	X:T
021	85	+
022	99	PRT
023	42	STO
024	00	00
025	91	R/S
026	76	LBL
027	12	B
028	99	PRT
029	55	÷
030	01	1
031	02	2
032	85	+
033	42	STO
034	00	00
035	91	R/S
036	76	LBL
037	13	C
038	99	PRT
039	55	÷
040	01	1
041	02	2
042	55	÷
043	91	R/S
044	76	LBL
045	14	D
046	99	PRT
047	95	=
048	24	CE
049	99	PRT
050	98	ADV
051	91	R/S
052	76	LBL
053	15	E
054	25	CLR
055	43	RCL
056	00	00
057	76	LBL
058	10	E'
059	16	A'
060	65	×
061	01	1
062	02	2
063	16	A'
064	65	×
065	01	1
066	06	6
067	95	=
068	58	FIX
069	00	00
070	52	EE
071	22	INV
072	52	EE
073	22	INV
074	58	FIX
075	99	PRT
076	98	ADV
077	98	ADV
078	98	ADV
079	91	R/S

DIMENSIONS OPERATIONS, (feet, inches, fractions)- Jules Bell. This program will add, subtract, multiply and divide two dimensions expressed in feet, inches and fractions of inches.

NOTE: THIS PROGRAM REQUIRES THE M/U MODULE.

User instructions:

Press E to initialize.
Enter feet and press A.
Enter inches and press B.
Enter fractions of inches in format NN.DD in which NN is the numerator and DD is the denominator of the fraction. Thus 5/16 is entered as 5.16 and 1/2 as 1.02.
Enter the FUNCTION DIGIT and press D.

For addition enter 1. For subtraction enter 2.
For multiplication enter 3. For division enter 4.

Enter second dimension, same method as above, through keys A, B and C.
The result will be printed only in three lines with descriptors, in feet, inches and 32nds of inches.

Note that zero entries must be entered as 0.

To multiply or didide by a factor, do all the steps up to and including the entering of either 3 for multiply or 4 for divide and pressing D. Then enter the factor and press 2nd A'. The result only will be printed.

19.	FEET
2.	INCH
31.0	32ND
8.	FEET
6.	INCH
1.0	32ND
9.	FEET
10.	INCH
4.2	32ND
21.	FEET
4.	INCH
0.0	32ND
10.	FEET
8.	INCH
0.0	32ND

000	91	R/S
001	76	LBL
002	15	E
003	25	CLR
004	22	INV
005	58	FIX
006	47	CMS
007	81	RST
008	76	LBL
009	11	A
010	82	HIR
011	08	08
012	91	R/S
013	76	LBL
014	12	B
015	82	HIR
016	07	07
017	91	R/S
018	76	LBL
019	14	D
020	42	STO
021	03	03
022	86	STF
023	40	IND
024	03	03
025	65	×
026	02	2
027	85	+
028	03	3
029	02	2
030	95	=
031	48	EXC
032	03	03
033	91	R/S
034	85	+
035	92	RTN
036	75	-
037	92	RTN
038	65	×
039	92	RTN
040	55	÷
041	92	RTN
042	76	LBL
043	16	A'
044	22	INV
045	87	IFF
046	06	06
047	00	00
048	66	66
049	87	IFF
050	05	05
051	00	00
052	66	66
053	87	IFF
054	01	01
055	00	00
056	66	66
057	87	IFF
058	02	02
059	00	00
060	66	66
061	86	STF
062	05	05
063	61	GTO
064	01	01
065	04	04
066	36	PGM
067	02	02
068	19	D'
069	36	PGM
070	02	02
071	17	B'
072	15	E
073	76	LBL
074	13	C
075	29	CP
076	67	EQ
077	00	00
078	92	92
079	82	HIR
080	06	06
081	59	INT
082	55	÷
083	82	HIR
084	16	16
085	22	INV
086	59	INT
087	55	÷
088	01	1
089	02	2
090	00	0
091	00	0
092	54	)
093	85	+
094	82	HIR
095	17	17
096	55	÷
097	01	1
098	02	2
099	54	)
100	85	+
101	82	HIR
102	18	18
103	54	)
104	87	IFF
105	06	06
106	01	01
107	13	13
108	86	STF
109	06	06
110	82	HIR
111	05	05
112	91	R/S
113	32	X:T
114	82	HIR
115	15	15
116	71	SBR
117	40	IND
118	03	03
119	32	X:T
120	95	=
121	42	STO
122	02	02
123	36	PGM
124	20	20
125	12	B
126	22	INV
127	58	FIX
128	32	X:T
129	03	3
130	05	5
131	67	EQ
132	01	01
133	47	47
134	03	3
135	67	EQ
136	02	02
137	16	16
138	04	4
139	05	5
140	67	EQ
141	01	01
142	47	47
143	04	4
144	67	EQ
145	02	02
146	54	54
147	02	2
148	01	1
149	01	1
150	07	7
151	01	1
152	07	7
153	03	3
154	07	7
155	69	OP
156	04	04
157	43	RCL
158	02	02
159	59	INT
160	69	OP
161	06	06
162	02	2
163	04	4
164	03	3
165	01	1
166	01	1
167	05	5
168	02	2
169	03	3
170	69	OP
171	04	04
172	43	RCL
173	02	02
174	22	INV
175	59	INT
176	65	×
177	01	1
178	02	2
179	95	=
180	52	EE
181	22	INV
182	52	EE
183	42	STO
184	02	02
185	59	INT
186	69	OP
187	06	06
188	00	0
189	04	4
190	00	0
191	03	3
192	03	3
193	01	1
194	01	1
195	06	6
196	69	OP
197	04	04
198	43	RCL
199	02	02
200	22	INV
201	59	INT
202	55	÷
203	93	.
204	00	0
205	03	3
206	01	1
207	02	2
208	05	5
209	95	=
210	58	FIX
211	01	01
212	69	OP
213	06	06
214	98	ADV
215	15	E
216	03	3
217	06	6
218	03	3
219	04	4
220	02	2
221	01	1
222	03	3
223	07	7
224	69	OP
225	04	04
226	43	RCL
227	02	02
228	59	INT
229	69	OP
230	06	06
231	03	3
232	06	6
233	03	3
234	04	4
235	02	2
236	04	4
237	03	3
238	01	1
239	69	OP
240	04	04
241	43	RCL
242	02	02
243	22	INV
244	59	INT
245	65	×
246	01	1
247	04	4
248	04	4
249	95	=
250	58	FIX
251	02	02
252	69	OP
253	06	06
254	98	ADV
255	15	E
256	43	RCL
257	02	02
258	58	FIX
259	02	02
260	99	PRT
261	98	ADV
262	15	E

DIMENSIONS OPERATIONS (feet, inches), John Worthington and Emil Regelman. This program will add, subtract, multiply and divide two dimensions entered in the form FF.II in which FF is the number of feet and II the number of inches. It will also permit to raise a mixed dimension to the power of a factor or of another mixed dimension ! It also allows to multiply or divide by a factor. Two routines are provided to convert dimensions from FF.II into FF.fractional feet and vice versa.
The program works only with the printer attached, but does not require a module.

FF.II → FF.ff	FF.ff → FF.II	X N	DIV N	Y^X N
+	-	X	DIV	Y^X

USER INSTRUCTIONS:

1. Always enter the first dimension into the t-register. Enter and press X:T.
2. To add to the first dimension, enter the second one and press A.
3. To subtract from the first dimension, enter the second one and press B.
4. To multiply by the first dimension, enter the second one and press C.
5. To divide the first by the second dimension, enter the second one and press D.
6. To raise the first to the power of the second one, enter the second one and press E.
7. To multiply by a factor, enter the factor and press 2nd C'.
8. To divide by a factor, enter the factor and press 2nd D'.
9. To raise to the power of a factor, enter the factor and press 2nd E'.
10. To convert feet.inches to feet.fractional feet, enter FF.II and press 2nd A'.
11. To convert feet.fractional feet to feet.inches, enter FF.ff and press 2nd B'.

The results of 2 through 9 are printed only; 10 and 11 are displayed only.
It would be rather simple to modify the program to add print commands to routines A' and B', if required. (it would play havoc in other routines, though)

```
      10.04    FT.I
          +
      12.09    FT.I
-----------
      23.01    FT.I

      23.01    FT.I
          -
      10.04    FT.I
-----------
      12.09    FT.I

      10.04    FT.I
          ×
       1.02    FT.I
-----------
12.00666667    FT.I

      100.1    FT.I
          ÷
      10.02    FT.I
-----------
9.110163934    FT.I

       6.05    FT.I
         Yx
       2.04    FT.I
-----------
76.0612820     FT.I

      10.08    FT.I
          ×
         2.
-----------
      21.04    FT.I

      21.04    FT.I
          ÷
         2.
-----------
      10.08    FT.I

      10.06    FT.I
         Yx
         3.
-----------
   1157.075    FT.I
```

```
000  76 LBL     045  76 LBL     091  17 B'      136  16 A'      180  87 IFF
001  69 OP      046  17 B'      092  71 SBR     137  17 B'      181  01  01
002  82 HIR     047  53  (      093  69 OP      138  71 SBR     182  01  01
003  07  07     048  53  (      094  16 A'      139  69 OP      183  90  90
004  01  1      049  52 EE      095  75  -      140  16 A'      184  16 A'
005  02  2      050  75  -      096  69 OP      141  55  ÷      185  17 B'
006  22 INV     051  59 INT     097  00  00     142  69 OP      186  71 SBR
007  28 LOG     052  82 HIR     098  01  1      143  00  00     187  69 OP
008  82 HIR     053  08  08     099  00  0      144  01  1      188  16 A'
009  08  08     054  54  )      100  00  0      145  00  0      189  76 LBL
010  02  2      055  65  ×      101  02  2      146  00  0      190  99 PRT
011  01  1      056  93  .      102  61 GTO     147  07  7      191  32 X:T
012  03  3      057  01  1      103  01  01     148  02  2      192  02  2
013  07  7      058  02  2      104  75  75     149  61 GTO     193  00  0
014  04  4      059  85  +      105  76 LBL     150  01  01     194  02  2
015  00  0      060  82 HIR     106  18 C'      151  75  75     195  00  0
016  03  3      061  18  18     107  86 STF     152  76 LBL     196  02  2
017  03  3      062  54  )      108  01  01     153  10 E'      197  00  0
018  82 HIR     063  22 INV     109  76 LBL     154  86 STF     198  02  2
019  38  38     064  52 EE      110  13  C      155  01  01     199  00  0
020  82 HIR     065  92 RTN     111  53  (      156  76 LBL     200  02  2
021  17  17     066  76 LBL     112  32 X:T     157  15  E      201  00  0
022  69 OP      067  11  A      113  16 A'      158  53  (      202  69 OP
023  06  06     068  53  (      114  17 B'      159  32 X:T     203  01  01
024  92 RTN     069  32 X:T     115  71 SBR     160  16 A'      204  69 OP
025  76 LBL     070  16 A'      116  69 OP      161  17 B'      205  02  02
026  16 A'      071  17 B'      117  16 A'      162  71 SBR     206  01  1
027  53  (      072  71 SBR     118  65  ×      163  69 OP      207  00  0
028  53  (      073  69 OP      119  69 OP      164  16 A'      208  00  0
029  24 CE      074  16 A'      120  00  00     165  45 Yx      209  02  2
030  75  -      075  85  +      121  01  1      166  69 OP      210  00  0
031  59 INT     076  69 OP      122  00  0      167  00  00     211  02  2
032  82 HIR     077  00  00     123  00  0      168  01  1      212  82 HIR
033  08  08     078  01  1      124  05  5      169  00  0      213  07  07
034  54  )      079  00  0      125  61 GTO     170  00  0      214  69 OP
035  55  ÷      080  00  0      126  01  01     171  04  4      215  05  05
036  03  3      081  04  4      127  75  75     172  05  5      216  32 X:T
037  55  ÷      082  07  7      128  76 LBL     173  06  6      217  54  )
038  02  2      083  61 GTO     129  19 D'      174  06  6      218  17 B'
039  05  5      084  01  01     130  86 STF     175  82 HIR     219  71 SBR
040  85  +      085  75  75     131  01  01     176  07  07     220  69 OP
041  82 HIR     086  76 LBL     132  76 LBL     177  69 OP      221  98 ADV
042  18  18     087  12  B      133  14  D      178  05  05     222  29 CP
043  54  )      088  53  (      134  53  (      179  32 X:T     223  22 INV
044  92 RTN     089  32 X:T     135  32 X:T                     224  86 STF
                090  16 A'                                      225  01  01
                                                                226  92 RTN
```

ASTRONOMICAL FORMULAE FOR CALCULATORS,- Jean Meeus, Volkssterrenwacht (Popular Observatory) at Hove and Vereniging voor Astronomie (Society for Astronomy) at Brussels, Belgium, 185 pages, Vol. 4, 1978. This book, written in English, intends to be a guide to the amateur astronomer who wants to do calculations. It is NOT intended as a general textbook on astronomy. Elementary knowledge of astronomy is taken for granted. It contains a wealth of formulas and worked-out problems to satisfy even the most demanding amateur astronomer.

It does NOT contain programs either. But it directs itself specifically to the amateur astronomer who has a programmable calculator. Using the formulas as a guide, it will be a cinch to develop complete and practical programs for either the AOS or the RPN variety of programmable calculators. In fact, this book will, in my humble opinion, keep astromer-programmers busy for years to come.

The book does not treat all possible subjects of astronomy. You will not find anything here on orbit determination, nor on occultation of stars by the moon, but you will find the following thirty-nine chapters:

1. Hints and tips.
2. Interpolation.
3. Julian Day and Calendar Date.
4. Date of Easter.
5. Ephimeris Time & Universal Time.
6. Geocentric Rectangular Coordinates of an Observer.
7. Sideral Time at Greenwich.
8. Transformation of Coordinates.
9. Angular Separation.
10. Conjunction between two Planets.
11. Bodies in Straight Line.
12. Smallest Circle containing three Celestial Bodies.
13. Position Angle of the Moon's Bright Limb.
14. Precession.
15. Nutation.
16. Apparent Place of a Star.
17. Reduction of the Ecliptical Elements from one Equinox to another one.
18. Solar Coordinates.
19. Rectangular Coordinates of the Sun.
20. Equinoxes and Solstices.
21. Equation of Time.
22. Equation of Kepler.
23. Elements of the Planetary Orbit.
24. Elliptic Motion.
25. Planets: Principal Perturbations.
26. Parabolic Motion.
27. Planets in Perihelion and Aphohelion.
28. Passages through the Nodes.
29. Correction for Paralax.
30. Position of the Moon.
31. Illuminated Fraction of the Moon's Disk.
32. Phases of the Moon.
33. Eclipses.
34. Illuminated Fraction of the Disk of a Planet.
35. Positions of the Satellites of Jupiter.
36. Semi-diameters of the Sun, the Moon and the Planets.
37. Stellar Magnitude.
38. Binary Stars.
39. Linear Regression; Correlation.

In writing a program to solve some astronomical problem, it will be required sometimes to study more than one chapter. For instance, in order to create a program for the calculation of the Sun's altitude for a given time of a given date at a given place, one must convert the time and date to Julian Date (chapter 3), then calculate the Sun's longitude for that time (chapter 18), its right ascension (chapter 8), the sideral time (chapter 7)and finally the required altitude of the Sun (chapter 8 again).

This is a delightful book.

PARABOLIC CURVE FIT,- According to Asa Reed, this program which appeared in v6n6/7p28, has to errors: Reg.61 should end in 00, not in 10, and step 008 should be E', not IxI.

BACKGAMMON.- Those who have this program by Brian Sladen, PPX # 918217, should make the following correction to it. Lem Matteson tells me that it concerns card 2A, page 21. Starting with step 328 and ending with step 349, key in these new steps:

RCL 00 - INT STO 00 = X 10 = EXC 00 DIV 10 = SUM 00 X:T ADV RTN

There is not enough space available here to print Lem's two page justification for this correction. Backgammon fanatics may send me a SASE to obtain a copy of Lem's epistle.

CERTIFICATE OF DEPOSIT COMPARISON PROGRAM , *J.W. Beattie, Lancaster PA.*

Purpose: To determine potential gain or loss if proceeds of Certificate of Deposit are prematurely withdrawn for re-investment at a higher interest rate.

Read card: Sides 1 & 3 Initialize program	ENTER:	PRESS: RST E	PRINTER: 0 DDATE
	Deposit date, or date of most recent interest payment. (See Note 1). MMDD.YYYY	R/S	nnnn.nnnn CDATE
	Current date: MMDD.YYYY	R/S	nnnn.nnnn -DAYS nnn CDATE nnnn.nnnn MDATE
	Maturity Date: MMDD.YYYY	R/S	nnnn.nnnn +DAYS nnn PRIN.
	Sum deposited: nnnnn.nn	R/S	nnnnn.nn % NOW
	Interest rate: nn.nn	R/S	nn.nn HOLD? SUM 1 nnnnn.nn PNLTY nnn.nn PRIN. nnnnn.nn FUT.%
	Projected interest rate: nn.nn	R/S	nn.nn SUM 2 nnnnn.nn + / - nnnn.nn ALT.%
	Alternate interest rate: nn.nn	R/S	nn.nn = DAYS nnn

NOTES

1. Program will calculate present value (P.VAL) and principal remaining after penalty as though interest has been accumulated from original date of deposit.

   When present value and principal, after penalty, must be modified by reason of periodic prior payments of interest, however, the date of the most recent interest payment must be substituted for the original date of deposit.

2. Program is designed to calculate a penalty of six months' interest by dividing annual interest by 2. When a specific situation requires an alternate calculation, GTO 138 to correct the divisor.

3. (= % 6 MO) was included in program to display what interest rate must be applied to principal, after penalty, to match SUM 1 in only six months.

4. (= DAYS) displays and prints the number of days which will be required to match SUM 1 if principal, after penalty, is invested at an alternate rate.

5. Responses printed as described in 3 and 4, above, may be verified from the keyboard as follows:

# 3	# 4
RCL	RCL
27	27
X	X
(= % 6MO)	RCL
÷	25
2	÷
+	RCL
RCL	34
27	X
=	(= DAYS)
SUM 1	+
(Approx) *	RCL
	27
	=
	SUM 1
	(Approx) *

* Allow for variation created by "FIX" selection.

```
000  76 LBL    048  00  00    096  42 STO    144  43 RCL    192  17  17
001  16 A'     049  69 OP     097  16  16    145  19  19    193  75  -
002  42 STO    050  01  01    098  43 RCL    146  75  -     194  43 RCL
003  06  06    051  69 OP     099  12  12    147  43 RCL    195  27  27
004  36 PGM    052  05  05    100  85  +     148  18  18    196  95  =
005  20  20    053  92 RTN    101  43 RCL    149  95  =     197  42 STO
006  11  A     054  16 A'     102  13  13    150  42 STO    198  11  11
007  43 RCL    055  91 R/S    103  95  =     151  27  27    199  55  ÷
008  06  06    056  17 B'     104  42 STO    152  99 PRT    200  43 RCL
009  99 PRT    057  18 C'     105  26  26    153  10 E'     201  27  27
010  10 E'     058  42 STO    106  65  ×     154  91 R/S    202  95  =
011  92 RTN    059  12  12    107  43 RCL    155  99 PRT    203  65  ×
012  76 LBL    060  99 PRT    108  16  16    156  55  ÷     204  02  2
013  17 B'     061  10 E'     109  95  =     157  19 D'     205  95  =
014  42 STO    062  43 RCL    110  85  +     158  42 STO    206  99 PRT
015  06  06    063  06  06    111  43 RCL    159  20  20    207  10 E'
016  36 PGM    064  16 A'     112  10  10    160  10 E'     208  91 R/S
017  20  20    065  91 R/S    113  95  =     161  43 RCL    209  99 PRT
018  12  B     066  17 B'     114  42 STO    162  20  20    210  55  ÷
019  43 RCL    067  18 C'     115  17  17    163  65  ×     211  19 D'
020  06  06    068  42 STO    116  99 PRT    164  43 RCL    212  42 STO
021  99 PRT    069  13  13    117  10 E'     165  27  27    213  25  25
022  10 E'     070  99 PRT    118  43 RCL    166  55  ÷     214  10 E'
023  92 RTN    071  10 E'     119  10  10    167  43 RCL    215  43 RCL
024  76 LBL    072  58 FIX    120  85  +     168  34  34    216  25  25
025  18 C'     073  02  02    121  53  (     169  65  ×     217  65  ×
026  36 PGM    074  91 R/S    122  43 RCL    170  43 RCL    218  43 RCL
027  20  20    075  42 STO    123  12  12    171  13  13    219  27  27
028  13  C     076  10  10    124  65  ×     172  95  =     220  55  ÷
029  92 RTN    077  99 PRT    125  43 RCL    173  85  +     221  43 RCL
030  76 LBL    078  10 E'     126  16  16    174  43 RCL    222  34  34
031  19 D'     079  91 R/S    127  95  =     175  27  27    223  95  =
032  01  1     080  99 PRT    128  42 STO    176  95  =     224  42 STO
033  00  0     081  55  ÷     129  19  19    177  99 PRT    225  28  28
034  00  0     082  19 D'     130  99 PRT    178  42 STO    226  43 RCL
035  95  =     083  42 STO    131  10 E'     179  23  23    227  11  11
036  92 RTN    084  14  14    132  43 RCL    180  10 E'     228  55  ÷
037  76 LBL    085  10 E'     133  10  10    181  43 RCL    229  43 RCL
038  15  E     086  10 E'     134  65  ×     182  23  23    230  28  28
039  03  3     087  43 RCL    135  43 RCL    183  75  -     231  95  =
040  04  4     088  10  10    136  14  14    184  43 RCL    232  58 FIX
041  42 STO    089  65  ×     137  55  ÷     185  17  17    233  00  00
042  00  00    090  43 RCL    138  02  2     186  95  =     234  99 PRT
043  76 LBL    091  14  14    139  95  =     187  42 STO    235  98 ADV
044  10 E'     092  55  ÷     140  42 STO    188  21  21    236  98 ADV
045  69 OP     093  43 RCL    141  18  18    189  99 PRT    237  22 INV
046  20  20    094  34  34    142  99 PRT    190  10 E'     238  58 FIX
047  73 RC*    095  95  =     143  10 E'     191  43 RCL    239  91 R/S
```

```
         365.  34
 1616133717.  35
 1516133717.  36
 2016134536.  37
 1516133717.  38
 3016133717.  39
 4716134536.  40
 3335243140.  41
 6100313243.  42
 2332271671.  43
 3641300002.  44
 3340421327.  45
 3331273745.  46
 3335243140.  47
 2141374061.  48
 3641300003.  49
 4700630020.  50
 6461073032.  51
 1327374061.  52
 6416134536.  53
          0.  54
          0.  55
          0.  56
          0.  57
          0.  58
          0.  59
```

Print code registers.

```
DDATE
      101.1975
CDATE
      715.1981
-DAYS
         2387.
CDATE
      715.1981
MDATE
     1231.1985
+DAYS
         1630.
PRIN.
       9000.00
% NOW
         16.50
HOLD?
SUM 1
      25343.14
P.VAL
      18711.49
PNLTY
        742.50
PRIN.
      17968.99
FUT.%
         17.50
SUM 2
      32011.88
+ / -
       6668.75
=%6MO
          0.82
ALT.%
         20.50
=DAYS
          731.
```

Sample output.

BATTERY POWER IN AN EMERGENCY, Bob Fruit.

I was stuck. I had a real need to use my TI-59, the battery packs were dead, and I couldn't get the wall recharger for several weeks. The problem I was looking at would probably require 20 hours of run time. If I had enough battery packs (which I don't) I could have restarted the calculator everytime it ran out of power. It would be inconvenient and would have required the wall recharger to charge up the battery packs in the first place.

The solution I found was to get a 6 volt lantern battery to power the calculator. I determined the polarity of the battery pack (the positive terminal is the one nearest to the module access door) and attached the battery to the calculator using alligator clips. The calculator ran fine without one puff of smoke.

Before considering to use the 6 volt lantern battery, I estimated the calculator runs on 4 volts from the battery pack, and estimated 4.5 volts at the battery pack terminals when using the wall recharger. This puts the lantern battery only 1.5 volts over the range normally experenced by the calculator. Because of my needs at the time, I decided to try it and see what would happen. As I have already said it worked great.

There is an added benefit. That is the amount of power in a 6 volt lantern battery. It is rated to have more then 55 watt hours of power. Compare that to the 2 watt hours in the TI supplied battery packs (estimated using normal values for ni-cad batteries).

Before I try this solution again I would be interested in hearing from people who are more hardware knowledgeable than I am. How close did I come to exceeding the calculators capacity to with stand the applied voltage? Could the excess power put too many amps through a critical component? I didn't use the card reader or use a module.

All I can say is this solution got me through a tight spot without any apparent trouble.

ACTIVE FILTER DESIGN HANDBOOK,- For use with programmable pocket calculators and mini-computers, George S. Moschytz and Peter Horn, both of the Swiss Federal Institute of Technology, Switzerland, 316 pp, 1981, $ 45.00, Wiley-Interscience, a division of Wiley & Sons, 1 Wiley Drive, Somerset, NJ 08873, USA.

This handbook provides specialists and non-specialist alike with all the necessary information for designing over twenty low-cost, well-proven active filters usable in most practical applications. It uses readily accesible equations, flow charts and computer print-outs in Basic, Fortran and TI-59 language. Detailed flowcharts permit the design programs to be rewritten in any other program language, such as APL or Pascal.

PRINTER HINTS, Bob Ericson casts another vote for a democratic printer carrying case. The one he uses is a $ 9.00 black Woolworths overnight case, with 1/8" foam liners added from a packing case. The inside hold-down straps have Velcro (R) in place of slider buckles.

Bob advices also to friction-fit the paper-roller axle with a plastic disk on the right side, so that the axle will dislodge less frequently when you carry the printer around a lot.

A LOOK AT THE DECIMAL TO FRACTION CONVERSION ROUTINE, Bob Fruit.

John worthington's and Emil Regelman's Decimal to Fraction Conversion Routine (V6N2P14) caught my attention. It appered to be an example of a heigher mathematics cannon being used to solve a simple problem. After spending the time to crack the algorithm used it no longer appears to be a case of heigher mathematics overkill.

To find out if the technique used was really necessary I compared the results from John and Emil's program with the results from what appeared to be the obvious solution to this problem. That is to take the interger portion of the decimal X 10^N, N being the number of decimal places wanted, over 10^N. Then use Euclid's algorithm to find the greatest common divisor (GCD) of the two numbers, and reduce the fraction to its lowest terms using the GCD.

```
00:R/S STO 01 R/S FIX 0 INV LOG EE INV EE STO 04 STO 02 X
16:RCL 01 = EE INV EE INT STO 03 STO 05 CP INV FIX RCL 04
32:- ( CE DIV RCL 03 STO 04 ) INT X RCL 03 = STO 03 INV X=T
50:0 30 RCL 05 DIV RCL 04 = X:T RCL 02 DIV rcl 04 = RST
```

Above is my quick and dirty program to try my alternative method. It is only 30% as long as is John and Emil's. Of course their program has bells and whistles to make it easier to use. I used the value of π for my experiments.

Analyzing the results shown here does say John and Emil's algorithm gives more pleasing results. The GCD method is fast and it does have a certain effiency at reproducing the given decimal exactly. Even though John and Emil's program is a lot slower I find I like its fractions a lot more.

10^N	GCD	J+E
2	157/50	355/113
3	1571/500	355/113
4	3927/1250	355/113
5	314159/100000	355/113
6	3141593/1000000	355/113
7	31415927/10000000	75948/24175
8	62831853/20000000	100798/32085
9	1570796327/500000000	103993/33102

A PROGRAM FOR LASER MATERIALS PROCESSING, *Robert T. Pitlak, Inrad, Northvale, NJ. Electro-Optical System Design, August 1981, pp 82-85. The program provides a quick estimation of the laser performance characteristics required for melting (i.e. welding) or vaporization (drilling) various materials.*

It is written for the TI-59 with the PC100 but, as it contains only 342 steps, it might be used on the TI-58 as well.

CALCULATE TIME RESPONSE FROM A CIRCUIT'S POLES AND ZEROS, *Colin Gyles, Data Precision Corporation, Danvers Industrial Park, Danvers, MA. 01923, Electronic Design, August 20, 1981, pp 147-149.*

The transfer function of a circuit can be expressed by its zeros and poles in the Laplacian or "s: domain. The "s to t program" for a TI-59 programmable calculator will provide the circuit's response to an impulse function in the time domain.

The author had the forsight to make the register format for the poles and the zeros deliberately identical to the one used by Javier Vizcaino in the PPX program # 658049, Open-loop Bode Diagram, so that the circuit's response, as well as the time response, could be conveniently obtained once a circuit's poles and zeros were determined.

NEWCOMER"S CORNER.- Michael Shanok.- (Mike is one of the four TI-59 seminar teachers.
----------------- Yours truly is another one. ED)

One of the simplest, yet least understood features of TI calculators is the Algebraic Operating System (AOS). A full mastery of its calculating hierarchy is often missed by experienced users, because it was at the beginning of the book, mixed in with all the "simple stuff" and therefore not studied carefully.

To begin with, algebraic hierarchy is a universal mathematical convention which we all unwittingly learned in algebra class. Because algebra seemed scary enough at the time, (and perhaps because the teacher didn't know either) we were not told that we were learning algebraic hierarchy.It is a "pecking order", establishing which functions are to be computed first.

At the top of the list are <u>single variable functions</u>, such as x^2, $\sqrt{x}$, 1/x, trig and log functions. Notice that all of these are immediately solved by the calculator, as they are called for, and are never left as pending operations.

Next are <u>double variable functions</u>, the powers and roots (y^x and $\sqrt[x]{y}$). Their solution awaits the pressing of an equal sign or of the closed parenthesis key, or the execution of any lower hierarchy function, rather than being performed immediately.

As an example, if we wish to solve for 3.5^2 raised to the $\sqrt{2}$ power, X 5, we would enter 3.5, press the x^2 key followed by the y^x key, then enter 2 and press the $\sqrt{x}$ key. As we then pressed the X key, the calculator would solve the expression (34.58..), because the y^x function had a hierarchy higher than that of the multiplication function. Notice that no parenthesis whatsoever are required.

The next lower hierarchy level is that of the <u>multiplication</u> and <u>division functions</u>. Below these are the <u>addition</u> and <u>subtraction functions</u>. Note that the addition of a list of "quantity X cost figures" requires no parenthesis. Each time a quantity is multiplied by a cost, pressing the + sign makes the calculator "look back" for a pending operation with a higher hierarchy, find the multiplication problem and solve it before proceding with the lower hierarchy addition problem. As an example, 15 X 7.5 + 11 X 3.6 + 2 X 7.8 = 167.7 needs no parenthesis for its solution. As a further exercise, note that the expression 2 + 8 X 4 is solved by letting the + sit as a pending operation, while 8 X 4 = 32 is first solved as a higher hierarchy operation, resulting in an answer of 34 when the = key is pressed.

Once a pending upper hierarchy operation is solved, the calculator continues to solve the simplified expression <u>from left to right</u>. This is the lowest level of hierarchy operation.

<u>Parenthesis</u> are the calculator's tools for intentionally blocking the AOS hierarchy. The skilled programmer uses them for no other purpose and seldom, if ever, uses them where they are not necessary. For instance, they would be employed if the user wanted to know the "breakout" of each quantity X the price in the previous example, by nesting each quantity X price between parenthesis. Nota that, when one expression is nested within another, the calculator solves the innermost nests are their closed parenthesis are encountered.

Finally, three important rules about parenthesis:

1. The calculator does not assume the convention of two concurrent nests of parenthesis to call for multiplication. For example, for (a+b)(c+d) to denote multiplication, it must be entered as (a+b)X(c+d).
2. The calculator assumes all expressions to start with an unlimited number of open parenthesis. As an example, the open parenthesis before a, in the previous example, could have been omitted.
3. An equal sign automatically closes all pending parenthesis. Therefore, the closed parenthesis after the d, in the previous expression, if followed by an = sign, could also have been omitted. Note that in modules, otherwise open and closed parenthesis have been included, so that the user may emply them as subroutines.

Next time you see an expression, such as ((((((14)$\sqrt{x}$) y^x RCL 01) X 5) + 7 ln)))= , just smile knowingly.

See you in October,