* TI PPC NOTES * V6N1, 1981

****

NEWSLETTER OF THE TI PROGRAMMABLE CALCULATOR CLUB.

9213 Lanham Severn Road, Lanham MD, 20801, USA

Welcome back to the TI PPC NOTES. With the appearance of our first issue in 1981 we begin our second year of existence, the sixth year if you take in account the SR-52-Notes. Somewhere in this issue you'll find how you can get hold of the back issues of SR-52-Notes. This in response to many inquiries.

A short re-statement of the goals and purpose of the TI PPC NOTES: To disseminate member-generated software, programming aids and tricks, to report on hardware modifications, new hardware, to tell you about books, programs and other material related to programmable calculators that has been published, to report on news about our beloved machines. All this with respect to algebraic or AOS system machines, such as the TI-58/59, TI-57, EC-4000, SR-52, SR-56, TI-55..... Because the great majority of our members own either a 58 or a 59, most of our software will be for those machines. But we will try to bring you material for the less known machines too, if members contribute it.

Every piece of software is reviewed by at least two independent reviewers, who check not only for validity of equations and formulas used, but also for adherence to proper programming techniques to ensure correct execution and friendliness toward the user.

Some of our members have been under the false impression that one of the goals of the TI PPC NOTES was to promote correct usage of Sheakespeare's beautiful language. It should be clear by now that I butcher our beloved English with unbridled abandon. But I will continue to make up for this flaw in my education by trying to write in an informal, easy-going style, devoid of all "obfuscation and blassé phrasology."

We will continue to be independent of Texas Instruments Inc, in spite of my personal contract with them as a teacher of TI-59 seminars in the TIPPP program. We agreed that I can and will wear two hats: One, as the editor of the TI PPC NOTES. I may laud or condemn TI products or practices, and two, as a TI contract teacher, I will promote everything TI. This to clarify my position vis-à-vis allusions by some of my friends to the fact that I was "maybe taken in by them." Just you wait, you scoffers. I'll fix your wagon in the pre-April 1-issue. It will creep up to you with stealth, like a thief in the night and you will not even be aware of it until it is too late. You are warned!

The FAST MODE keeps on to be practical and members are learning how to use it, in spite of its limitations. You will find again some very useful programs in this issue that employ the FAST MODE: three SFF programs, two random integers programs and the "star" contribution, a method to use up to three levels of subroutines in FAST MODE.

An add for a cigarette company shows an elegantly dressed woman smoking one of the company's brands. The caption reads:"You have come a long way, baby." This, rather sexist remark, applies equally well to the calculator-computer technology that evolved in our live time. In the November 24, 1980 issue of Future Computing Illustrated I saw a comparison between the (1955) IBM 650 computer and the modern TI-59 calculator. The IBM-650 had 2000 tubes, needed 17.7 KVA of power to operate , had a volume of 270 ft^3 and its weight was 5650 lbs. The "lowly" 59 has the equivalent of 166500 transistors, needs only 0.00018 KVA of power, occupies only 0.017 ft^3 of space and weighs a mere 0.67 lbs. The memory capacity is of the same order, while the 59 is about ten times faster. But the price of the IBM, in 1955 dollars was $ 200000, while the TI-59 costs not even $ 225 in 1980 dollars. Assuming similar technical advances, the power of an IBM 370/168 should be available in a hand-held computer by the year 2000.

Maurice E.T. Swinnen.

IN THIS ISSUE:

BOOKS ..... 2 -3
HAND-HELD CALCULATORS & THE AIR FORCE 3
THE TI-59 & SALES 3
49 ACCOUNTS, *Wm C. Carpenter* 4
SPEEDY FACTOR FINDERS, *Skillman, Hanson, Gustavsson* 5 -6
ASTRONOMY 6
3 LEVELS OF SBR IN FAST MODE, *Dave Leising* 7
SR-52 NOTES ..... 7 -8
100 RANDOM INTEGERS, *Richard Snow, Jeff Rosedale* .. 9-10
HYPERBOLIC FUNCTIONS, *Paul Wright* 11
ASTRONOMY 11
INTERPOLATION, *Fred Fullam* 12
OBLIQUE TRANSFORM, *Fred Fullam* 12
ORTHOGRAPHIC PROJECTION, *Fred Fullam* 13-14
REPARTITIONING THE TI-58C, *Patrick Acosta* 15-16
FILTER DESIGN 16
CODEC PROGRAM 16

BOOKS.- In v5n8p3 we told you about the CALCULATOR CALCULUS book by George McCarthy, and available through EduCalc Mail Store, 27963 Cabot Road, So.Laguna CA 92677, USA. That book, by the way, is their # E-45, and costs $ 14.95.
The same firm sells also the following calculator-related books:

POCKET PROGRAMMABLE CALCULATORS IN CHEMISTRY, by Barnes/Waring, Stock # E-44, 363 p, softbound, $ 16.95. Contains 29 programs, the equations used, instructions, register contents and a sample problem. Each program in both algebraic and RPN logic.

POCKET CALCULATOR SUPPLEMENT FOR CALCULUS, by Rosser/de Boor, Stock # E-32, 291 p, softbound, $ 11.50. Programs are given for the TI-57 and the HP-33E.

APPLIED MATHEMATICAL PHYSICS WITH POCKET CALCULATORS, by Robert Eisberg, Stock # E-22, 176 p, softbound, $ 14.50. Programs are given for the SR-56 and the HP-25.

COMBINATORIAL ALGORITHMS FOR COMPUTERS AND CALCULATORS, by Nijenhuis/Wilf, 302 p, hardbound, $ 25.95, Stock # E-25. Contains a ready-to-use collection of FORTRAN subroutines for the solution of combinatorial problems, such as "what is the minimum length of road connecting the 48 state capitals?" or "estimate the probability that a shuffled deck contains no straight." 62 algorithms in total.

TRIGONOMETRY USING CALCULATORS, by Elich/Elich, Stock # E-36, 359 p, hardbound, $ 17.95. A text book, suitable for self-study. Treats both algebraic and RPN logic scientific calculators.

FINANCIAL ANALYSIS USING CALCULATORS, by Greynolds/Aronofski/Frame, Stock # E-28, 427 p, softbound, $ 13.95. A book for the practicing professional. Works on a non-programmable machine but programmability is desirable nontheless. Many exercises and application problems.

CALCULATOR MATHEMATICS FOR THE REAL ESTATE PROFESSIONAL, by Lawrence Rosen, Stock # E-24, 223 p, hardbound, $ 19.95. A very practical book. Examples given for both algebraic and RPN machines.

THE HANDHELD CALCULATOR - USE AND APPLICATION, by Hyatt/Feldmnan, Stock # E-43, 256 p, softbound, $ 14.95. Introductory text for both algebraic and RPN logic entry.

SCIENTIFIC ANALYSIS ON THE POCKET CALCULATOR, by Jon Smith, Stock # E-33, 450 p, hardbound, $ 18.95. For both RPN and algebraic logic machines. I am most impressed with this book, that deals with evaluation of functions, Fourier analysis, numerical integration, linear system simulation, Chebyshev and rational approximation, roots, statistics and probability, matrix analysis, complex numbers and functions and financial calculations. The book is from the famous Wiley-Interscience series.

COUNTDOWN: SKYDIVER, ROCKET AND SATELLITE MOTION ON PROGRAMMABLE CALCULATORS, by Eisberg/Hyde, Stock # E-27, 107 p, softbound $ 7.95. May be used with almost any programmable calculator, even the least expensive ones.

HOW TO PROGRAM YOUR PROGRAMMABLE CALCULATOR, by Snover/Spikell, Stock # E-29, 271 p, softbound, $ 7.95. A practical problem solver. All the programs are given for the TI-57 or EC-4000 (Radio Shack, a twin to the TI-57) Covers evaluation functions, looping and decision making. Over 160 examples and exercises with solutions for business, finance, science, math and gambling. Contains also programs for the HP-33E.

TAKE A CHANCE WITH YOUR CALCULATOR - PROBABILITY PROBLEMS FOR PROGRAMMABLE CALCULATORS, by Lennart Rade, Stock # E-39, 163 p, softbound, $ 9.95. The programs are given for the HP-25, but the text is applicable to any calculator. Explains all the aspects of probability in a clear fashion.

COMPUTATIONAL ANALYSIS WITH THE HP-25 POCKET CALCULATOR, by Peter Henrici, Stock # E-26, 280 p, softbound, $ 17.95. Although intended for the HP-25, each of the 30 programs are given with method, flow diagram, examples, storage, operating instructions and timing, such that they are easily adapted to other calculators.

(over)

ENERGY ANALYSIS WITH A POCKET CALCULATOR, by G.A. Patterson, Stock # E-70, 107 p, softbound, $ 9.95. Basic Science Press.

ENGINE THERMODYNAMICS WITH A POCKET CALCULATOR, by G.A. Patterson, Stock # E-71, 120 p, sofbound, $ 13.95.

BASIC FLUID SYSTEMS ANALYSIS, by G.A. Patterson, Stock # E-72, 85 p, softbound, $ 12.95. The above three books for programmable calculators.

ARITHMETIC AND CALCULATORS, by Chinn/Dean/Tracewell, Stock # E-23, 488 p, softbound, $ 11.95. This book is, in my opinion, the ideal book for those who feel they are deficient in math and who sometimes have trouble understanding what we are talking about in the TI PPC NOTES. It is intended to be used with any type of algebraic-entry calculator. It has many practical examples, covers mathematics from the ground up, and teaches real-world applications. I highly recommend it.

PROGRAMMING PROGRAMMABLE CALCULATORS, by Harold Engelsohn, Stock # 74, 211 p, softbound, $ 11.95. Written especially for those who have trouble understanding the TI manual. Intended for the SR-52, SR-56, TI-57, TI-58, TI-59, EC-4000, PR-100 (Commodore), and for the APF programmable. It gives plenty of detail, examples and flow charts.

PROGRAMMABLE POCKET CALCULATORS, by Mullish/Kochan, Stock # 75, 254 p, softbound, $ 9.95. This excellent book is intended only for programmables that use RPN logic: Novus Math PR, Sinclair Sci. Pro., HP 25, 25C, 55, 65, 19C, 29C and 33E.

CELESTIAL NAVIGATION WITH A POCKET CALCULATOR, by Jonah Slocum, Stock # 73, 116 p, softbound, $ 14.95. For yachtsmen, pilots, navigators. Usable for both RPN and algebraic-entry calculators. Teaches astronomy, plane navigation, spherical navigation, time measurement and astronomical triangles. Many examples and programs in both RPN and algebraic.

In all the above descriptions, the word ALGEBRAIC is used to denote the AOS system as used in the TI calculators.

PLEASE DO NOT WRITE TO ME, THE EDITOR, IF YOU WANT TO PURCHASE THESE BOOKS. WRITE TO THE EDUCALC MAIL STORE, 27963 CABOT ROAD, SOUTH LAGUNA CA, 92677, USA.

If you order any books, it never hurts mentioning the TI PPC NOTES as the place where you found out about those books.

HAND-HELD CALCULATORS AND THE AIR FORCE.- The Dec 4, 1980, issue of Electronics, page 34, runs an article about the Air Force wanting to purchase 4,000 hand-held calculators to be used instead of the on-board fuel management computers. The Air Force hopes to save the tidy sum of $ 40 million dollars in fuel this way. What they are looking for are modified programmable printing calculators with aircraft characteristics encoded in read-only memory which can accept real-time data. The machine would then print out altitude, air speed, and engine management for minimum fuel use. For those familiar with airplane types, the numbers given are:the C-141, the C-5A, the B-52 and the KC-135. The purchase of the printing calculators is expected to cost around $ 1.3 million. The whole thing is expected to pay for itself in less than a month. Seems to me that either a TI-59 with a special module or an HP-41C could fill the bill perfectly. The Air Force expects delivery by August 1981.

THE TI-59 AND SALES.- My friend Ray Cummins, in Beaufort, N.C., brings to my attention a sales program produced by RAB, Radio Advertising Bureau, Inc., 484 Lexington Ave., New York, N.Y. 10027, tel.: (212) 599-6666. This special program, on three magnetic cards comes with a nice instruction manual and is intended to analyze the "reach" and "frequency" of newspapers, alone or in combination with a radio-broadcast station. It shows the user how to improve the efficiency of a newspaper-only campaign. The SONAR (for Synthesis of Newspapers and Radio) system costs $ 350.00 and is available from the address above.

---

49 ACCOUNTS.- Wm C. Carpenter, Bakersfield CA, is the author of this practical program. It allows you to keep inventory on 49 different items, accounts, what have you. In its present form it permits you to record the item number, a descriptor, the date and an amount. Of course, you may modify the program yourself, to accomodate it to your particular needs.

The program and the stored data may be recorded on two mag cards. That means, you need two mag cards for each list of up to 49 items, accounts....

On top of that, you may store a "master" name, that is the name of that particular list. Pressing of one user-defined key will recall and print the entire list of accounts and all its data. Zero containing accounts are skipped.

Instructions:

Key in the program in partitioning 6 OP 17 and record on side 1 of a mag card. Keep that card as a "master."

To use the program:

1. Load the master card, bank 1.
2. Enter a one- or two-digit account number and press A. Account numbers may range from 1 through 49.
3. Enter a one- to ten-digit numeric OR the code for up to four letters or characters. This is intended to be the descriptor. Press B.
4. Enter the date in the form MMDD.YY and press D.
5. Enter a value (money) up to 99,999.99. Press C. Errors are corrected by repeating the step before entering another account number. CAUTION: dates and values occupying too many places may interfere by cross sums or printing erratic information, as they share the same place register.
6. To enter the file name, enter 50, press A, enter the code for a four-letter descriptor and press B.
7. To obtain the entire list, press E. If you want the master list name included, press 2 E. Items 2, 3, 4 and 5 may be repeated as often as needed, up to 49 times per two mag cards.
8. Record the entire list, program and data, on two mag cards.

```
          50.      ?
30173736.          METS
       0.00      ?DATE
       0.00      ? AMT

          49.
  131415.          ABC
  113.81         DATE
    9.75           AMT

           3.
  151436.          CBS
  112.81         DATE
   15.12           AMT

           2.
  391415.          NBC
  111.81         DATE
  112.69           AMT

           1.
  241428.          IBM
  110.81         DATE
   67.25           AMT
       0.          AMT
```

If it is required to recall part of an entire list, STF 0 from the keyboard, enter the account number and press A, B, C and D. Use the RST key to restore to normal, that is automatic, operation.

The print out listing is complete when 0.AMT is printed.

```
000 76 LBL
001 99 PRT
002 69 OP
003 06  06
004 22 INV
005 58 FIX
006 06  6
007 69 OP
008 17 ,17
009 25 CLR
010 92 RTN
011 76 LBL
012 16 A'
013 32 X:T
014 01  1
015 00  0
016 69 OP
017 17  17
018 92 RTN
019 76 LBL
020 14  D
021 16 A'
022 01  1

023 06  6   D
024 01  1    A
025 03  3    T
026 03  3    E
027 07  7
028 01  1
029 07  7
030 69 OP
031 04  04
032 58 FIX
033 02  02
034 87 IFF
035 00  00
036 00  00
037 50  50
038 73 RC+
039 00  00
040 52 EE
041 85  +
042 32 X:T
043 55  ÷
044 01  1
045 52 EE

046 06  6
047 95  =
048 72 ST+
049 00  00
050 25 CLR
051 73 RC+
052 00  00
053 75  -
054 52 EE
055 95  =
056 65  ×
057 01  1
058 52 EE
059 06  6
060 95  =
061 22 INV
062 52 EE
063 61 GTO
064 99 PRT
065 76 LBL
066 13  C
067 16 A'
068 01  1

069 03  3
070 03  3
071 00  0
072 03  3
073 07  7
074 69 OP
075 04  04
076 58 FIX
077 02  02
078 87 IFF
079 00  00
080 00  00
081 91  91
082 73 RC+
083 00  00
084 75  -
085 52 EE
086 85  +
087 32 X:T
088 95  =
089 72 ST+
090 00  00
091 25 CLR

092 73 RC+
093 00  00
094 61 GTO
095 99 PRT
096 76 LBL
097 12  B
098 16 A'
099 69 OP
100 30  30
101 87 IFF
102 00  00
103 01  01
104 08  08
105 32 X:T
106 72 ST+
107 00  00
108 73 RC+
109 00  00
110 69 OP
111 04  04
112 69 OP
113 20  20
114 61 GTO

115 99 PRT
116 76 LBL
117 11  A
118 69 OP
119 00  00
120 42 STD
121 00  00
122 44 SUM
123 00  00
124 98 ADV
125 61 GTO
126 99 PRT
127 76 LBL
128 15  E
129 85  +
130 09  9
131 08  8
132 95  =
133 42 STD
134 00  00
135 86 STF
136 00  00
137 16 A'.

138 29 CP
139 73 RC*
140 00  00
141 67  EQ
142 01  01
143 53  53
144 43 RCL
145 00  00
146 55  ÷
147 02  2
148 95  =
149 11  A
150 12  B
151 14  D
152 13  C
153 69 OP
154 30  30
155 97 DSZ
156 00  00
157 01  01
158 37  37
159 81 RST
```

SPEEDY FACTOR FINDERS.- Robert Caldwell's program on v5n9-10p9 will not run as shown.
---------- That was due to a typo. (mea culpa, mea maxima culpa)
Steps 161 through 168 need to be changed as follows:

161: 1 0 D 2 D 1 0 D

Robert's program also has the same quirk that Bill Skillman found in a former Louder/ Vanderburgh program, published in 52-Notes: It prints an unwanted "1", following the print out of the highest prime factor, but only some of the time. It does it with 103569859 but not with 987654321.(see 52-Notes v3n11p4)

Bill Skillman rewrote his speedy factor finder (SFF) in Fast Mode. (he finally made the plunge) The results are 1 min 29 sec and 1 min 8 sec for both test numbers respectively. As you might recall, the former record in v5n9-10p8 was 1 min 33sec and 1 min 11 sec by Bjorn Gustavsson.

The instructions for Bill's program, the first of the series, are:
Read card, side 1 and press A. Insert card again, press CE and enter the number. Press R/S and see the number and its factors printed. If hand-held, R/S each factor and CE. In both cases, printer and hand-held, the last factor is flashed.

Note the peculiar sequence in steps 034 through 037. Bill says it is faster than EQ followed by GTO.Bill uses OP 69 to produce a non-halting flasher.

I keep reporting them in chronological order, folks. That is, the time of arrival counts. The next one is from Palmer O. Hanson. In fact, I received two from Palmer: one on December 7 and another, and faster one, on December 18, 1980. Palmer's program executes at an "incredible" speed: 1 min 17 sec and 59 sec respectively. And it may be worked both in Fast Mode and in Normal Mode. (This is case you get sick from all that "speed freaking").His instructions are: Fast mode:

1. Load side 1. Press RST R/S to get into Fast Mode.
2. Reload side 1, followed by side 2. DISPLAY WILL FLASH. IGNORE IT.
3. Enter number and press R/S. Display stops with a flashing 1.
   Factors are printed with a question mark.
4. To recover number and prime factors when no printer is available, press CLR SBR 309.
   Displays number. Press R/S. Displays first factor. Press R/S.....
5. To test another integer after using steps 3 and 4: Enter number and press R/S.

Palmer's program is the famous Louder/Vanderburgh program from v3n11p4 in Fast Mode.

Normal Mode:

1. Load sides 1 and 2 of the mag card.
2. Enter number and press A. See number and prime factors printed. Stops with flashing "1" in the display.
3. Read out the integer and primes if no printer available: Press B, displays number. Press R/S to see first prime, press R/S again.....Flashing "1" means END.

The next one to arrive is by our mathematician in Sweden: Björn Gustavsson. His program runs neck on neck with Palmer's with respect to speed: 1 min 16 sec and 59 sec again. Bjorn's technique is completely different from any program I have seen so far. It requieres a very short program, but data to be loaded in registers 01 through 52. Watch out for his sequences, such as DSZ 56 082. Key in as: DSZ STO 56 GTO 082, and delete the STO and the GTO commands.

His instructions are, for Fast Mode only:

1. Load bank 1. Press A to get into Fast Mode.
2. Load bank 1 again. A "3" will be displayed. Enter bank 3; a "4" will be displayed. Enter bank 4. Paper advances.
3. Enter the number (integer) and press R/S. Integer and prime factors printed.

This should do it for now with respect to SFF programs. But it is by no means the end of this chapter. Faster SFF programs will arrive, using new algorithms. Palmer Hanson has been dilligently at work over the holidays, he writes me. I'll let you know the results in next issue.

:PROGRAMS ON NEXT PAGE:

000	00	0
001	00	0
002	00	0
003	76	LBL
004	11	A
005	36	PGM
006	02	02
007	71	SBR
008	02	02
009	39	39
010	09	9
011	00	0
012	00	0
013	25	CLR
014	22	INV
015	58	FIX
016	61	GTO
017	01	01
018	25	25
019	00	0
020	00	0

021	82	HIR
022	36	36
023	04	4
024	44	SUM
025	03	03
026	82	HIR
027	15	15
028	55	÷
029	82	HIR
030	16	16
031	95	=
032	22	INV
033	59	INT
034	22	INV
035	67	EQ
036	40	IND
037	03	03
038	82	HIR
039	15	15
040	32	X:T
041	82	HIR

042	16	16
043	82	HIR
044	65	×
045	67	EQ
046	01	01
047	21	21
048	99	PRT
049	29	CP
050	87	IFF
051	07	07
052	00	00
053	26	26
054	52	EE
055	22	INV
056	52	EE
057	91	R/S
058	61	GTO
059	00	00
060	26	26
061	01	1
062	61	GTO

063	00	00
064	21	21
065	02	2
066	61	GTO
067	00	00
068	21	21
069	02	2
070	61	GTO
071	00	00
072	21	21
073	04	4
074	61	GTO
075	00	00
076	21	21
077	02	2
078	61	GTO
079	00	00
080	21	21
081	04	4
082	61	GTO
083	00	00
084	21	21

085	02	2
086	61	GTO
087	00	00
088	21	21
089	04	4
090	61	GTO
091	00	00
092	21	21
093	06	6
094	61	GTO
095	00	00
096	21	21
097	02	2
098	61	GTO
099	00	00
100	21	21
101	06	6
102	61	GTO
103	00	00
104	21	21
105	82	HIR
106	15	15

107	34	fX
108	75	-
109	07	7
110	03	3
111	42	STO
112	03	03
113	82	HIR
114	16	16
115	95	=
116	77	GE
117	00	00
118	73	73
119	82	HIR
120	15	15
121	99	PRT
122	69	DP
123	69	69
124	98	ADV
125	52	EE
126	22	INV
127	52	EE
128	91	R/S

129	76	LBL
130	15	E
131	29	CP
132	82	HIR
133	05	05
134	99	PRT
135	98	ADV
136	06	6
137	01	1
138	42	STO
139	03	03
140	69	DP
141	07	07
142	69	DP
143	19	19
144	25	CLR
145	02	2
146	82	HIR
147	06	06
148	61	GTO
149	00	00
150	26	26

000	00	0
001	00	0
002	00	0
003	00	0
004	00	0
005	36	PGM
006	02	02
007	71	SBR
008	02	02
009	39	39
010	09	9
011	00	0
012	22	INV
013	58	FIX
014	22	INV
015	57	ENG
016	01	1
017	99	PRT
018	25	CLR
019	91	R/S
020	99	PRT
021	98	ADV
022	61	GTO
023	03	03
024	25	25
025	43	RCL
026	00	00
027	65	×
028	01	1
029	06	6
030	85	+
031	07	7
032	01	1
033	95	=
034	42	STO
035	03	03
036	43	RCL
037	01	01
038	55	÷
039	43	RCL
040	02	02
041	99	PRT
042	72	ST*
043	04	04
044	69	DP
045	24	24
046	95	=
047	42	STO
048	01	01
049	83	GD+
050	03	03
051	66	PAU
052	91	R/S
053	76	LBL
054	11	A
055	98	ADV

056	29	CP
057	47	CMS
058	42	STO
059	01	01
060	42	STO
061	09	09
062	99	PRT
063	01	1
064	00	0
065	42	STO
066	04	04
067	98	ADV
068	02	2
069	42	STO
070	02	02
071	43	RCL
072	01	01
073	55	÷
074	43	RCL
075	02	02
076	95	=
077	22	INV
078	59	INT
079	67	EQ
080	00	00
081	25	25
082	69	DP
083	20	20
084	03	3
085	42	STO
086	02	02
087	43	RCL
088	01	01
089	55	÷
090	43	RCL
091	02	02
092	95	=
093	22	INV
094	59	INT
095	67	EQ
096	00	00
097	25	25
098	69	DP
099	20	20
100	05	5
101	42	STO
102	02	02
103	43	RCL
104	01	01
105	55	÷
106	43	RCL
107	02	02
108	95	=
109	22	INV

110	59	INT
111	67	EQ
112	00	00
113	25	25
114	69	DP
115	20	20
116	07	7
117	42	STO
118	02	02
119	43	RCL
120	01	01
121	55	÷
122	43	RCL
123	02	02
124	95	=
125	22	INV
126	59	INT
127	67	EQ
128	00	00
129	25	25
130	69	DP
131	20	20
132	04	4
133	44	SUM
134	02	02
135	43	RCL
136	01	01
137	55	÷
138	43	RCL
139	02	02
140	95	=
141	22	INV
142	59	INT
143	67	EQ
144	00	00
145	25	25
146	69	DP
147	20	20
148	02	2
149	44	SUM
150	02	02
151	43	RCL
152	01	01
153	55	÷
154	43	RCL
155	02	02
156	95	=
157	22	INV
158	59	INT
159	67	EQ
160	00	00
161	25	25
162	69	DP
163	20	20
164	04	4

165	44	SUM
166	02	02
167	43	RCL
168	01	01
169	55	÷
170	43	RCL
171	02	02
172	95	=
173	22	INV
174	59	INT
175	67	EQ
176	00	00
177	25	25
178	69	DP
179	20	20
180	02	2
181	44	SUM
182	02	02
183	43	RCL
184	01	01
185	55	÷
186	43	RCL
187	02	02
188	95	=
189	22	INV
190	59	INT
191	67	EQ
192	00	00
193	25	25
194	69	DP
195	20	20
196	04	4
197	44	SUM
198	02	02
199	43	RCL
200	01	01
201	55	÷
202	43	RCL
203	02	02
204	95	=
205	22	INV
206	59	INT
207	67	EQ
208	00	00
209	25	25
210	69	DP
211	20	20
212	06	6
213	44	SUM
214	02	02
215	43	RCL
216	01	01
217	55	÷
218	43	RCL
219	02	02

220	95	=
221	22	INV
222	59	INT
223	67	EQ
224	00	00
225	25	25
226	69	DP
227	20	20
228	02	2
229	44	SUM
230	02	02
231	43	RCL
232	01	01
233	55	÷
234	43	RCL
235	02	02
236	95	=
237	22	INV
238	59	INT
239	67	EQ
240	00	00
241	25	25
242	69	DP
243	20	20
244	06	6
245	44	SUM
246	02	02
247	43	RCL
248	01	01
249	55	÷
250	43	RCL
251	02	02
252	95	=
253	22	INV
254	59	INT
255	67	EQ
256	00	00
257	25	25
258	69	DP
259	20	20
260	04	4
261	44	SUM
262	02	02
263	43	RCL
264	01	01
265	55	÷
266	43	RCL
267	02	02
268	95	=
269	22	INV
270	59	INT
271	67	EQ
272	00	00
273	25	25
274	43	RCL

275	01	01
276	34	fX
277	32	X:T
278	43	RCL
279	02	02
280	77	GE
281	02	02
282	90	90
283	05	5
284	42	STO
285	00	00
286	29	CP
287	61	GTO
288	01	01
289	48	48
290	01	1
291	32	X:T
292	43	RCL
293	01	01
294	67	EQ
295	03	03
296	00	00
297	99	PRT
298	72	ST*
299	04	04
300	32	X:T
301	69	DP
302	69	69
303	98	ADV
304	61	GTO
305	00	00
306	51	51
307	76	LBL
308	12	B
309	29	CP
310	09	9
311	42	STO
312	00	00
313	73	RC*
314	00	00
315	67	EQ
316	03	03
317	25	25
318	66	PAU
319	91	R/S
320	69	DP
321	20	20
322	61	GTO
323	03	03
324	13	13
325	01	1
326	69	DP
327	69	69
328	61	GTO

329	00	00
330	51	51
331	76	LBL
332	15	E
333	42	STO
334	07	07
335	66	PAU
336	91	R/S
337	42	STO
338	08	08
339	32	X:T
340	69	DP
341	00	00
342	05	5
343	00	0
344	00	0
345	00	0
346	69	DP
347	01	01
348	02	2
349	00	0
350	01	1
351	00	0
352	01	1
353	00	0
354	00	0
355	04	4
356	07	7
357	69	DP
358	02	02
359	43	RCL
360	07	07
361	99	PRT
362	65	×
363	01	1
364	00	0
365	00	0
366	85	+
367	69	DP
368	05	05
369	69	DP
370	00	00
371	06	6
372	04	4
373	69	DP
374	02	02
375	32	X:T
376	99	PRT
377	95	=
378	69	DP
379	05	05
380	61	GTO
381	00	00
382	55	55

/SEE BJÖRN'S SFF PROGRAM ON PAGE 8./

ASTRONOMY.- Tom Wysmuller sends me a Navy Technology Transfer Sheet which can be of interest to the astronomers, amateur and otherwise, among our members:

BACKUP DOCUMENTATION PACKAGE FOR FACT SHEET ARTICLE "COMPUTER ALMANAC TURNS HAND CALCULATOR INTO NAVIGATIONAL TOOL." ALMANAC FOR COMPUTERS, 1980.

I inquired at the office of Dr. P. Kenneth Seidelman, who is the director for the Nautical Almanac Office of the U.S. Naval Observatory, Washington DC 20390. I was informed that the same office offers a complete copy of the nautical almanac at the bargain price of $ 5.00 US. Make your checks payable to the Naval Observatory. Use the same addresss as above. If you want to inquire by phone, use: (202) 254-4571.
The fact sheet # is 051003. It contains 16 pages: calendar 1980, navigational tables, astronomical tables, stellar tables.
I am willing to purchase the almanac for members who live outside the US and send it by means of any of the postal rates you deem adequate. Please send enough money to cover postage. I will return any excess.

THREE LEVELS OF SUBROUTINES IN FAST MODE ?- Dave Leising accomplished this "impossible" feat by working some special TI-59 calisthenics over the holidays. He calls it FAST MODE SYNTHETIC SUBROUTINE PROTOCOL.

Dave is quick to outline some associated penalties which one has to "endure":

1. Register 00, or any other of the user's choice, is reserved for use as the subroutine return register.
2. A subroutine call requires ten program steps.
3. The x-register is used upon subroutine call. It is, however, undisturbed during subroutine return.
4. Control program operations require 28 locations of program memory.

The panalties may or may not be outweighed by the advantages of having subroutine capability available in the Fast Mode. But subroutines have been missing dearly in Fast Mode programs. So, I think it is a very handy enhancement of this type of programming. It wont be long before we'll see it used in real programs, as opposed to this demonstration program Dave wrote.

To call a subroutine, place in the X-register a number of the form ABC.DEF, in which ABC is the absolute address of the start of the called subroutine and in which DEF is the absolute address of the return. This is immediately followed by a GTO PQR in which PQR is the absolute address of the Control Program, Call Routine.

Every subroutine is ended by a GTO STU in which STU is the absolute address of the Control Program, Return Routine.

The breakdown of the demonstration program is as follows:

000 through 029	Initialize Fast Mode and Load Cards.
030 through 044	Subroutine Control: Call Routine.
045 through 057	Subroutine Control: Return Routine.
058 through 148	Demonstration: Main Program.
149 through 225	Demonstration: Subroutine 1.
226 through 310	Demonstration: Subroutine 2.
311 through 359	Demonstration: Subroutine 3.

User instructions for the demonstration program are:

1. Load card side 1. Press RST R/S. You have acquired Fast Mode.
2. Load card sides 1 and 2. Pres R/S twice to accomplish dummy reads for the non-existent card sides 3 and 4.
3. Demo program will execute, printing out an indented Structured-Program format dynamic illustration of its own operation.
4. At completion, press R/S to repeat execution. Or press RST GTO 058 TRACE R/S to observe the details of operation.

The program operates by using four three-digit fields of register 00 to store absolute addresses. The address required at any given time is shifted into the integer portion of register 00, where it becomes active for indirect GTO.

SEE THE PROGRAM LISTING NEXT PAGE.

SR-52 NOTES.- Richard Vanderburgh, 9459 Taylorsville Road, Dayton OH, 45424, started the 52-Users Club in June 1976. He was the editor of the famous SR-52 Notes. Although the name is a little misleading, Richard dedicated most of his excellent newsletter to the TI-59 once it appeared on the market. Richard still has the back issues available. Since I refer repeatedly to the 52-Notes, many members have asked how they can obtain these back issues. Well, they are still only $ 1.00 US each.

The first 12 issues, volume 1 from June 76 to December 76, and volume 2 from January 77 to May 77, deal exclusively with the SR-52. But starting with the June issue of 77, the newsletter deals more and more with the TI-59. These are: Volume 2, June through December 77, 7 issues; volume 3, January through December 78, 12 issues: volume 4, January through March 79, 3 issues. A grand total of 22 issues that deal with the TI-59. This is a total of 130 typewritten pages for $ 22.00 US.

Please, do not write to me for these back issues of the 52-Notes. Write Richard instead at the above address and don't forget to include your check or money order.

Three levels of SBR, Dave Leising.

```
1.
2.
0.
0.

PROGRAM
 SUBROUTINE (I)
  SUBROUTINE (II)
   SUBROUTINE (III)
  RETURN
 RETURN
RETURN
HALT
```

```
000  00   0
001  00   0
002  00   0
003  00   0
004  00   0
005  36 PGM
006  02   02
007  71 SBR
008  02   02
009  39   39
010  09   9
011  00   0
012  22 INV
013  58 FIX
014  22 INV
015  57 ENG
016  01   1
017  99 PRT
018  25 CLR
019  91 R/S
020  99 PRT
021  25 CLR
022  91 R/S
023  99 PRT
024  25 CLR
025  91 R/S
026  99 PRT
027  61 GTD
028  00   00
029  58   58
030  48 EXC
031  00   00
032  22 INV
033  59 INT
034  55   ÷
035  03   3
036  22 INV
037  28 LOG
038  95   =
039  48 EXC
```

```
040  00   00
041  44 SUM
042  00   00
043  83 GD*
044  00   00
045  48 EXC
046  00   00
047  22 INV
048  59 INT
049  65   ×
050  03   3
051  22 INV
052  28 LOG
053  95   =
054  48 EXC
055  00   00
056  83 GD*
057  00   00
058  98 ADV
059  69 DP
060  00   00
061  03   3
062  03   3
063  03   3
064  05   5
065  03   3
066  02   2
067  02   2
068  02   2
069  03   3
070  05   5
071  69 DP
072  01   01
073  01   1
074  03   3
075  03   3
076  00   0
077  00   0
078  00   0
079  00   0
```

```
080  00   0
081  00   0
082  00   0
083  69 DP
084  02   02
085  69 DP
086  05   05
087  01   1
088  04   4
089  09   9
090  93   .
091  00   0
092  09   9
093  07   7
094  61 GTD
095  00   00
096  30   30
097  69 DP7
098  00   00
099  03   3
100  05   5
101  01   1
102  07   7
103  03   3
104  07   7
105  04   4
106  01   1
107  03   3
108  05   5
109  69 DP
110  01   01
111  03   3
112  01   1
113  00   0
114  00   0
115  00   0
116  00   0
117  00   0
118  00   0
119  00   0
120  00   0
121  69 DP
122  02   02
123  69 DP
124  05   05
125  69 DP
126  00   00
127  02   2
128  03   3
129  01   1
130  03   3
131  02   2
132  07   7
133  03   3
134  07   7
135  00   0
```

```
136  00   0
137  69 DP
138  01   01
139  69 DP
140  05   05
141  98 ADV
142  98 ADV
143  98 ADV
144  25 CLR
145  91 R/S
146  61 GTD
147  00   00
148  58   58
149  69 DP
150  00   00
151  03   3
152  06   6
153  04   4
154  01   1
155  01   1
156  04   4
157  03   3
158  05   5
159  69 DP
160  01   01
161  03   3
162  02   2
163  04   4
164  01   1
165  03   3
166  07   7
167  02   2
168  04   4
169  03   3
170  01   1
171  69 DP
172  02   02
173  01   1
174  07   7
175  00   0
176  00   0
177  05   5
178  05   5
179  02   2
180  04   4
181  05   5
182  06   6
183  69 DP
184  03   03
185  69 DP
186  05   05
187  02   2
188  02   2
189  06   6
190  93   .
191  01   1
```

```
192  09   9
193  07   7
194  61 GTD
195  00   00
196  30   30
197  69 DP
198  00   00
199  03   3
200  05   5
201  01   1
202  07   7
203  03   3
204  07   7
205  04   4
206  01   1
207  69 DP
208  01   01
209  03   3
210  05   5
211  03   3
212  01   1
213  00   0
214  00   0
215  00   0
216  00   0
217  00   0
218  00   0
219  69 DP
220  02   02
221  69 DP
222  05   05
223  61 GTD
224  00   00
225  45   45
226  69 DP
227  00   00
228  03   3
229  06   6
230  04   4
231  01   1
232  01   1
233  04   4
234  69 DP
235  01   01
236  03   3
237  05   5
238  03   3
239  02   2
240  04   4
241  01   1
242  03   3
243  07   7
244  02   2
245  04   4
246  69 DP
247  02   02
```

```
248  03   3
249  01   1
250  01   1
251  07   7
252  00   0
253  00   0
254  05   5
255  05   5
256  02   2
257  04   4
258  69 DP
259  03   03
260  02   2
261  04   4
262  05   5
263  06   6
264  00   0
265  00   0
266  00   0
267  00   0
268  00   0
269  00   0
270  69 DP
271  04   04
272  69 DP
273  05   05
274  03   3
275  01   1
276  01   1
277  93   .
278  02   2
279  08   8
280  04   4
281  61 GTD
282  00   00
283  30   30
284  69 DP
285  00   00
286  03   3
287  05   5
288  01   1
289  07   7
290  03   3
291  07   7
292  69 DP
293  01   01
294  04   4
295  01   1
296  03   3
297  05   5
298  03   3
299  01   1
300  00   0
301  00   0
302  00   0
303  00   0
```

```
304  69 DP
305  02   02
306  69 DP
307  05   05
308  61 GTD
309  00   00
310  45   45
311  69 DP
312  00   00
313  03   3
314  06   6
315  04   4
316  01   1
317  69 DP
318  01   01
319  01   1
320  04   4
321  03   3
322  05   5
323  03   3
324  02   2
325  04   4
326  01   1
327  03   3
328  07   7
329  69 DP
330  02   02
331  02   2
332  04   4
333  03   3
334  01   1
335  01   1
336  07   7
337  00   0
338  00   0
339  05   5
340  05   5
341  69 DP
342  03   03
343  02   2
344  04   4
345  02   2
346  04   4
347  02   2
348  04   4
349  05   5
350  06   6
351  00   0
352  00   0
353  69 DP
354  04   04
355  69 DP
356  05   05
357  61 GTD
358  00   00
359  45   45
```

SUBROUTINE (III)

SUBROUTINE (II)

SUBROUTINE (I)

SUB ROUTINE

RETURN

RETURN

RETURN

RETURN

PROGRAM

HALT

SFF program, Björn Gustavsson.

```
10.   01
 2.   02
10.   03
 2.   04
 4.   05
 2.   06
 4.   07
 6.   08
 2.   09
 6.   10
 4.   11
 2.   12
 4.   13
 6.   14
 6.   15
 2.   16
 6.   17
 4.   18
 2.   19
 6.   20
 4.   21
 6.   22
 8.   23
 4.   24
 2.   25
 4.   26
 2.   27
 4.   28
 8.   29
 6.   30
```

```
 4.   31
 6.   32
 2.   33
 4.   34
 6.   35
 2.   36
 6.   37
 6.   38
 4.   39
 2.   40
 4.   41
 6.   42
 2.   43
 6.   44
 4.   45
 2.   46
 4.   47
 2.   48
 4.   49
 2.   50
 2.   51
 1.   52

000  76 LBL
001  11   A
002  00   0
003  00   0
004  00   0
005  36 PGM
```

```
006  02   02
007  71 SBR
008  02   02
009  39   39
010  09   9
011  00   0
012  22 INV
013  58 FIX
014  22 INV
015  57 ENG
016  03   3
017  91 R/S
018  04   4
019  91 R/S
020  98 ADV
021  25 CLR
022  91 R/S
023  98 ADV
024  29 CP
025  42 STD
026  54   54
027  99 PRT
028  02   2
029  42 STD
030  55   55
031  05   5
032  02   2
033  42 STD
034  00   00
035  05   5
036  42 STD
```

```
037  58   58
038  98 ADV
039  43 RCL
040  54   54
041  55   ÷
042  43 RCL
043  55   55
044  95   =
045  22 INV
046  59 INT
047  67   EQ
048  01   01
049  56   56
050  73 RC*
051  00   00
052  69 DP
053  30   30
054  44 SUM
055  55   55
056  97 DSZ
057  58   58
058  00   00
059  39   39
060  43 RCL
061  54   54
062  34 FX
063  85   +
064  01   1
065  09   9
066  07   7
067  54   )
```

```
068  55   ÷
069  02   2
070  01   1
071  00   0
072  95   =
073  59 INT
074  67   EQ
075  01   01
076  09   09
077  94 +/-
078  42 STD
079  56   56
080  42 STD
081  57   57
082  43 RCL
083  54   54
084  55   ÷
085  43 RCL
086  55   55
087  95   =
088  22 INV
089  59 INT
090  67   EQ
091  01   01
092  16   16
093  73 RC*
094  00   00
095  44 SUM
096  55   55
097  97 DSZ
098  00   00
```

```
099  00   00
100  82   82
101  04   4
102  08   8
103  42 STD
104  00   00
105  97 DSZ
106  56   56
107  00   00
108  82   82
109  43 RCL
110  54   54
111  99 PRT
112  91 R/S
113  61 GTD
114  00   00
115  23   23
116  43 RCL
117  55   55
118  22 INV
119  49 PRD
120  54   54
121  99 PRT
122  43 RCL
123  56   56
124  75   -
125  43 RCL
126  57   57
127  75   -
128  53   (
129  53   (
130  43 RCL
131  54   54
```

```
132  34 FX
133  85   +
134  01   1
135  09   9
136  07   7
137  54   )
138  55   ÷
139  02   2
140  01   1
141  00   0
142  54   )
143  59 INT
144  95   =
145  22 INV
146  77   GE
147  00   00
148  78   78
149  43 RCL
150  54   54
151  99 PRT
152  91 R/S
153  61 GTD
154  00   00
155  23   23
156  43 RCL
157  55   55
158  22 INV
159  49 PRD
160  54   54
161  99 PRT
162  61 GTD
163  00   00
164  39   39
```

100 RANDOM INTEGERS.- In v5n8p1 I posed the problem of "generating at least 100 non-repetetive random integers, predetermined between an upper and a lower limit." I said that the problem "had me completely stomped." It had me stumped also, of course. Up to this writing I recieved two solutions: one by Richard Snow and another one by Jeff Rosedale. The first one is ultra-fast: it produces 100 integers in about 2 min 40 sec. The second one is slower: 100 integers take about 17 min 40 sec. But Jeff's program can handle up to 300 integers and with slight modifications could generate about 990 integers. Richard's program is limited to a maximum of 100 integers.

Several people questioned the practicallity of such a program. But, suppose you have 100 employees, as the original requestor has. And suppose you have to write shift schedules for those employees. This means, who will work the first shift, who the second, etc. You could write the name of each of the employees on a separate piece of paper and draw them from the hat. The first 25 would be assigned to the first shift, the next 25 to the second one, the following 25 to the third shift and the rest to the swing shift, that is the work shift that replaces the shift on regular days off. This is the way (or some variation of it) that has been used in most places. At regular intervals a new "random" list is drawn, to give people variation in their work.

Some enterprising outfit publishes little booklets with computer-generated lists of random numbers. By assigning a number to each employee and then opening the booklet at random, one can periodically generate a new random work assignment list.

I have seen once a TI-59 program to generate 100 random integers. But is was extremely slow, as after each randomly selected integer, it checked to see if that one had already been generated. If not it would be printed. If yes, it would generate a new random integer, check it again, and so on. A really endless task, which would become more tedious the longer the list of random integers was. Generating 100 integers was almost an overnight proposition.

In Richard's program a different approach has been employed. Richard credits it to an article he once saw in Personal Computing. Only the difference between each one of 100 integers is randomly produced. The range of random integers (max-min) is divided by the number of random integers left to be generated.(contents of register 00 in a DSZ loop) The result is fed to a random number generator making sure that at least a "1" is being generated. The small random number is subtracted from the range of random integers in HIR 03 but added to the previously computed random integer. This produces a list of increasing random integers which are stored in the t-register and in registers 01 to 99 in reverse order.

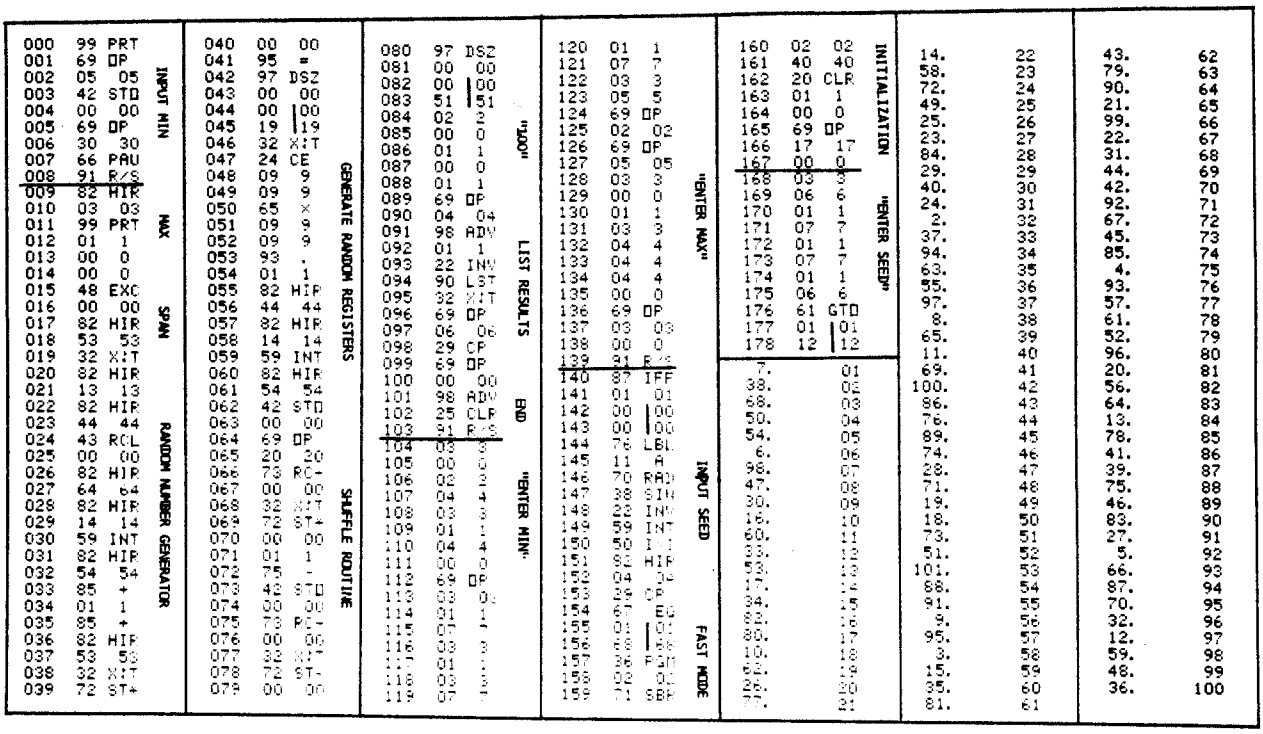
The next part of his program shuffles the contents of the registers in another DSZ loop. The contents of each register is exchanged with that of the t-register. Finally the shuffled contents of all the registers including the t-register are listed.

The FAST MODE was added to the program to speed up the process. Its speed increased from an original 5½ minutes to 2 min 35 sec. Prompting was added to make the program "friendlier-to-the-user."

User instructions:

1. *Key in the program and record it on one card side. Besides the usual difficulties of keying in HIRs(I hope everybody knows by now how to do it) watch out for step 162. This is not an ordinary CLR (code 25) but a 2nd CLR (code 20)*
2. *Enter a number between 0 and 10^{12} and press A. If that seed is not satisfactory, the printer will prompt you for a new seed, by printing ENTER SEED.*
3. *Insert card side 1 again.*
4. *ENTER MIN is printed. Enter the minimum random number to be generated and press R/S.*
5. *ENTER MAX is printed. Enter the maximum random number to be generated and press R/S. This number should be at least 100 more than the minimum entered. Both MIN and MAX can be negative values, if desired.*
6. *Wait almost 3 minutes for program execution, after which the printer will list the 100 integers with the number of the register they were stored in.*
7. *To produce a new list, press R/S and repeat steps 4 and 5.*

(over)



Jeff Rosedale calls his program NONREPRNDINTLIST, for Non-Repetitious Random Integer List. It can handle up to 300 integers. The lower limit of this list can be any number. All the integers in the list will be found between the lower limit you selected and the number of integers your requested, minus one. For example, if you enter 250 as the lower limit and 100 as the number of integers wanted, the range will be from 250 to 349, inclusive.

The method used is as follows: Registers 00 through 29 contain each ten ones, i.e. "1111111111". Each "1" represents the number between o and 9 (position near the decimal point) and its register # is multiplied by 10 and added to itself to obtain the numbers from 0 to 299. Each random number is assigned one of these numbers. When it is generated, it is checked against its number. If the "1" is there, it will be printed. If not, another random number will be generated. When it is printed, the corresponding "1" is deleted, to avoid it being printed again. This process continues until all the random integers are printed. This way all the memories become zeroed.

The program may be modified to handel up to 990 integers. All you have to do is to convert the working registers, R30 to R35, to HIRs. You will have to use R99 instead of R35, because of the IND function it provides. HIRs have no IND functions. Then you will have to store ones in registers R00 through R98.

The present program fits nicely on one mag card, sides 1 and 4. An extended list will need two mag cards.

Instructions:

Load registers 00 through 29 with the number "1111111111". That is, enter ten ones in the display and key: STO 01, STO 02.....until you are blue in the face, or whatever your particular skin color becomes after strenuous exercise. Or, you can do it the easy way by means of this short routine. Press LRN and put this program in user memory:

```
LBL A CMS  1 1 1 1 1 1 1 1 1 1  STO 30  LBL B  RCL 30  STO IND 31  1  SUM 31  30  X:T
RCL 31  X=T  C  GTO B  LBL C  R/S
```

Press LRN again to go out of the LRN mode and press A. The program will take about 20 sec to put ones in registers 00 through 29. It will stop with "30" in the display.
Now press LRN again and key in the program below, all 88 steps. Press LRN again.
Record the program on side 1 and the data on side 4 of one and the same mag card.
***To run the program**, that is to produce a list of random integers, do as follows:*
1. Enter the number of integers required and press A.
2. Enter the lower limit of the range and press B.
3. Enter a seed, a number between 0 and 1, and press C. Then have a little patience.

(over)

000 76 LBL	013 13 C	026 31 31	039 00 0	052 28 LOG	064 01 1	076 95 =
001 11 A	014 42 STO	027 95 =	040 75 -	053 59 INT	065 00 0	077 22 INV
002 42 STO	015 33 33	028 59 INT	041 59 INT	054 42 STO	066 95 =	078 74 SM*
003 31 31	016 43 RCL	029 42 STO	042 42 STO	055 36 36	067 59 INT	079 35 35
004 42 STO	017 33 33	030 34 34	043 35 35	056 35 1/X	068 67 EQ	080 43 RCL
005 30 30	018 22 INV	031 85 +	044 95 =	057 65 ×	069 00 00	081 34 34
006 92 RTN	019 39 COS	032 43 RCL	045 65 ×	058 73 RC*	070 16 16	082 99 PRT
007 76 LBL	020 22 INV	033 32 32	046 01 1	059 35 35	071 43 RCL	083 97 DSZ
008 12 B	021 59 INT	034 95 =	047 00 0	060 95 =	072 36 36	084 30 30
009 42 STO	022 42 STO	035 48 EXC	048 85 +	061 22 INV	073 55 ÷	085 00 00
010 32 32	023 33 33	036 34 34	049 01 1	062 59 INT	074 01 1	086 16 16
011 92 RTN	024 65 ×	037 55 ÷	050 95 =	063 65 ×	075 00 0	087 91 R/S
012 76 LBL	025 43 RCL	038 01 1	051 22 INV			

HYPERBOLIC FUNCTIONS.- Paul B. Wright, 16860 Slover, Fontana CA, 92335, has written a superprogram by that name. Unfortunately, it contains 25 pages of text and program listings, so that I would have to dedicate two issues to it. Paul is willing to send you a copy, if you send him $ 5.00 US to cover copying and postage. A description follows:

The hyperbolic functions are of real variables using the Gudermannian Function. Tables of db vs Hyperbolic Functions as well as db vs Voltage (Current) and Power Ratios are generated and may be chosen for either manual or automatic operation. (with or without the printer) The function to be tabulated may be started at any value and have any chosen incremental db step.

The functions programmed are: decibells (db); Θ in nepers; ϕ,the Gudermannian of ϕ abbreviated gdϕ; Sinhϕ; Tanhϕ; Tanh$\phi/2$; Coshϕ^2; k, a voltage (current) ratio; k^2, a power ratio; the reciprocals of the hyperbolic and k functions; Return Loss; Reflection Loss; Voltage Standing Wave Ratio; (VSWR). Tables of db and db vs VSWR may also be generated. The total program package has 25 pages and is comprised of the following:

1. A brief theory of the Gudermannian Function and the relationships with the trigonometric and hyperbolic functions.
2. Two tables that summarize all of the functions used in the program and from which concise definitions of the various functions may be obtained. Two algorithms are given that make it possible to get hyperbolic functions using the Gudermannian Functions from ordinary trigonometric functions. It is not necessary to use the definitions in terms of the exponential functions to get the hyperbolic functions. The Gudermannian is more efficient and much more economical of programming space.
3. Eleven examples are given to show the usage of the algorithm, tables and program instructions in the design of : minimum loss, T, π, bridged T, lattice networks together with the T to π and π to T transformations. Return, reflection losses and VSWR are included.
4. A 32 section index of instructions.
5. 181 Instructions for the 32 sections.
6. A 476 step program that may be recorded on sides 1 and 2 of a mag card.

Either degrees or radians may be used in the calculations, which will work with or without the printer. Even a non-programmable calculator may be used, be it at a slower pace. Besides calculating hyperbolic functions this program gives db, VSWR and loss as a bonus. When given one function, any or all other functions may be obtained.

Please DO NOT WRITE TO ME FOR THIS PROGRAM. Write Paul Wright instead, at the address mentioned above and don't forget to send Paul some compensation for his time and effort.

ASTRONOMY- David Bartholomew tells me that he has had trouble to have the HP programs in the Astronomy package give the correct answers. I don't have my HP-67 anymore, so I am unable to check it.

Those members who have purchased the package (translation from Display) would you please write me if you experienced problems with the HP programs and if they gave you answers different from the ones for the TI-59? Of course, I would be even more grateful if you could send me (eventual) corrections.

INTERPOLATION.- by Fred Fullam, Jr. This program is handy when straight-line interpolation or rationing is desired. For example, if you know two values from a table and you want to know a value between them. Suppose at 32 degrees F the value is 3.16×10^6, while at 100 degrees F the value has decreased to 1.76×10^6. We would like to now the value at 75 degrees F.

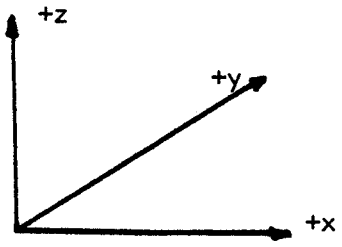
The instructions are for calculator use only, without the printer:
Initialize, press A. When "1" appears in the display, enter 32 and press R/S.
When a "2" appears, enter 3.16 EE 6 and press R/S. When a "3" appears, enter 100 and press R/S. Now a "4" will appear. Enter 1.76 EE 6 and press R/S.
When finally a "5" appears, enter 75 and press B. The result appears as 2.27×10^6.
You can now interpolate any value. For example, enter 50 and press B. The result is 2.78×10^6, etc.

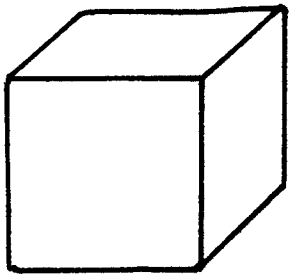
For a new set of interpolations, initialize by pressing A again.

000 76 LBL	007 43 RCL	014 61 GTO	021 04 04	028 43 RCL	035 75 -	042 43 RCL
001 11 A	008 00 00	015 23 LNX	022 75 -	029 02 02	036 43 RCL	043 03 03
002 47 CMS	009 22 INV	016 76 LBL	023 53 (	030 54)	037 01 01	044 75 -
003 76 LBL	010 57 ENG	017 12 B	024 53 (	031 55 ÷	038 54)	045 43 RCL
004 23 LNX	011 91 R/S	018 72 ST*	025 43 RCL	032 53 (	039 54)	046 05 05
005 69 DP	012 72 ST*	019 00 00	026 04 04	033 43 RCL	040 65 ×	047 95 =
006 20 20	013 00 00	020 43 RCL	027 75 -	034 03 03	041 53 (	048 91 R/S

OBLIQUE TRANSFORM, by Fred Fullam, Jr. This program is used when drawing three-dimensional figures. It will transform three-dimensional information (x, y, z) into two-dimensional information (x, y) such that the three-dimensional figure can be drawn on a two-dimensional plane.

The formulas used are:
$x2 = x3 + y3\ (F) \cos A$ and $y2 = z3 + y3\ (F) \sin A$.
in which x3, y3 and z3 are the three-dimensional values and
x2 and y2 are the two-dimensional values.
F = forshortening factor, ususally between .5 and 1.
A = projection angle, usually between 30 and 60 degrees.





F = .5
A = 30 degrees

1 inch cube

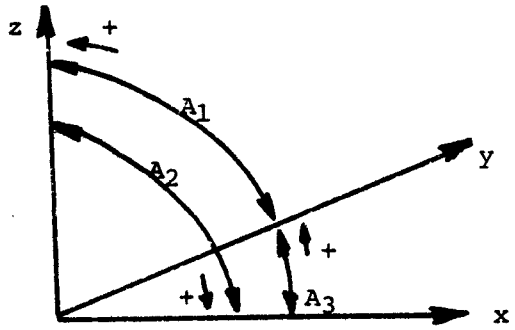
Instructions:

Enter the forshortening factor F and press E. This initializes the program.
When a "1" appears in the display, enter x3 and press R/S.
When a "2" appears, enter y3 and press R/S.
When a "3" appears, enter z3 and press R/S.
x2 will now appear in the display.
Press B and y2 will appear in the display.

Of course, you can enhance this program by adding PRT statements in strategic places.

000 76 LBL	010 43 RCL	020 11 11	030 69 DP	040 23 LNX	050 02 02	060 65 ×
001 15 E	011 11 11	021 95 =	031 20 20	041 76 LBL	051 85 +	061 43 RCL
002 47 CMS	012 95 =	022 42 STO	032 43 RCL	042 33 X²	052 43 RCL	062 02 02
003 42 STO	013 42 STO	023 12 12	033 00 00	043 00 0	053 01 01	063 85 +
004 11 11	014 13 13	024 76 LBL	034 67 EQ	044 42 STO	054 95 =	064 43 RCL
005 91 R/S	015 43 RCL	025 11 A	035 33 X²	045 00 00	055 91 R/S	065 03 03
006 42 STO	016 12 12	026 04 4	036 91 R/S	046 43 RCL	056 76 LBL	066 95 =
007 12 12	017 38 SIN	027 32 X:T	037 72 ST*	047 13 13	057 12 B	067 91 R/S
008 39 COS	018 65 ×	028 76 LBL	038 00 00	048 65 ×	058 43 RCL	
009 65 ×	019 43 RCL	029 23 LNX	039 61 GTO	049 43 RCL	059 12 B	

ORTHOGRAPHIC PROJECTION,- by Fred Fullam, Jr. This program is similar to the Oblique program, except that it allows the object to be rotated about each of the 3-D axis. As before, the three-dimensional coordinate system is:



but three angles have been added: A1 is the angle the object is rotated through around the x-axis. A2 is the rotation angle about the y-axis, while A3 is the rotation angle about the z-axis. The signs for the angles follow the right-hand rule: point the thumb of the right hand in the positive direction of the axis and the fingers of the right hand point in the positive angle direction.

The formulas used are as follows:

x2 = cosA2 cosA3 x3 - cosA2 sinA3 y3 + sinA2 y3

y2 = (sinA1 sinA3 - cosA1 sinA2 cosA3) x3 + (sinA1 cosA3 + cosA1 sinA2) y3 + (cosA1 cosA2) z3.

These formulas rotate the object about the x-axis first, then the y-axis and finally about the z-axis.

To operate the program, which is for calculator-only use, initialize by pressing E. When "11" appears in the display, enter angle A1 and press R/S. When a "12" appears enter angle A2 and press R/S. When a "13" appears, enter angle A3 and press R/S.

When a "1" appears, enter x3 and press R/S. When a "2" appears, enter y3 and press R/S. When a "3" appears, enter z3 and press R/S.

x2 appears now in the display. Press B to obtain y2.

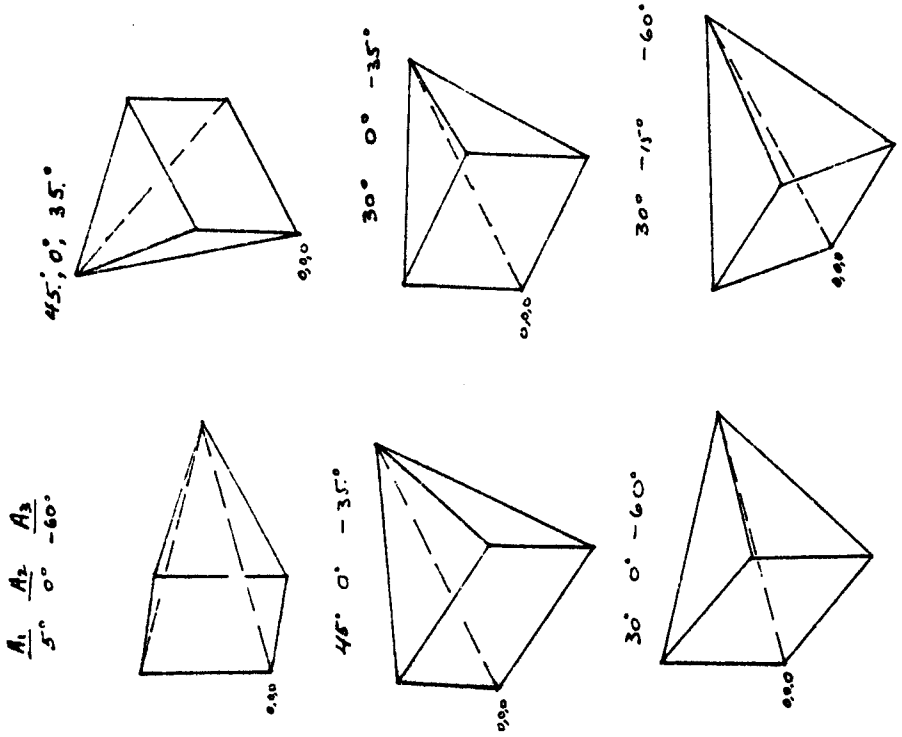
To obtain the next point, remember that the three angles have been entered already. So, just press A. When a "1" appears, enter x3 and press R/S. When a "2" appears enter y3 and press R/S. When a "3" appears, enter z3 and press R/S. X2 will appear in the display. Press B to obtain y2, etc. etc. for all the subsequent points. Start each time by pressing A.

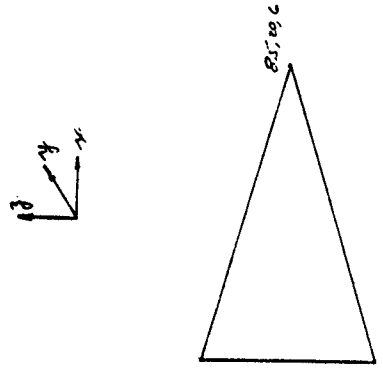
Two possible enhancements come to mind: inclusion of PRT statements and a scaling factor routine. The latter was published in one of the issues of the TI PPC NOTES in 1980: v5n6p8-9.

On the next page are given several examples of how a pyramid may be rotated about the three axis, in order to obtain different views of that object.

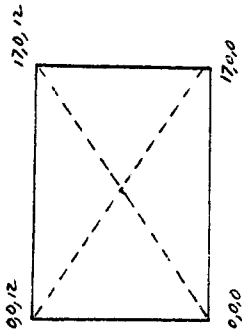
```
000  lbl LBL      027  00  00      054  13  13      081  75  -       108  43 RCL      135  00  00      162  43 RCL
001  15  E        028  43 RCL      055  39 COS      082  53  (       109  16  16      136  61 GTO      163  03  03
002  47 CMS       029  11  11      056  42 STO      083  43 RCL      110  95  =       137  24 CE       164  95  =
003  01  1        030  38 SIN      057  13  13      084  11  11      111  42 STO      138  76 LBL      165  91 R/S
004  04  4        031  42 STO      058  43 RCL      085  65  x       112  20  20      139  34 FX       166  76 LBL
005  32 X:T       032  14  14      059  12  12      086  43 RCL      113  43 RCL      140  00  0       167  12  B
006  01  1        033  43 RCL      060  65  x       087  15  15      114  11  11      141  42 STO      168  53  (
007  00  0        034  11  11      061  43 RCL      088  65  x       115  65  x       142  00  00      169  43 RCL
008  42 STO       035  39 COS      062  13  13      089  43 RCL      116  43 RCL      143  53  (       170  19  19
009  00  00       036  42 STO      063  95  =       090  13  13      117  12  12      144  43 RCL      171  65  x
010  76 LBL       037  11  11      064  42 STO      091  95  =       118  95  =       145  17  17      172  43 RCL
011  23 LNX       038  43 RCL      065  17  17      092  42 STO      119  42 STO      146  65  x       173  01  01
012  69 DP        039  12  12      066  43 RCL      093  19  19      120  21  21      147  43 RCL      174  54  )
013  20  20       040  38 SIN      067  12  12      094  43 RCL      121  76 LBL      148  01  01      175  85  +
014  43 RCL       041  42 STO      068  65  x       095  14  14      122  11  A       149  85  +       176  43 RCL
015  00  00       042  15  15      069  43 RCL      096  65  x       123  04  4       150  53  (       177  20  20
016  67  EQ       043  43 RCL      070  16  16      097  43 RCL      124  32 X:T      151  43 RCL      178  65  x
017  33 X²        044  12  12      071  95  =       098  13  13      125  76 LBL      152  18  18      179  43 RCL
018  91 R/S       045  39 COS      072  94  +/-     099  95  =       126  24 CE       153  65  x       180  02  02
019  72 ST*       046  42 STO      073  42 STO      100  85  +       127  69 DP       154  43 RCL      181  54  )
020  00  00       047  12  12      074  18  18      101  53  (       128  20  20      155  02  02      182  85  +
021  61 GTO       048  43 RCL      075  43 RCL      102  43 RCL      129  43 RCL      156  54  )       183  43 RCL
022  23 LNX       049  13  13      076  14  14      103  11  11      130  00  00      157  85  +       184  21  21
023  76 LBL       050  38 SIN      077  65  x       104  65  x       131  67  EQ      158  53  (       185  65  x
024  33 X²        051  42 STO      078  43 RCL      105  43 RCL      132  34 FX       159  43 RCL      186  43 RCL
025  00  0        052  16  16      079  16  16      106  15  15      133  91 R/S      160  15  15      187  03  03
026  42 STO       053  43 RCL      080  95  =       107  65  x       134  72 ST*      161  65  x       188  95  =
                                                                                                      189  91 R/S
```

Orthographic Projection, Fred Fullam. EXAMPLES.





START WITH A PYRAMID AS SHOWN BELOW



X3	Y3	Z3
0	0	0
17	0	0
0	0	12
17	0	12
8.5	20	6

5°, 0°, -60°	
X2	Y2
0	0
8.5	-3.28
0.0	11.95
8.5	10.67
21.6	6.21

45°, 0°, 35°	
X2	Y2
0	0
13.9	6.9
0	8.5
13.9	15.4
-4.6	19.3

30°, 0°, -35°	
X2	Y2
0	0
13.9	-4.9
0	10.4
13.9	5.5
18.5	10.9

30°, 0°, -60°	
X2	Y2
0	0
8.5	-7.4
0	10.4
8.5	3.0
21.6	6.5

45°, 0°, -35°	
X2	Y2
0	0
13.9	-6.9
0	8.5
13.9	1.6
19.4	13.4

REPARTITIONING THE TI-58C: By now you have heard of the extra 32 steps (480-511) or four memories in the TI-58. Unfortunately these steps are usually accessible only from the keyboard using the STF IND...sequence. But in the TI-58C, the memory corresponding to steps 480 to 487 is used by the constant memory feature to set the partition, the FIX n display mode, and to check for memory loss at turn-on. By changing the contents of this memory the TI-58C can be repartitioned so that these 32 steps or four memories are directly accessible.

The first 13 digits of this memory store an unscaled ln 10. If this number is not present at turn-on, the calculator erases all data and program memory to protect the user from unsuspected memory loss. Instead the user suffers from known memory loss.

The next two digits of the memory (i.e. the first digit of step 480 and the last digit of step 481) set the partition at turn-on. By writing the proper key-codes in these steps, the TI-58C can be partitioned like a TI-59 or entirely new partitions may be generated. First, steps 480-481 must be accessed. To do this, write in program memory LBL A FIX 0 R/S, enter 609.000000609 into the display, and press A. Then press STF IND 7 INV LRN. You should be at step 480. Write |x| RCL in steps 480-481. Then cycle the on-off switch, press Op 16 and you will see a partition of 0.09. This partition means you have no memories and 1200 program steps. Of course there is no hardware beyond step 511- any code you see in these steps seems to be miscellaneous garbage (electrons that got lost in the circuits, perhaps). Now you can repartition to any of the primary partitions in Table 1 by simply going to steps 480-481, entering the proper code and cycling the on-off switch.

However, the most useful partitions are those pseudo-59 partitions in the table. To access them, get into one of the first four primary partitions then simply repartition from the keyboard via Op 17 as you would with a TI-59. These give the TI-58C user an extra 32 steps (480-511) or an extra four memories. For instance, a partition of 479.59 gives 480 program steps and four valid (hard-ware-backed) memories numbered 56 to 59. Memory 59 (or steps 480,481) must be used with care since any change in FIX n or partitioning will alter the contents (but note that the reverse is true only if the on-off switch is cycled) and ln 10 must be restored before turn-off or the calculator will be cleared at turn-on. Also, at turn-off, a psuedo-59 partition will revert to the corresponding primary partition.

The last digit of step 480 stores the FIX n display mode. This digit will equal n+2 in a FIX n display for $0 \leq n \leq 7$. In FIX 9 or FIX 8 this digit will be a zero, but in FIX 8 the first digit of step 480 will be increased by one without affecting the partition. This brings up an important point. A Partition of 0.39 in FIX 9 and a partition of 0.49 in FIX 8 both store code 20 41 in steps 480 and 481, but the calculator goes into the correct partition and FIX n at turn-on. How does the calculator determine which partition and FIX n is correct? I was afraid you'd never ask. Well, in the FIX 8 case the keycode 20 is actually the hexidecimal keycode 1A. (This was determined through other means. None of the hex keycodes ending in A do anything new). Putting a 1 in the last digit of step 480 gives a sort of FIX -1 display. This rounds off the display (but not the display register) to the tens' place and adds a negative sign. Thus, a 5 is displayed as -10 in the last three places of the display and a 2.3 is displayed as -00. In an exponential display, if the most significant digit (MSD) is less than 5, the display shows 0.00; if the MSD is greater than or equal to 5, the display will show 0.XX where XX is the exponent +1.

That's about all there is to repartitioning the TI-58C. Doing something useful with all this is left as an exercise for the reader. Patrick W. Acosta

Continues on next page.

Code in Step... 480	481	Primary Partition	Memories	Pgm. Steps	Pseudo-59 Partition***
90	41**	239.39*	90	240	--
80	41	159.99*	100	160	--
70	41	79.09*	110	80	--
60	41	0.19*	120	0	--
50	41	0.09	0	1200	--
40	41	0.19	0	1120	--
30	41	0.29	0	1040	--
20	41	0.39	0	960	959.00
10	41	0.49	0	880	879.09
00	41	0.59	0	800	799.19
90	40	0.69	0	720	719.29
80	40	0.79	0	640	639.39
70	40	0.89	0	560	559.49
60	40	479.00	0	480	479.59
50	40	399.09	10	400	399.69
40	40	319.19	20	320	319.79
30	40	239.29	30	240	239.89
20	40	159.39	40	160	159.99
10	40	79.49	50	80	--
00	40	0.59	60	0	--

**Any keycode from 42 to 49 may be substituted for 41 with identical results.
*These partitions allow pseudo-59 partitions to be accessed via Op 17.
***Pseudo-59 partitions have the usual number of memories and program steps.

FILTER DESIGN.- The journal "r.f. design" in their Nov/Dec 1980 issue on pages 22 through 31 published an excellent article and TI-59 program titled NARROWBAND BUTTERWORTH OR CHEBYSHEV FILTER DESIGN USING THE TI-59 CALCULATOR. The program permits the design of one of the two above filters of any ripple, orders 2 through 9. Formulas and charts are given. Although the program works perfectly, a few "disturbing" sequences were spotted. For example, I saw the following : 0 STO 06 0 STO 07 ... all the way to 0 STO 15. We all know that, once you have a zero in the display, it is not going to disappear from it, unless you overwrite it with something else. So, the author could have saved himself and his "victims" quite a lot of unnecessary zero punching.
What is a little more "disturbing", however, is, that the author says in his initialization instruction: Press 4 OP 17 CLR. Then press either B for Butterworth or C for Chebyshev. I know you have the solution already. Just write 4 OP 17 as the first steps in both LBL B and LBL C, of course. Or, if you want to be fancy, write SBR 000 in both LBL B and LBL C. Then write, starting at step 000: 4 OP 17 CLR RTN.
The author is Lee R. Watkins, Martin Marietta, Waterloo, CO. He acknowledges the help of F. Stillwell in condensing the program to a more efficient form.

CODEC PROGRAM COMPANDS SAMPLES FOR μ-LAW SIMULATION.- Clive McCarthy, Northern Telecom Inc. Santa Clara, CA. Electronics, Dec 18, 1980, pp 114-115. This clever program simulates the operation of a pulse-coded-modulation encoder that conforms to the industry-standard μ-255 companding law. Given a sinusoidal input signal of any amplitude, phase, and frequency, the program finds the corresponding binary output data from the standard approximation of a logarithmic compresssion curve.
Telecommunications engineers note this TI-59/PC100 program.

Manuel