52-NOTES

Volume 2 Number 2 48/39 February 1977

Newsletter of the SR-52 Users Club
published at
9459 Taylorsville Road
Dayton, OH 45424

- -

TIPS

Clever_Uses of_INV_P/R: Dix Fulton (83) suggests the following sequences to handle raising a negative number (in Reg 01) to an integer power (in Reg 02): for the SR-52: RCL 01 STO 00 0 INV *P/R X RCL 02 = cos X RCL 00 y^x RCL 02 =; for SR-56: RCL 1 xEt 0 *f(n) R-P X RCL 2 = cos X xEt y^x RCL 2 =. Jared Weinberger (221) notes that: 0 INV *P/R for the SR-52 or 0 *f(n) R-P for the SR-56 replaces the contents of Reg 00 (t) with its absolute value. For a practical application of these facets of INV *P/R see "Short, fast quadratic solutions" in the ROUTINES column below.

More_on Charger Connection: Larry Mayhew (145) reports that John Allen's charger connection method (V1N7p4) appears to cause execution speed to slow slightly (he had to change the constants in his timer program (V1N7p6) to around 9600... a change of 1-2%). This condition appears to hold until the machine is turned off. Subsequent "normal" turn-ons apparently produce normal execution rates. Anyone care to venture a guess as to what's going on?

A Decrement-Only_dsz: (52) Larry has found the sequence ... *dsz 0 op ... (where op is any non-numeral instruction) to work whenever Reg 00 is to be decremented without a zero-test for transfer.

1/0=0: Larry further notes that in cases where reciprocals of sequenced numbers are calculated and it is desired to have 1/0=0, the sequence: n y^x 1 +/- = will work, where n GE 0. However, results are not always exact, and rounding may be necessary if integer results are required.

Trig_Function Anomalies: In the course of trying to get a practical program to work, Larry found that there are normal trig sequences which can hang up and produce an error condition. There appear to be critical numbers close to 90° (pi/2 in radian mode) which when operated upon by INV sin produce numbers slightly larger than one, and other critical numbers close to zero which when operated upon by INV cos also produce such numbers (SR-56 too). Unfortunately, some of these critical numbers can be created by the sin and cos functions themselves. The sequence (in degree mode): 0 INV cos sin INV sin produces an error condition, since 0 INV cos sin produces 89.9999999987 and the machine sine of that is 1.000000000001. The worst case I have found so far is to take the sine of 89.9999999540 which produces 1.000000000004. So the moral of the story is: if in doubt, round!

- -

The SR-52 Users Club is a non-profit loosely organized group of SR-52/56 owners/users who wish to get more out of their machines by exchanging ideas. Activity centers on a monthly newsletter, 52-NOTES edited and published by Richard C Vanderburgh in Dayton, Ohio. The SR-52 Users Club is neither sponsored nor officially sanctioned by Texas Instruments, Incorporated. Membership is open to any interested person: $6.00 includes six future issues of 52-NOTES; back issues start June 1976 @ $1.00 ea.

TIPS (con)

More_on + STO 60_(52): Chuck Sanford (214) and Dallas Egbert (384) note that the sequence: + STO 60 = leaves a pending INV, and Chuck points out that + STO 60 = EE is a short manual method of displaying the 4th through 13th digits (of numbers greater than one). The EE executes as INV EE.

The Printed "?" as an Identifier (PC-100): Sandy Greenfarb (200) has found that purposely creating an error condition prior to a *prt is a handy way to mark selected outputs, since the printed "?" is easy to spot as well as to create.

Step_223_(99) *list: executes as a normal HLT (R/S) when run with the printer; otherwise a run-past-last-step error condition is created.

Optimized Hardware_Interrupts: Jared Weinberger (221) points out that the approach suggested in V1N2p4 (top) requires more execution time than necessary, since advantage can be taken of the fact that the machine executes the sine function quicker if the result is zero than otherwise. If the sequence is rearranged as follows: ... *pi sin *ifzro *pi HLT *LBL *pi ... then uninterrupted loop execution occurs in radian mode. SR-56 mechanization needs to take into account the fact that the choice of angle modes during program execution is between degrees and grads. The sequence starting at step 00: 90 cos *x=t 07 R/S ... RST executes uninterrupted when in degree mode and halts at the R/S when in grad mode. (The t register must be zeroed prior to execution)

Re-softening_a_Hardened Display: Dallas Egbert (384) notes that the sequence EE INV EE softens a hardened display to the extent that successive keyed numerals are positioned at the LSD of the mantissa. For example, key: 1.23 = EE INV EE 45 and you get 1.2345. Dallas also notes that any hardened number followed by EE INV EE CE can be fully restored via Ed Haas' routine (V1N6p3). However, the EE rounds the mantissa to ten places,

ROUTINES

Short Fibonacci Sequence_(56): Paralleling the SR-52 routine (V1N1p3), Sandy Greenfarb (200) has devised a short SR-56 Fibonacci sequencer which, slightly modified, lists starting at step 00 as: + *pause xEt RST; (xEt means x exchange t). In run mode key: CLR xEt 1 RST R/S, and see each successive Fibonacci number pause-displayed. (Replace the *pause with a *prt for printed output via the PC-100).

Short,_Fast Quadratic Solutions: A couple of years ago, John Herro (HP-65 Users Club member #47) found a clever use of the rectangular to and from polar conversion functions (see TIPS above) in developing an efficient algorithm for calculating both real and complex roots of quadratic equations with real coefficients. The following routines are my mechanizations of John's algorithm: for the SR-52: *LBL A STO 01 *1/x X HLT div 2 = STO 02 *x^2 - HLT div RCL 01 = STO 00 0 INV *P/R div 2 = *EXC 00 *rtx *EXC 00 *P/R HLT RCL 00 - RCL 02 = HLT RCL 02 +/- - RCL 00 = *rtn; for the SR-56: starting at step 00: STO 1 *1/x X R/S div 2 = STO 2 x^2 - R/S div RCL 1 = xEt 0 *f(n) R-P div 2 = xEt *rtx xEt *f(n) P-R R/S xEt + (CE - RCL 2) R/S RCL 2 = +/- R/S. To run them (SR-56 instructions in parentheses): Key a, press A (RST, R/S); key b, RUN (R/S); key c, RUN (R/S); see imaginary part of root displayed; key RUN (R/S) and see real part of root one; key RUN (R/S) and see real part of root two. a, b, c are the coefficients of a quadratic in the form $ax^2 + bx + c = 0$.

- - - - - - - - - - - -

ANALYSIS OF A SUPER PROGRAM

Barbara Osofsky (420) has indeed produced a one-card program that not only calculates the determinant and inverse of a 5 X 5 matrix, but also does a 4 X 4 determinant and solves a system of four simultaneous equations. While many of you may be wearying of matrix programs, I think you will find the time taken to see how this one is structured to be well spent, since many constructs are applicable to other how-to-get-it-all-to-fit situations, and the analysis that follows is concerned with programming techniques, not applied matrix theory. As it lists below, Barbara's program was modified to reduce the amount of manual keying required, and a few labels and registers were rearranged for I/O ease.

Let's begin by looking at the input routine, beginning at step 138. At first glance it looks straight forward enough, except for the error test and the extra summing. But note that routines D', D and A' all run right through it! Thus routine A is used each time D', D or A' is called (which is quite a few) besides for inputting data. Routine B' does most of the arithmetic, and is called 60 times for the 5 X 5 determinant and 300 times more for the inverse. Note that routine B' is written "in-line", that is, it doesn't call any subroutines itself. This is necessary because it itself is a third-level call (by E to C' to B'), and it would run slower, even if it could be shortened with subroutine calls. Note that space is saved by running it into E'. Routine D' heads up the D' D A' A chain, and calls a part of itself four times before running directly through it (the A' A part). The only conditional branch occurs as an *iferr flag at step 145 (see V2N1p4). The flag is set with a user call to the undefined label C, and reset with the CE at step 166. An important output (the 5 X 5 determinant) is created just prior to execution of step 180; yet without parentheses it is pushed into Reg 60 with a + operator and safely preserved during the execution of some 130 succeeding steps, until it finally emerges at step 191 for disposition. This is possible because no intervening display arithmetic occurs.

Throughout, it is careful structuring and ordering of subroutines that saves the most space, and makes it possible for the user to accomplish so much with a one-card program. The program itself had to be fit within 200 steps, since 3 program storage registers are used for data.

SR-52 Program: 5 X 5 Determinant and Inverse; 4 X 4 Determinant and
(with PC-100) 4 Simultaneous Equations Barbara Osofsky/Ed

5 X 5 User Instructions:

1. Initialize: 95 STO 66
2. Input the 25 elements (e_i) with column-wise catenation: key e_i, press A, see 1; repeat for i=1,2, ... 25.
3. Get the Determinant: press B then E; see determinant displayed after about 2 minutes.
4. Ignore 4 lines of print: On printer press ADV, tear off tape, press PRINT, ADV.
5. Get Inverse: Press B then E; first column printed, see 1 displayed.
6. Key next column number (r), press RUN, C then D then E; rth column printed, 1 displayed; repeat for r=2,3,4,5.

5 X 5 User Instructions (con)

Notes:

1. If elements are input with row-wise catenation, inverse elements are printed by rows.
2. Before step 3 execution, the number 1 must be displayed; before step 5 execution, the determinant must be displayed.
3. At step 6, do not cancel error condition caused by pressing C; routine D ends by flashing 5.

4 X 4 User Instructions:

1. Initialize: 95 STO 66 1 A
2. Input the constant column (c_i): Key c_i, press A; see 1; repeat for i=1,2,3,4.
3. Input coefficients (a_{ij}) by column: Key a_{ij}, press A; see 1; repeat for i (row) =1,2,3,4,5 and j (column) =1,2,3,4 where a_{1j}=0 are four dummy elements that must head each column. a_{54} is the last coefficient, and should be in Reg 19 (following RCL 19, replace the 1 in the display before proceeding).
4. Get Determinant: Key B then *C'; determinant is printed; on printer key ADV.
5. Key 0 = +/- B then E; see printed: -1, x_1, x_2, x_3, x_4; -1 displayed. Original input ordering of elements is returned (Reg 95-19).

Program Listing:

```
000:  *LBL *B' STO 68 (*IND RCL 66 X *IND RCL 68 - 1 SUM 66 +/- SUM 68
024:  *IND RCL 66 X *IND RCL 68) X *LBL *E' 1 SUM 66 *rtn *LBL *C' (8
046:  STO 66 4 *B' *E' 17 *B' +/- + 19 *B' STO 66 7 *B' -14 *B' *E' 17
072:  *B' + 19 *B' *E' 2 *B' + 9 *B' *E' 2 *B' - 4 *B' 6 STO 66 12 *B')
097:  div RCL 67 B X *prt RCL 95 + *rtn *LBL *D' 10 INV SUM 69 *LBL D
119:  95 STO 66 *A' *A' *A' *A' *LBL *A' *IND RCL 66 *IND *EXC 69
138:  *LBL A *IND STO 66 5 *iferr 150 1 SUM 69 SUM 66 *rtn *LBL B +/-
160:  STO 67 *rtn *LBL E CE 0 STO 69 *C' D *C'D *C' D *C' D *C' 5
181:  SUM 69 *D' *D' *D' *D' 0 = *pap HLT + 94 = STO 69 HLT
```

- - - - - - - - - - - - -

MAGAZINE REVIEW

Members have asked for information concerning personal programming-related periodicals. Herewith a start. I invite reviews of other related periodicals from the membership.

BYTE: (the small systems journal) is aimed at the computer hobbyist, and appears to be a fairly respectable commercial operation (total paid circulation 30 Sep 76 of 62,280). Published monthly by Virginia and Manfred Peschke at 70 Main St, Peterborough, NH 03458 at $12 per year, you'll have to scrounge for back issues. Topics range over both hardware and software aspects of building and using the class of machines centered on the 8080 and 6800 type microprocessors, although personal programmable calculators have occasionally been the subject of articles and letters.

POPULAR COMPUTING: is published monthly by Fred Gruenberger (27) at Box 272 Calabasas, CA 91302; Jan 77 is the 46th issue. Each issue runs 20 unreduced typewritten 8½x11 pages, with almost no advertising. Topics generally concern numerical problems and puzzles whose solutions are apt to require the use of large computing systems. However, there are problems which can be productively explored with personal programmables. On occasion computer-related philosophical topics are aired. Subscription is $20.50 per year; back issues $2.50 each.

A FRACTURED DIGITS, ALL FLAGS, GAME APPLICATION (52)

Alan Charbonneau (306) found that dice rolling for the poker-like game of Yahtzee could be mechanized for the SR-52 in such a way as to take advantage of display variations (V1N2p5), as well as to pose a selection requirement nicely met by using all 5 flags. With a few optimization revisions, Alan's program appears below. Routine D' is called by the main program (which starts at step 038) each time a new die roll is required. Dice values are temporarily stored and selectively changed, then integratedwith a skeleton string in Reg 06 for Reg 60 manipulation to produce the 5-dice display. The flags "freeze" selected dice for partial rolls. The required flag manipulations spotlight how nice it would be to have a function that only resets all flags (not do everything *rset does). Perhaps there is a "puzzling sequence" (V2N1p1) that would do the trick. p73 is sort of an inverse of what is required since it causes a branch to step 000, but does not reset flags.

SR-52 Program: Yahtzee   Alan Charbonneau (306)/Ed

User Instructions:

1. Key random number seed: 0 LT seed LT 1, press *A'; see 9D99; press =, see first roll of 5 dice: d_1-d_2-d_3-d_4-d_5
2. (Optional) Save dice by pressing one or all of A, B, C, D, E for corresponding dice positions
3. Press RUN to roll remaining dice (if step 2 is omitted, all 5 dice are rolled); see 9D99; press =, see next dice configuration. Go to step 2 for dice save, or repeat step 3 for new roll.

Program Listing:

```
000:  *LBL *D' RCL 98 X *pi = STO 98 - .5 = *fix 0 *D.MS INV SUM 98
022:  RCL 98 X 5.5 + 1 = *D.MS *IND STO 00 *rtn 5 STO 00 *ifflg 0 048
047:  *D' INV *stflg 0 *dsz 0 *ifflg 1 059 *D' INV *stflg 1 *dsz 0
064:  *ifflg 2 070 *D' INV *stflg 2 *dsz 0 *ifflg 3 081 *D' INV *stflg 3
084:  *dsz 0 *ifflg 4 092 *D' INV *stflg 4 1 EE 95 STO 99 5 STO 00
106:  RCL 97 STO 06 *IND RCL 00 X RCL 99 = SUM 06 100 INV *PROD 99
131:  *dsz 112 RCL 96 + STO 60 RCL 06 HLT CLR GTO 038 *LBL *A' STO 98
156:  CLR GTO 038 *LBL A CLR *stflg 0 GTO 037 *LBL B CLR *stflg 1
175:  GTO 037 *LBL C CLR *stflg 2 GTO 037 *LBL D CLR *stflg 3 GTO 037
197:  *LBL E CLR *stflg 4 GTO 037 0 0 *pap cos 6666 . *pap *ifzro
217:  90000 *ifzro *ifzro
```

- - - - - - - - - -

FORUM

Sick_Machine_Exchange: Robert Hausafus (431) suggests calling your nearest TI machine exchange center before taking your machine in; they may not have the unit you need in stock, in which case ask them to order one for you. Robert's problem with a PC-100 may be a common one: inoperative character dots.

Applications_Topics: Thomas Watkins (347) suggests that each month 52-NOTES focus on one applications topic (games, surveying, aviation/navigation, physics/chemistry, etc), emphasizing routines that maximize user efficiency and ease. I invite members to send me their algorithms, routines, approaches; or a description of problems encountered trying to get a particular applications program to work. As info builds sufficiently on any one topic, I'll publish the material in 52-NOTES.

SR-56 PROGRAM EXCHANGE UPDATE (see V1N6p6)

Dave Johnston's 1 Feb 77 Catalog lists 46 programs on math, statistics, physics, games, and misc. Members wishing to have access to SR-56 programs covering technical disciplines or specific problems not already addressed are invited to speak up. Send your requests to both Dave and me.

MISCELLANY

Corrections to TI Software: Since most applications library and PPX-52 programs address specialized topics, corrections and revisions will not be given space in 52-NOTES. TI is the appropriate focal point.

Display Arithmetic Modification: Machines shipped after late October 1976 were manufactured with a new chip that automatically truncates the 13th place of a display operand just prior to its interaction with a Reg 60 operand. Display values operated upon singly and data register contents are not affected, and retain up to 13 places in the usual manner. A quick test to see how your machine behaves is to key: 3 *1/x - STO (CE for SR-56) =. A resulting zero signifies the new chip; -3D-13 the old. A word of caution: While most programs written for old-machine execution should work on the new ones (upward compatibility), the reverse may not be true since some rounding or undesired register arithmetic can be dispensed with when programming the new machines.

SR-52 Successor: Unofficial sources point to the introduction of a new machine in ten weeks or so that will probably be marketed as SR-52 II. It is likely to combine SR-52 and SR-56 features and fill in some or all of the "missing" SR-52 registers (20-59). Printer compatibility will probably be limited to the PC-100A (which has a 3-position calculator selector switch).

Program Listing Conventions: Although some have suggested dropping the * shift indicator, I'm going to keep it for program listings, as it's too easy to forget the shift key without the reminder when punching a new program from a listing. When functions are referred to in the text of an article, the * will be omitted. User instructions designed for both SR-52 and SR-56 use will show SR-56 instructions in parentheses following the SR-52 instructions they replace.

A New Facet to Registers 20-59: It turns out that these 40 "non-registers" point to Reg 00 or step 000 when invoked with *IND. For example key: 999 STO 00 CLR *IND RCL 45; see 999. Or, key: GTO 123 *IND GTO 59 LRN and see step 000.

SR-60 Notes: SR-60 users are invited to share ideas, discoveries, problems, etc. Send your material to Tom Ferguson (421) 3308 N 87th Ave Phoenix, Arizona 85037, who has kindly volunteered to serve as a focal point. If there appears to be sufficient interest, I will publish his consolidations in 52-NOTES from time to time.

Mode Terminology: To clear up some queries, there are really only two operational modes: learn and run (or calculate), and I prefer "RUN" to "calculate" since it is shorter. The machine is functionally in the same mode whether functions are executed manually, SST or under program control.

Membership Address Changes: 248: 4092 Knollwood Grand Blanc MI 48439; 306: 1885 Garfield #9 San Luis Obispo, CA 93401; 347: 2340 Moraine Circle #6 Rancho Cordova, CA 95670.