52-NOTES

Volume 3 Number 5 48/39/38 May 1978

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

AN ANALYSIS AND COMPARISON OF 3 CALENDAR PROGRAMS (59/PC-100A)

Both Jared Weinberger (221) and Lou Cargile (625) have written programs to generate and print calendars, using the ML-20 CROM program, and outputting in a standard month-by-month weekday-column format. This is the sort of problem where neither its complexity nor the best approaches to its solution are apt to become fully apparent until at least one approach has been tried. And so it wasn't until I examined their programs in detail that I appreciated the degree of the challenge and was motivated to try to find a better approach. Both programs require 3 card-sides for recording, but neither uses all the recordably available memory, and since it is unlikely that a comparable program could be made to fit on a single mag card, the payoff in optimization is in the reduction of run time.

With the advantage of having Jared's and Lou's programs in hand, I wrote a third one that runs in just three fourths the time required by either of the others (26 minutes versus 34-35 to print 12 months). The analysis and comparisons which follow are intended to be informative and constructive, and I hope it goes without saying that it is not my intention to impugn either Jared's or Lou's programming abilities. In the same spirit, I welcome better approaches from others who now have all 3 programs to start with, and it is my hope that we will all learn something from this exercise.

In addition to the usual user instructions and program listings, I am including high-level algorithms, which may make it easier to see how each program works. The most critical function to try to optimize is the generation of the print code required to print the successive integers 1,2,...31. Jared's approach to this requirement is the slowest, partly because he needed a routine to translate 4-digit year numbers besides the day numbers. Lou dodged this requirement by letting the printer print the year as the displayed value, with the month tag-printed in the same line via Op 4 Op 6. But I expect that a routine optimized to only translate integers in the 1-31 range into print code would still run slower than a print-code counter, Lou devised just such a counter, and this difference appears to give his program a slight speed advantage. But both Jared and Lou use a number of x-t comparisons where they could have used Dszs (which are faster) for branch control in their print code generators, and Lou uses 4 flags, 3 of which require repeated setting and resetting, as well as testing.

While inefficient approaches to meeting requirements posed only once per printed month aren't so critical as the once-per-day ones, the

The SR-52 Users Club is a non-profit loosely organized group of TI PPC 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, Inc. Membership is open to any interested person: $6.00 includes six future issues of 52-NOTES; back issues start June 1976 @ $1.00 each.

extra execution time can add up. Both Jared and Lou constructed independent sequences to calculate the number of days in a month, since it apparently didn't occur to them that given the first day of 2 successive months, ML-20 returns the number of days in the first month. There is also some time lost putting month, day and year information into the required ML-20 input format, instead of directly storing the separate values in Reg 1, 2, and 3, and processing via SBR 086 or 177 (see V2N11p1 and Fred Fish's ML Survival Kit p 20-1).

So try all 3 programs, look them over, and see if you can write a faster one (one which runs say, under 2 minutes per month). Incidently, although the ML manual doesn't say so, 9999 appears to be the upper limit on the year, unless inputs are stored separately. But at some point the 366-day year every 4 years rule won't hold.

TI-59/PC-100A Program: Calendar Printer   Jared Weinberger (221)

Algorithm:

1. Print month header.
2. Call ML-20 D with 1st day of specified month and year to get day of week; convert the returned 1,2,...6,0 to 1,2,...7.
3. Determine number of days in month:
   For Feb: Call ML-20 A with Feb 28th, and ML-20 B with March 1st to get the number of days between Feb 28th and March 1st; use this result to get the number of days in Feb. For other months: Test for April, June, Sept, Nov to get the number of days.
4. Fill a cleared block of 7 registers with consecutive base ten integers beginning with 1 in a position determined by the corresponding day of the week.
5. Convert the stored integers to print code via a translator; pack print buffers, and print a line of days.
6. Clear the integer block, fill with the next group of integers, and go to step 5.
7. Repeat steps 5 and 6 until all days have been printed.
8. For whole year: Increment month, and repeat steps 1-7 until all months have been printed.

User Instructions: For one month: key month, press A; key year, press B; press D. For whole year: Key year, press B; press E.

Program Listing:

```
000:  LA S29 rtn LB S30 rtn CP x=t 036 ÷ log Int S8 Op28 INV log +
024:  xXt 100 Prd 21 8 xGEt 036 2 + 1 - Int SUM21 = X 10 Dsz8 023 CLR
050:  Exc21 xXt 1 x=t 059 xXt rtn 0 rtn R1 SBR 010 X 1 EE 6 = INV EE
073:  S28 R2 SBR 010 + R28 = Op1 R3 SBR 010 X 1 EE 4 = INV EE S28 R4 ÷
103:  10 = Int SBR 010 + R28 = Op2 R4 ÷ 10 = INV Int X 10 = SBR 010 S0
133:  R4 CP x=t 148 R0 INV x=t 148 + 1 = EE 8 S28 R5 SBR 010 X 100 +
162:  R28 = Op3 R6 SBR 010 X 1 EE 6 = INV EE S28 R7 SBR 010 + R28 = Op4
192:  Op5 rtn LD Op0 R29 + 34 = S28 R*28 Op1 R30 SBR 010 Op4 Op5 Adv R31
223:  Op1 R32 Op2 R33 Op3 R34 Op4 Op5 Adv R29 X 2 INV log + R30 ÷ 4 INV
252:  log + 1 = Pgm20 D CP INV x=t 265 7 S10 R29 xXt 2 INV x=t 316 R30
277:  ÷ 4 INV log = S0 + 228 = Pgm20 A R0 + 301 = Pgm 20 B Pgm20 C xXt
306:  1 x=t 311 + 28 GTO 340 R29 xXt 4 x=t 338 6 x=t 338 9 x=t 338 11
333:  x=t 338 1 + 30 + 1 = S11 Pgm 1 SBR CLR 8 xXt 1 S13 R13 S*10 1
359:  SUM10 SUM13 R10 INV x=t 354 SBR 061 0 S7 Pgm 1 SBR CLR 1 S12 R11
384:  xXt R13 x=t 413 8 xXt R13 S*12 1 SUM12 SUM13 R12 INV x=t 382 SBR
408:  061 GTO 372 SBR 061 Adv Adv rtn LE 0 S29 1 SUM29 SBR 197 12 xXt
433:  R29 INV x=t 424 Adv Adv rtn
```

Prestored Data:

```
31:  3641003032 37410043 1700372300 2135003613 2513314000 2117144000
37:  3013351523 1333352427 3013450000 2541311700 2541274500 1341224000
43:  3617333740 3215374000 3132424000 1617154000
```

TI-59/PC100A Program Calendar Printer — Lou Cargile (625)

Algorithm:

1. Determine whether to print 1 month or balance of specified year.
2. Call ML-20 D with 1st day of the specified month and year to get day of week for 1st day; convert returned 1,2,...6,0 to 1,2,...7.
3. Check for leap year, Feb, odd or oven month, July, Aug, to get number of days per month.
4. Print month header.
5. Use the calculated day of week to determine where to begin filling print buffers from a quasi base seven counter; print filled buffers and refill from the counter until all days have been printed.
6. For more than one month: Repeat steps 2-5, incrementing the month until all months for the specified year have been printed.

User_Instructions: For one month: Key MM.YYYY, press B; for balance of year: Key MM.YYYY, press A.

Program Listing:

```
000: 12 xXt R8 INV xGEt 010 R/S 1 SUM6 GTO 026 LA INV Stflg4 INV Stflg
023: 0 S6 R6 S0 Int S8 INV SUM0 X 100 + 1 = SUM0 R6 INV Int X 10000 =
056: S10 R0 Pgm20 D S07 0 xXt R7 INV x=t 076 7 SUM7 R10 ÷ 4 = INV Int
083: INV x=t 089 Stflg0 31 S13 0 xXt R8 ÷ 2 = INV Int INV x=t 134 R8
108: xXt 2 INV x=t 129 29 S13 INV Ifflg0 145 1 SUM13 GTO 145 7 xXt
131: GTO 138 R8 xXt 7 INV xGEt 145 1 SUM13 R8 + 47 = S9 Op0 R*9 Op4
159: R10 Op6 SBR368 R46 Op1 R45 Op2 R44 Op3 R43 Op4 Op5 SBR369 INV
188: Stflg1 INV Stflg3 1 S00 0 S14 S16 5 S15 INV Stflg2 R14 SBR289 R14
214: SUM16 SBR236 Ifflg2 204 SBR236 Ifflg2 204 SBR236 GTO 204 Dsz15
238: 275 R16 Int Op*0 Op20 R0 xXt 5 INV x=t 266 1 S0 Op5 Stflg2 Ifflg3
264: 281 R16 INV Int S16 5 S15 100 Prd16 rtn Adv INV Ifflg4 288 R/S
288: RST Dsz7 338 Ifflg1 304 .01 SUM14 Stflg1 .01 SUM14 INV Dsz13 332
314: R14 INV Int xXt .08 x=t 339 .13 x=t 345 rtn 0 S14 Stflg3 rtn rtn
339: .02 SUM14 rtn .88 SUM14 R14 Int xXt 1 x=t 359 rtn SUM14 rtn LB
364: Stflg4 GTO 021 R47 Op1 Op2 Op3 Op4 Op5 rtn
```

Prestored Data:

```
43:  21000036 4300003700 370000 36000030 2020202020 25133100 21171400
50:  30133500 13333500 30134500 25413117 25412745 13412200 36173337
57:  32153700 31324200 16171500
```

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

TI-59/PC-100A Program: Calendar Printer — Ed

Algorithm:

1. Check for ML module presence; alert user with flashing display if absent.
2. Determine user's choice: 1 month, balance of start year, or print from start month and year until end of specified year.
3. Call ML-20 SBR 086 with 1st day of specified month and year to get corresponding absolute day, and ML-20 SBR 177 to get corresponding day of week. Call ML-20 SBR 086 with 1st day of next month and calculate difference between returned absolute day value, and first one to get number of days in specified month.
4. Print month headers.

5. Use calculated day of week to initialize a general purpose print-buffer-packer routine to print first line of days, followed by successive lines until all days have been printed. Use a quasi base seven counter to supply the print-buffer-packer routine with code for successive integers.
6. For more months, as determined by step 2, cycle months, and increment years as required; repeat steps 3-5 until done.

User Instructions:

1. Initialize: Press A, see module confirmation printed; flashed display indicates wrong module.
2. Key start year, press R/S; key start month, press R/S.
3. For 1 month: Press B;
   For balance of year: Press C; for specified period: Key end year, press D.

Program Listing:

```
000:  LC' 5 S17 1 EE 8 S19 CLR R21 Op*20 0 S21 1 Sum20 Dsz18 037 1 S20
028:  Op5 Op0 Stflg0 4 S18 rtn LE' Op28 INV Dsz11 053 INV Dsz12 066 rtn
053:  1 S8 Op29 10 S11 8 S12 rtn 2 SUM08 9 S12 rtn LD' INV Stflg0 CLR 1
080:  xXt E' R9 x=t 093 X R19 = SUM21 100 INV Prd19 INV Dsz17 145 R8 X
107:  R19 = SUM21 100 INV Prd19 Dsz15 128 0 S18 C' Adv rtn INV Dsz17 149
133:  100 INV Prd19 Dsz17 D' C' GTO D' C' GTO 104 C' Ifflg0 D' GTO 133
156:  LA' Op00 1 S8 S09 10 S11 8 S12 5 S17 4 S18 1 EE 8 S19 1 S20 0 S21
189:  R16 xXt 1 x=t 232 2 x=t 234 3 x=t 244 4 x=t 260 5 x=t 282 6 x=t
214:  274 2 S17 1 S18 100 S19 4 S20 D' rtn D' rtn 2 S17 100 S19 D' rtn
244:  4 S17 3 S18 1 EE 6 S19 2 S20 D' rtn 1 S17 3 S18 1 S19 2 S20 D'
273:  rtn 1 S18 4 S20 D' rtn 3 S17 2 S18 1 EE 4 S19 3 S20 D' rtn LA 1
301:  xXt Pgm1 SBR Write x=t 312 Op55 R/S Adv Adv Adv R/S S13 R/S S14
321:  R/S LB R14 S01 1 S2 R13 S3 Pgm20 SBR086 S10 Pgm20 SBR177 S16 R14
351:  + 1 = S1 R13 S3 Pgm20 SBR086 - R10 = S15 Op0 R14 + 47 = S20 R*20
383:  Op4 R13 Op6 R47 Op1 R46 Op2 R45 Op3 R44 Op4 Op5 A' rtn LC 13 -
414:  R14 = S22 B 1 SUM14 Dsz22 419 CLR rtn LD - R13 + 1 = S23 C 1 SUM13
443:  S14 Dsz23 439 Adv Adv R/S
```

Prestored Data:

```
44:  2135003613 1700372300 37410043 3641003032 251331 211714 301335
51:  133335 301345 25413117 25412745 134122 36173337 321537 313242
59:  161715
```

- - - - - - - - - -

PROGRAM/ROUTINE COPYRIGHT

Legal and legislative bodies are recognizing that it is difficult to apply copyright law (even the new one) to computer-readable works, so it is not surprising that we laymen are confused as to what is and isn't allowed. Ken Widelitz, a lawyer who writes the Legal/Business Forum in KILOBAUD magazine, addresses this topic (Nov 77 p14 and Apr 78 p6) in some detail. An important concept which Widelitz touches on is the so-called Fair Use Doctrine, which effectively says that the less you copy, and the farther removed your use is from competing in the marketplace with the original, the surer you can be that you are acting within the law. An important widely accepted legal interpretation says that ideas themselves are not copyrightable; the expression of them is. Then there is the question of "derivative" works. Widelitz notes that there is no clear-cut dividing line between very basic techniques (such as commonly used sorting routines) which are generally held to be uncopyrightable, and more specialized ones which are, and recommends that users seek legal advice for important specific cases.

Interested members will find Widelitz' 2 articles both informative and thought provoking, and may find helpful a new National Bureau of Standards publication (#500-17, $4.00 USGPO Wash DC 20402): "Copyright In Computer-Readable Works: Policy Impacts of Technological Change".

Dave Johnston (5) and Lee Eastman (713) wonder if translations of programs for different machines are protected by original copyrights. I suggest applying the Fair Use Doctrine, and if it appears that your intentions might conflict significantly with the originator's interests, get legal advice. Members who have been involved in computer-readable copyright litigation are invited to share their experiences.

52-NOTES SUBJECT INDEX

Many of you have suggested that it would be helpful to have a topical index, and I agree, except perhaps as to how it should be generated and maintained. To a large extent, the choice of alphabetically listed names to categorize published material is subjective, and I suspect that the best index is generated by the user himself, and maintained as a continuously updated card file. Getting started is a bit of a chore, but refamiliarizes the user with the published material, as well as providing him with a relevant reference. However, for those who don't want to bother, here is an index which I have incorporated into a new Club brochure to help prospective members assess 52-NOTES, and identify back issues of interest. It covers V1n1 through V3N4, with the V, N, and p designators omitted: ABSOLUTE ADDRESSING 115, 173, 282, 2114, 333, 336; ADVANCED PROGRAMMING TECHNIQUES 135, 154, 281, 2112, 343; ALGORITHMS 145, 154, 164, 222, 254, 263, 293, 296, 2103, 2105, 323, 334, 342; AOS 134, 175, 232, 235, 2125, 312; BOOK/PERIODICAL REVIEWS 141, 152, 171, 216, 224, 232, 236, 244, 256, 265, 274, 2124, 331, 345; BUILT-IN FUNCTIONS 145, 163, 174, 221, 266, 273, 276, 282, 285, 296, 2116, 2121, 315, 321, 335, 345; CLUB NEWS/EDITORIALS 111, 126, 151, 164, 241, 246, 271, 274, 283, 2101,2124, 2126, 313, 345, 346; CROMS 261, 275, 286, 2103, 2105, 2111, 2122, 2124, 313, 315, 335; DIAGNOSTICS 144, 173; DISPLAY (non-fractured) 121, 136, 156, 222, 274, 2124, 314, 334; DYNAMIC CODE GENERATION 123, 135, 344; ERROR STATES/PRODUCERS 112, 162, 174, 254, 266, 282, 2113, 2124, 313; FIBONACCI NUMBERS 113, 222, 313, 333; FLAGS 123, 162, 233, 2102, 336; FRACTURED DIGITS/DISPLAY 113, 125, 146, 163, 166, 175, 222, 225, 234, 251, 2123, 311, 313, 316, 332, 346; GAMES 112, 116, 132, 143, 146, 243, 265, 2106, 325, 346; HARDWARE 122, 136, 144, 156, 161, 173, 174, 212, 216, 221, 225, 236, 253, 264, 2123, 331, 334, 336, 346; INDIRECT ADDRESSING/POINTERS 131, 154, 164, 215, 224, 276, 283, 315, 323; INPUT/OUTPUT 113, 115, 135, 215, 223, 236, 263, 281; 293, 295, 2102, 2103, 2106, 2126, 332, 334; INTEGER/FRACTION TRUNCATION 113, 124, 136; INTERRUPTS 123, 176, 222, 314, 325, 336; LABELS 115, 141, 173, 211, 254, 283, 286, 2114, 326, 332, 346; MAG CARD READ/WRITE 124, 134, 144, 156, 294, 2105, 2122, 314, 322, 335; MATRIX CALCULATIONS 134, 214, 223, 255, 283, 2125; MULTIPLE PRECISION 255, 2101 2121, 312, 313; NOTATION/CONVENTIONS/DEFINITIONS 126, 141, 161, 171, 226, 265, 292, 333; OPTIMIZATION 173, 2115, 2122, 222, 312, 321, 335; PARTITIONING 276, 283, 292, 2104, 2105, 2122; PRINTER 131, 136, 222, 284, 293, 295, 2102, 2105, 2121, 2126, 314, 321, 326, 346; PRODUCT REVIEWS 161, 171, 226, 253, 261, 331; PROGRAM EXCHANGE (CLUB) 166, 246, 256, 275, 2116, 326; PSEUDOS/HIRs 124, 153, 173, 211, 245, 264, 274, 291, 2106, 2122, 315, 326, 336; RANDOM NUMBERS 231; REGISTER ARCHITECTURE 114, 121, 145, 163, 165, 172, 174, 213, 226, 254, 273, 2115, 335; ROUNDING/PRECISION 162, 171, 175, 221, 266, 2102, 341; WIPE-OUT/CRASH 122, 152, 163, 174, 175, 274.

PERIODICAL REVIEW: TI-59 NEWSLETTER, edited by Howard Gosman (754), monthly since Nov 77, $24 per year.

Howard and a few associates have launched this newsletter as a business venture (not a users club), claiming to target it to the non-mathematician, "average" PPC user. The content and style of the first 5 issues characterize this periodical as oriented toward the non-programmer interested in business, finance, and gambling. Except for a "beginners corner" which has addressed the use of built-in statistics functions, there is little in the way of how-to programming tutorials or other technical discussion. There is a fair amount of non-59 and even non-TI PPC coverage over a broad range of generally non-programming topics. Over 2000 subscribers are claimed, with growth to 5000 anticipated.

TIPS AND MISCELLANY

56/57/58 Program Exchange: Dave Johnston (5) has a new catalog dated 1 May 78 listing 32 math, 9 statistics, 10 operations research, 25 physics, 12 other physical sciences, 30 engineering and technology, 3 life and behavioral sciences, 13 games, 7 finance, and 1 general info-test routine... programs written for the SR-56, of which 47 have been translated for the 57, and 28 for the 58. 29 additional programs were written for only the 57; 32 for the 58. Send Dave 20¢ and a SASE for a copy of his catalog.

CROM Download into Used User Memory: A CROM download (Op 9) alters only the steps which it fills, leaving other steps as they were before the download. While the presence of user code remnants is unlikely to affect execution of the downloaded CROM code, it can be a source of confusion when the CROM code is listed. Fred Fitzgerald (252) suggested (later confirmed by Tom Cox) that just such a situation caused the Leisure Library-13 HIR confusion (V3N1p5, V3N2p6, and V3n3p6): The HIR instruction at step 240 was a user-code remnant, not part of the downloaded CROM. The safest way to avoid such confusion is to cycle the power switch before downloading.

TI Service Manuals and Parts: David Swindell (877) reports that detailed service manuals and some parts are available for the older TI machines from Bootstrap Enterprises, Box 614 Richardson, TX 75080.

More on Memory Add-On (V3N3p4): Simon Hughes (837) reports that he has successfully modified his 58, and has volunteered assistance to Toronto-area members who need help. Simon's business phone is (416) 294-9838.

EE and Eng (57,58,59): Fred Fish (606) notes that INV Eng returns either an EE or Eng display (hard or soft) to a turn-on display immediately, while INV EE on a soft EE display doesn't remove the EE format until another key is pressed. As the manual notes (V-8), the Eng format continues to prevail following INV EE or CLR on an Eng display.

CROM Use of Flags For Print Suppression: Roy Chardon (515) noticed that RE-2 uses flag 0 to suppress printing, but that no mention of this is made in the RE manual. Further investigation reveals that RE-2 never sets flg 0, but that RE-3 and RE-13 do, prior to calls to RE-2 to bypass a Prt which would otherwise execute with printer connection. Similarly, RE-4 uses flag 5 to suppress printing, but never sets it. RE-13 sets flag 5 in the same sequence in which it sets flag 0 on the call to RE-2, which in turn calls RE-4. Other members finding similar flag use in CROM code are invited to share their discoveries, to help minimize the unsuppressable printing problem, which is often a deterrent to CROM use by user programs.