

HP Forum Archive 15

[[Return to Index](#) | [Top of Index](#)]

S&SMC #9, Act II

Message #1 Posted by [J-F Garnier](#) on 26 May 2005, 7:26 a.m.

Valentin wrote:

"... the number of solutions seems to grow forever that's not the case. Actually, 2492 is the maximum number of solutions for any N (in this case, for N=9 and N=10). After that (N=11, 12, etc), the number of solutions decreases steadily till N=25, which again has a unique, 25-digit solution".

My personal challenge was then to find this unique solution on the HP-71B (actually using Emu71). I wrote a program (non-recursive, with no GOSUB) that solves the problem in less than 40 seconds with Emu71 in Fast mode. I will post my solution in a few days to let you play with this challenge, if you like it.

J-F

Note: Emu71 Fast mode is just Emu71 with /fx option, e.g. "Emu71 /f50" with a 1.7GHz processor, or even "Emu71 /f80" with a 2.3GHz processor (close to a virtual Saturn running at 80MHz...).

Emu71 /f50 ?

Message #2 Posted by [Valentin Albillo](#) on 26 May 2005, 7:32 a.m.,
in response to message #1 by J-F Garnier

Hi, Jean-François:

J-F posted:

"Note: Emu71 Fast mode is just Emu71 with /fx option, e.g. "Emu71 /f50" with a 1.7GHz processor, or even "Emu71 /f80" with a 2.3GHz processor (close to a virtual Saturn running at 80MHz...)." "

I didn't know that you could use a number after the "/f", I was using just "Emu71 /f", per se, no number after.

What does this number mean and how can you deduce its optimum value for a given processor and/or processor speed ?

As for your "personal challenge", that was my intention when mentioning that 25-digits had just an unique solution, that someone would "take the bait" and would decide to try and find that solution. I'm glad that you did, and even more that you used Emu71 to do it, with such good timing.

Best regards from V.

Emu71 Fast Mode - explanation

*Message #3 Posted by **J-F Garnier** on 26 May 2005, 8:02 a.m.,
in response to message #2 by Valentin Albillo*

The number after the /f option set the size of the emulated opcode 'burst' before checking keyboard, updating display and timers and so on. As Emu71 is basically a DOS program, these calls to the OS I/O primitives also set the relative CPU usage given by the Windows OS due to the automatic idle detection. Max value is 255, but if you set a too high value, Emu71 realtime clock will not be emulated correctly.

There is no direct relation with the equivalent Saturn frequency, I said that /f80 was close to a 80MHz Saturn, but it's only by chance.

A convenient way to optimize the parameter for a particular machine is to use the CPU usage display (in the Task Manager utility), and to target about 80% CPU usage.

J-F

Valentin: you may notice that the speed-up is not effective in your last challenge solution. This is due to the CALL function, I don't know why right now and I'm working to correct this. That's why I don't use recursive calls in Emu71 for 'force brute' approach. BTW such challenges are very usefull to test emulator behaviors...

Re: Emu71 Fast Mode - explanation

*Message #4 Posted by **J-F Garnier** on 26 May 2005, 8:44 a.m.,
in response to message #3 by J-F Garnier*

Correction:

With /f50 on a 1.7GHz processor the speed-up is actually about x140, measured by the following test program:

```
240 SUB SPEED
250 ! test emulator speed
260 ! a HP71B needs about 20 s for 2000 iterations
270 N=2000
275 T=TIME
280 FOR I=1 TO N @ NEXT I
```

```

290 T=TIME-T @ IF T<2 THEN N=N*2 @ GOTO 275
295 DISP T;"s. SPEED=";IP(N/T)/100;"TIMES HP-71B SPEED"
300 END SUB

```

J-F

Re: Emu71 Fast Mode - explanation

Message #5 Posted by [Valentin Albillo](#) on 26 May 2005, 9:05 a.m.,
in response to message #4 by J-F Garnier

Thanks a lot for the detailed explanation, Jean-François ! :-)

I'm truly glad that you find my challenges useful and interesting. Certainly, were it not for your [Emu71](#) and kind answers to my requests, many of my 71B-related articles and challenges would've never seen the light.

Being able to use [Emu71](#) for development and testing purposes increases productivity a hundredfold, which, with the very little free time I can gather for these activities, is absolutely of the essence.

Best regards from V.

S&SMC #9, Act II - Solution and RPL Programming Challenge

Message #6 Posted by [J-F Garnier](#) on 29 May 2005, 3:03 p.m.,
in response to message #1 by J-F Garnier

Below is my solution. First, I noticed that to be divisible by 25, a number should end with 00,25,50 or 75 so it is enough to find all the 24-digit solutions. This means that double length arithmetic can be used on Saturn (and Capricorn...) machines (2*12 digits), with only the implementation of MOD, *10 and div10 operations, which is quite easy to do.

```

10 ! S&SMC #9, Act II
20 ! JFG, May 2005
30 H=24 ! # of digits (24 max)
40 C=0 ! solution counter
50 FOR I=10 TO 98 STEP 2 ! loop for 1st and 2nd digit positions
60 L=3 ! start tests from 3rd position
70 M=I*10 ! lower part of candidate number
80 N=0 ! higher part
90 'A': X=MOD(N*1.E+12,L)+MOD(M,L) @ IF X>=L THEN X=X-L ! x= [n,m] mod L
100 IF X<>0 THEN X=L-X ! find Lth digit
110 IF X>9 THEN 'D' ! is it a possible candidate?
120 M=M+X ! yes, built it.

```

```
130 IF L=H THEN C=C+1 @ DISP N;M @ GOTO 'C' ! found one!
140 'B': N=N*10+M DIV 10000000000 @ M=MOD(M,10000000000)*10 ! go on [n,m]*=10
150 L=L+1 @ GOTO 'A'
160 'C': X=X+L @ IF X<=9 THEN M=M+L @ GOTO 'B' ! next try, if possible
170 'D': M=MOD(N,10)*10000000000+M DIV 10 @ N=N DIV 10 ! else [n,m]/=10
180 X=MOD(M,10)
190 L=L-1 @ IF L>2 THEN 'C'
200 NEXT I
```

The only three 24-digit solutions are:

```
144408645048  225636603816
360852885036  840078603672
402852168072  900828009216
```

So the unique 25-digit solution is: 3608528850368400786036725.

You can notice that I used a few GOTOs and 4 labels. Actually, I'm a supporter of structured programming style with WHILE/LOOP/REPEAT/etc structures (such as in the HP71 JPCROM), so it can be a challenge to rewrite the program with no explicit GOTO.

Will someone implement this simple but non-trivial algorithm on HP48's RPL for instance?

J-F

Re: S&SMC #9, Act II - Solution and RPL Programming Challenge

Message #7 Posted by [Raymond Del Tondo](#) on 29 May 2005, 9:59 p.m.,
in response to message #6 by J-F Garnier

Hi J-F,

nice exercise.

For an HP-48 (or generally RPL) solution
I think the runtime structure will have to be rewritten,
as you do jumps into and out of 'loops' in your BASIC program.

My first steps would be maybe reordering
the test logic to omit labels 'C' and 'D',
and in consequence omitting label 'B'.

The following code leaves out some assignments,
and maybe has still some logic error inside,
but it is intended only for showing the principle
how it could be solved completely w/o GOTO;-)

However, it's deep in the night and I'll go to sleep...

Regards

Raymond

```

:: %24 'H' STO %0 'C' STO
NINEYEIGHT TEN
DO
  :: %3 'L' STO INDEX@ #10* UNCOERCE 'M' STO %0 'N' STO

* 'A' -Loop:
  :: BEGIN
  ::
* some code
  X %9 %> ITE
  :: (this seco was label 'D')
  L %2 %> (leaves flag on stack for 'C')
  TRUE (leaves flag on stack for 'A' test)
  ;
  :: M X %+ 'M' STO
  L H %= NOTcaseTRUE ('GOTO C')

  C %1+ 'C' STO
  FALSE (SKIP C)
  FALSE (leaves flag on stack for 'A' test)
  ;

  caseTRUE (leaves flag on stack for 'A': Exit 'A')

  ITE
  :: (this seco was label 'C')
  X %9 %> (leaves flag on stack for 'B')
  ;
  TRUE

```

```

                ?SKIP
                :: (this seco was label 'B')
                  L %1+ 'L' STO
                ;

                FALSE
                ;
                UNTIL
                ;

                INDEX@ #1+ INDEXSTO
                ;
LOOP
;

```

Re: S&SMC #9, Act II - Solution with JPCROM

Message #8 Posted by [J-F Garnier](#) on 31 May 2005, 11:39 a.m.,
in response to message #6 by J-F Garnier

Here is my structured programming style solution, using the HP-71B JPCROM:

```

10 ! S&SMC #9, Act II
20 ! JFG, May 2005 - with JPCROM
30 H=24 ! # of digits (24 max)
40 C=0 ! solution counter
50 FOR I=10 TO 98 STEP 2 ! loop for 1st and 2nd digit positions
60   L=2 ! start of search
70   M=I ! lower part of candidate number
80   N=0 ! higher part
90   REPEAT
100    REPEAT
110     N=N*10+M DIV 10000000000 @ M=MOD(M,10000000000)*10 ! go on [n,m]*=10
120     L=L+1
130     X=MOD(N*1.E+12,L)+MOD(M,L) @ IF X>=L THEN X=X-L ! x= [n,m] mod L
140     IF X<>0 THEN X=L-X ! find Lth digit
150     IF X<=9 THEN M=M+X ! if possible candidate, build number
160   UNTIL X>9 OR L=H ! until no more possible, or solution reached
170   IF X<=9 AND L=H THEN C=C+1 @ DISP N;M ! found one!
180   REPEAT
190     M=MOD(N,10)*10000000000+M DIV 10 @ N=N DIV 10 ! go back [n,m]/=10

```

```
200      X=MOD(M,10)
210      L=L-1
220      X=X+L ! next try
230      UNTIL X<=9 OR L=2 ! until a possible try, or end of search
240      M=M+L
250      UNTIL L=2 ! until end of search
260      NEXT I
```

The algorithm is now much easier to read and understand, at the expense of some redundant tests.

Nice exercise, indeed.

J-F

Yep, the JPC rom is a jewel - every 71b owner should have it!

*Message #9 Posted by [Gene](#) on 31 May 2005, 12:01 p.m.,
in response to message #8 by J-F Garnier*

Wonderful lex file.

Gene

Re: S&SMC #9, Act II - Solution with JPCROM

*Message #10 Posted by [Valentin Albillo](#) on 31 May 2005, 12:13 p.m.,
in response to message #8 by J-F Garnier*

Hi, J-F:

Excellent solution to the extended version of my S&SMC#9. Frankly, I didn't expect anyone at all to come up with the unique 25-digit solution but then I wasn't expecting you to try your hand at it, either ! :-)

By the way, despite your titanic effort the topic isn't exhausted yet. What about bases other than 10 ? ;-)

Glad that you enjoyed my challenge, hope to see you attempt the next one as well.

Best regards from V.

Re: S&SMC #9, Act II - Solution with JPCROM

*Message #11 Posted by [Raymond Del Tondo](#) on 31 May 2005, 1:22 p.m.,
in response to message #8 by J-F Garnier*

Hi J-F,

did you see my last post?

Your today's solution will be even easier to port to RPL;-)

Regards

Raymond

Re: S&SMC #9, Act II - Solution with JPCROM

*Message #12 Posted by [J-F Garnier](#) on 31 May 2005, 3:20 p.m.,
in response to message #11 by Raymond Del Tondo*

Hi Raymond,

Yes I read your post, this was one reason to me to write a GOTO-less version (the other reason was to demonstrate the JPCROM powerfull features).

But User RPL is good enough for this problem :-)

J-F

Edited: 31 May 2005, 3:25 p.m.

[[Return to Index](#) | [Top of Index](#)]



[Go back to the main exhibit hall](#)