



HP Forum Archive 11

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

Short & Sweet Math Challenges #6

Message #1 Posted by [Ex-PPC member](#) on 3 Mar 2003, 7:11 a.m.

Just to light up a little this week, here's a new Short & Sweet Math Challenge (tm), for you to try on your favorite HP calc [NOT-NOT-NOT on your PC and/or Mac, that's NOT the point :-)]

The Challenge

Find a 10-digit integer number ABCDEFGHIJ where all its digits A,B,...,J must be different, such that A is divisible by 1, AB is divisible by 2, ABC is divisible by 3, and so on.

For instance, 1234567890 isn't a solution, because 1 is divisible by 1, 12 is divisible by 2, and 123 is divisible by 3, but 1234 is *not* divisible by 4.

Requirements

You may try it in any HP programmable calculator; the shortest and more elegant the program (within the selected model's capabilities, of course), the better.

Solutions and Generalization

Next week I'll provide a short program for the HP-71B that solves this challenge, as well as another program to solve the more generalized challenge of finding all numbers N, from 1-digit numbers onwards, allowing for repeated digits, that have the property that their first M-digits (from the left) form a number divisible by M (for instance, 123 would be such a solution for M=3, but 1234 wouldn't for M=4).

If you would like to tackle this generalized challenge, try to write a short program which will find and optionally print out solutions, and see if you can establish whether there are a finite or infinite number of solutions, and in the former case what's the maximum value for M, and how many solutions are there having 1,2,..., M digits.

Re: Short & Sweet Math Challenges #6

Message #2 Posted by [Patrick](#) on 4 Mar 2003, 2:34 p.m.,
in response to message #1 by [Ex-PPC member](#)

Here is my "first" solution. I've tried to use only features common to a bunch of HP programmables (e.g., didn't use MOD function, no recall arithmetic). This works almost verbatim on my HP-12C for example, which has one of the simpler programming models.

I haven't thought about it too hard yet, but I think this program produces the smallest number having the desired properties: 1,020,005,640.

BTW, the innocent looking subtraction in step 010 is, I think, critical to this algorithm working for all ten digits. Otherwise, it works only up to 9 digits.

Change the number in step 2 to one less than the number of digits to produce. In this version, I used 9 to produce a 10 digit result. Alternatively, have the number of digits you want in the X register when the program starts and replace the first two steps by the three steps 1, -, 10^x .

```
001 EEX 002 9 003 ENTER 004 1 005 STO 0 006 10 007 * 008 1 009 STO + 0 010 - 011 RCL 0 012 + 013 LAST X 014 / 015 INT 016 RCL 0 017 *
018 x<=y? (can substitute x<y? here if you like) 019 GTO 006 020 RTN
```

Re: Short & Sweet Math Challenges #6

Message #3 Posted by **Patrick** on 4 Mar 2003, 2:35 p.m.,
in response to message #2 by Patrick

Apologies for how the program listing turned out... I had them all on separate lines before I posted but they all got glued together onto one line. You can make it out with line numbers 001, 002, ... etc.

Re: Short & Sweet Math Challenges #6

Message #4 Posted by **Ex-PPC member** on 4 Mar 2003, 2:44 p.m.,
in response to message #2 by Patrick

Thanks for your very interesting program, Patrick, and yes, your result is correct, except for the fact that the original challenge stated:

"Find a 10-digit integer number ABCDEFGHIJ *where all its digits A,B,...,J must be different*, such that A is divisible by 1, AB is divisible by 2, ABC is divisible by 3, and so on."

i.e.: the solution must have ten digits and all of them must be unique, no repeated digits. So, each digit 0,1,2,...,9, must appear *once and only once* in the number.

See if you can solve it subject to that restriction :-)

Re: Short & Sweet Math Challenges #6

Message #5 Posted by **Patrick** on 4 Mar 2003, 3:26 p.m.,

in response to message #4 by Ex-PPC member

Oops.

I'm on it.

Re: Short & Sweet Math Challenges #6

*Message #6 Posted by **Frédéric** on 5 Mar 2003, 4:29 a.m.,
in response to message #4 by Ex-PPC member*

What we can say

For ABCDEFGHIJ to be divisible by 10 => J must be equal to 0

So the problem is now for ABCDEFGHI.

For ABCDE to be divisible by 5 => E must be equal to 5

Now for the reason of odd and even numbers :

A,C,G,I are in {1,3,7,9}and

B,D,F,H are in {2,4,6,8}

So we must have $4! \cdot 4!$ combinations : 576 possibilities

This is my very little contribution.....

PS : I really like these S&SMCs ... but my old brain is really tired.

Re: Short & Sweet Math Challenges #6

*Message #7 Posted by **Michael F. Coyle** on 11 Mar 2003, 5:18 p.m.,
in response to message #1 by Ex-PPC member*

I know it's kind of late in the game, but I do have a solution to this problem, written for the 48GX. I have to leave in a few minutes so I don't have time to post the code but I will send it to anyone interested. A brief description follows.

It solves the more general problem of finding all of the 1, 2, 3, ..., 10 digit numbers with the required properties (first N digits divisible by N, no repeated digits). It starts with a list of all the valid one-digit answers (1..9). For each of these, it forms trial 2-digit numbers by tacking on each of the digits 0..9 and seeing if the resulting 2-digit number meets the criteria; if so, it is added to a list of 2-digit solutions.

This is then repeated with the 2-digit numbers to get 3-digit solutions, and so on. This is all done in a big loop, of course. (CS fans will recognize this as a "breadth-first search.")

The program takes about 25 minutes to run and produces a list of 10 lists, giving all the solutions.

The number of solutions for each size number starts at 9 for one digit and increases to 220 for 5-digit numbers and then decreases down to just one 10-digit answer. (I will not include it here in case anyone else is still working on it.) I manually verified that the 10-digit answer and its smaller prefixes met both criteria.

Hopefully this will still be of interest to someone. For me, the fun was just getting a working program.

S&SMC#6: An HP-71B solution

Message #8 Posted by *Ex-PPC member* on 13 Mar 2003, 6:37 a.m.,
in response to message #7 by Michael F. Coyle

Michael wrote:

I know it's kind of late in the game, but I do have a solution to this problem, written for the 48GX [...] Hopefully this will still be of interest to someone. For me, the fun was just getting a working program.

Thanks for your interest in my challenge, perhaps it was a little more difficult than the usual "do this simple thing in the least possible number of programming steps", but it certainly offered an opportunity to show off one's favorite HP calc and some advanced programming techniques, right ?

As promised, here's a solution to both the original and the extended challenge, for the HP-71B. As usual, it's far from optimal, but it can be written very easily and gets the job done quickly:

This short (164 bytes) program recursively finds all numbers that have the property that the number consisting in the first N leftmost digits is divisible by N, for N ranging from 2 to 9 digits. Repeated digits are allowed for this version.

```
10 DESTROY ALL @ OPTION BASE 0 @ DELAY 0,0 @ STD
20 FOR I=1 TO 9 @ CALL TST((I),2,9) @ NEXT I @ DISP "DONE!" @ END
30 SUB TST(N,D,M) @ N=10*N @ FOR I=0 TO 9 @ X=N+I @ IF MOD(X,D) THEN 50
40 DISP D;X @ IF D#M THEN CALL TST(X,D+1,M)
50 NEXT I @ END SUB
```

It simply starts from 1-digit numbers and calls a subprogram that adds a new digit, testing all 10 possibilities for divisibility. When a suitable candidate appears, the subprogram calls itself, to add yet another digit, and so on till the number is 9 digits long and all possibilities have been tested. Brief comments:

- Line 10 simply sets up some initial conditions; you may want to change the DELAY to slow down the display, but the STD is important to keep the presentation tidy.
- Line 20 is the "main program": it just establishes a loop to find solutions from 2 to 9 digits long, and calls the subprogram to do the actual work. That's it. You could search for numbers up to 12 digits long just by changing the 9 to 12 !
- Line 30 is the subprogram: it tests all 10 possibilities by adding a new digit to the already existing ones, and if the divisibility condition is met, it simply calls itself to add yet another digit
- Notice how the recursive call CALL TST(X,D+1,M) at line 40 depends on a condition, else the recursion would go ever deeper.

If you RUN this program, you'll get an output like this:

2 10 -> 3 102 -> 4 -> 1020 -> 5 -> 10200 -> 6 102000 -> 7 1020005 -> 8 10200056 -> 9 -> 102000564 -> 6 102006, etc

To find a 10-digit solution with no repeated digits, as the original challenge asked for, first we notice that for a 10-digit number to be divisible by 10, it must end in 0. So we just need to find 9-digits solutions, 0 excluded, then add a final 0 to all of them.

We just need a very few changes to the program above, like this 200+ byte version:

```
10 DESTROY ALL @ OPTION BASE 0 @ DELAY 0,0 @ STD
20 FOR I=1 TO 9 @ CALL TST((I),2,9) @ NEXT I @ DISP "DONE!" @ END
30 SUB TST(N,D,M) @ N=10*N @ FOR I=1 TO 9 @ X=N+I @ IF MOD(X,D) THEN 50
32 A$=STR$(X) @ FOR J=1 TO LEN(A$) @ IF POS(A$[J+1],A$[J,J]) THEN 50
34 NEXT J
40 DISP D;X @ IF D#M THEN CALL TST(X,D+1,M)
50 NEXT I @ END SUB
```

- As before, line 10 sets things up, but STD is even more mandatory, for the STR\$ conversion below.
- The loop at line 30 in the subprogram begins at 1 instead of 0, as 0 must be the final, 10th digit.
- Lines 32 and 34 simply check for repeated digits, by converting the candidate number (which already meets the divisibility requirement) to a string, then using the POS function to find if any character occurs more than once. It's a simple, non-optimal idea that works.

The rest of the program is just the same. When RUN, it outputs:

2 12 -> 3 123 -> ... -> 9 381654729

and so, the one and only 10-digit solution is 3816547290.

That's all, hope you enjoyed it, and see how recursion makes an easy job of apparently complex tasks.

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



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