♦MoHPC♠ *The Museum of HP Calculators*

HP Forum Archive 16

Valentine's Day Mini-Challenges !
*Message #1 Posted by **Valentin Albillo** on 14 Feb 2007, 6:45 a.m.*

Hi all,

It's been quite a long time since I posted my latest HP-15C Mini-Challenge (or S&SMC for that matter), but since ***today it's my name's day (St. Valentine)***, in order to commemorate the event and to mark time before the release of my new "S&SMC Spring's Special" (to be posted next April 1st), you may want to try these two short pieces, a truly *very easy* **HP-15C Mini-Challenge**, just to sharpen your HP calc programming teeth, then a somewhat more meaty **Name-Your-Model Mini-Challenge**, which you can attack with any newer or vintage HP model of your choice (other pocket-sized brands also welcome). Let's see:

The HP-15C Mini-Challenge

You must write an HP-15C program which, given as input a positive integer N between 0 and 5,000,000,000 (both included) in the display, it will compute and output *the number of trailing zeros* of N! (where the exclamation mark means the *factorial function*, not that this will be an awesome feat on your part :-)

For instance:

```
         0 [A] ->          0  (0! = 1 , has no trailing zeros)
         1 [A] ->          0  (1! = 1 , has no trailing zeros)
        13 [A] ->          2  (13! = 6,227,020,800, which ends in 2 zeros)
        69 [A] ->         15  (69! ends in 15 zeros)
      1000 [A] ->        249  (1000! ends in 249 zeros)
      2007 [A] ->        500  (2007! ends in exactly 500 zeros!)
   1234567 [A] ->     308638  (1000000! ends in 308,638 zeros)
4567890123 [A] -> 1141972520  (4567890123! ends in 1,141,972,520 zeros)
```

You must optimize primarily for size, then for speed, and must try to achieve a solution *in 10 steps or less*, including the initial LBL A but no RTN at the end, since the end of program memory acts as a default RTN instruction.

Within two or three days I'll post *my original 10-step solution*, which will compute the answer in a few seconds at most for any N, and no doubt many of you will either exactly duplicate or surpass it.

If you don't have a real, physical HP-15C, no problem. Just google for **Nonpareil** and download a freeware, exact Windows emulator which uses the original HP-15C ROMs for utmost emulation accuracy.

The Name-Your-Model Mini-Challenge

Once encouraged by your amazing success with the previous Mini-Challenge, take your HP calc model of choice to try and achieve the following:

You must write a program for your HP calc which, given as input a positive integer N between 0 and 5,000,000,000 (for 10-digit models) or between 0 and 500,000,000,000 (for 12-digit models), it will compute and output the rightmost digit of N! *disregarding all trailing zeros*.

For instance:

```
               0 ->  1  (0! = 1, last non-zero digit is 1)
               1 ->  1  (1! = 1, ditto)
              13 ->  8  (13! =  6,227,020,800, last nonzero digit is 8)
              69 ->  4  (69! = 1711..424000000000000000, last nonzero is 4)
            1000 ->  2  (1000! = 402..347200..000, last nonzero is 2)
            2007 ->  2  (2007! = 430..883200..000, last nonzero is also 2)
         1234567 ->  4  (1234567! last nonzero digit is a 4)
      4567890123 ->  8  (4567890123! last nonzero digit is a 8)
```

You must again optimize for speed and size. Within two or three days I'll post *my original 6-line solution for the HP-71B*, which will compute the answer for any N almost instantaneously (if you don't have a physical HP-71B but would like to try that particular HP model, google for **Emu71** for a truly excellent freeware emulator).

Well, that's all, let's now see what you can do with your own favorite HP model and your considerable programming skills !

Best regards from V.

Re: Valentine's Day Mini-Challenges !

*Message #2 Posted by **Les Wright** on 14 Feb 2007, 9:27 a.m.,*
in response to message #1 by Valentin Albillo

Sixteen steps, and seems a little slow, but I think not bad for a first effort:

```
LBL A
FIX 0
5
/
```

```
STO 1
INT
STO 2
LBL 0
5
STO/ 1
RCL 1
INT
STO+ 2
x!=0? (g TEST 0)
GTO 0
RCL 2
```

I got the series from the OEIS.

There must be something intrinsic to the 15C that you want us to optimize in this one....

Les

Re: Valentine's Day Mini-Challenges !

Message #3 Posted by ***Valentin Albillo*** *on 14 Feb 2007, 9:35 a.m.,*
in response to message #2 by Les Wright

Hi, Les:

Les posted:

> *"Sixteen steps, and seems a little slow, but I think not bad for a first effort:"*
>
> > Make that *"for a* ***second*** *effort"*, Les.
> >
> > Though you quickly and silently deleted your *first* 23-step effort, I got to see it (remember my change-notifying script I commented in a recent post ?). Didn't your mom tell you that good boys shouldn't tell lies ? You're now in a state of sin !! :-)
> >
> > Thanks for *both* your efforts and don't hesitate to go for a third, who's counting anyway ?

Best regards from V.

Re: Valentine's Day Mini-Challenges !

*Message #4 Posted by **Les Wright** on 14 Feb 2007, 9:51 a.m.,*
in response to message #3 by Valentin Albillo

Yes, you caught me.

I used the formula from MathWorld blindly, then found a much simpler series at the OEIS which saved me several steps and saved me from having to test if n < 5 first.

I could cut down this program to 15 steps if the user is permitted to FIX 0 before running.

I think you limitation of N to just 5 billion will save me a step or two--no need for a conditional test, just take the series out to k=13 terms, after which FLOOR(50000000/5^k) goes to zero.

There is an OEIS entry for the last nonsignificant digit of n!, but the formula is much more complex.

Les

Re: Valentine's Day Mini-Challenges !

*Message #5 Posted by **Gerson W. Barbosa** on 15 Feb 2007, 11:32 p.m.,*
in response to message #4 by Les Wright

> Quote:
>
> There is an OEIS entry for the last nonsignificant digit of n!, but the formula is much more complex.

Hello Les,

Do you mean the formula in this page?

It's really difficult to read, but once we manage to read it we see the formula is not so complex, though the principle behind it surely is. And not so difficult to implement on the HP-33S:

```
A0001 LBL A
A0002 STO A
A0003 0
A0004 STO i
A0005 1
A0006 STO P
B0001 LBL B
```

```
B0002 RCL A
B0003 5
B0004 RMDR
B0005 x!
B0006 LASTx
B0007 RCL* i
B0008 2
B0009 x<>y
B0010 y^x
B0011 *
B0012 STO* P
B0013 RCL A
B0014 5
B0015 INT/
B0016 x=0?
B0017 GTO C
B0018 STO A
B0019 1
B0020 STO+ i
B0021 GTO B
C0001 LBL C
C0002 RCL P
C0003 6
C0004 *
C0005 10
C0006 RMDR
C0007 RTN



LBL A:  LN= 42  CK=5E04
LBL B:  LN=111  CK=86B3
LBL C:  LN=45   CK=CEA4
```

This is just a test and still somewhat limited. It doesn't work for n=1 and when the resulting product in register P is equal or greater than 1E12. I think modular arithmetic can fix that.

Testing with some of Valentin's examples:

```
   13 -> 8
   69 -> 4
 1000 -> 2
 2007 -> 2
```

These have to be checked:

```
  7777 -> 4
100000 -> 6
```

Regards,

Gerson.

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-33S

*Message #6 Posted by **Gerson W. Barbosa** on 15 Feb 2007, 11:55 p.m.,*
in response to message #5 by Gerson W. Barbosa

> Quote:
>
> I think modular arithmetic can fix that.

```
A0001 LBL A
A0002 STO A
A0003 0
A0004 STO i
A0005 1
A0006 STO P
B0001 LBL B
B0002 RCL A
B0003 5
B0004 RMDR
B0005 x!
B0006 LASTx
B0007 RCL* i
B0008 2
B0009 x<>y
B0010 y^x
B0011 *
B0012 10
B0013 RMDR
B0014 STO* P
B0015 RCL A
B0016 5
B0017 INT/
B0018 x=0?
B0019 GTO C
B0020 STO A
B0021 1
```

```
B0022 STO+ i
B0023 GTO B
C0001 LBL C
C0002 RCL P
C0003 6
C0004 *
C0005 10
C0006 RMDR
C0007 RTN
```

> Quote:
>
> I think modular arithmetic can fix that.

I think lines B00012 and B00013 do the trick :-)

```
1234567 -> 4   (a couple of seconds)
```

It still doesn't work for n=1. A couple of lines would do, but it's close to 03:00 AM here...

Regards,

Gerson.

It doesn't work for 4567890123... This will have to wait... Sorry!

Edited: 15 Feb 2007, 11:58 p.m.

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-33S

Message #7 Posted by ***Egan Ford*** *on 16 Feb 2007, 12:10 a.m.,*
in response to message #6 by Gerson W. Barbosa

> Quote:
>
> It doesn't work for 4567890123... This will have to wait... Sorry!

Do a mod 4 on i after the i++, see my 15C solution it works for numbers up to 9,999,999,999.

UPDATE: Opps, not true. My perl prototype was limited to 10 digit numbers for all variables, I assumed the port to the 15C would also function. Perl unlike the 15C will return a FRAC for n > 5,000,000,000 / m. When attempting to mod 5 numbers > 5,000,000,000 the 15C always returns 0.

Edited: 16 Feb 2007, 4:40 p.m. after one or more responses were posted

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-33S

Message #8 Posted by ***Gerson W. Barbosa*** *on 16 Feb 2007, 12:13 a.m.,*
in response to message #7 by Egan Ford

Thanks,

I'll try tomorrow. Too sleepy to think...

Gerson.

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-33S - Fixed!

Message #9 Posted by ***Gerson W. Barbosa*** *on 16 Feb 2007, 7:45 a.m.,*
in response to message #7 by Egan Ford

> Quote:
>
> Do a mod 4 on i after the i++, see my 15C solution it works for numbers up to 9,999,999,999.

Thanks!

```
A0001 LBL A
A0002 STO A
A0003 1
A0004 STO P
A0005 x>=y?
A0006 RTN
A0007 0
A0008 STO i
B0001 LBL B
B0002 RCL A
B0003 5
B0004 RMDR
B0005 x!
```

```
B0006 LASTx
B0007 RCL* i
B0008 2
B0009 x<>y
B0010 y^x
B0011 *
B0012 10
B0013 RMDR
B0014 STO* P
B0015 RCL A
B0016 5
B0017 INT/
B0018 x=0?
B0019 GTO C
B0020 STO A
B0021 1
B0022 STO+ i
B0023 RCL i
B0024 4
B0025 RMDR
B0026 STO i
B0027 GTO B
C0001 LBL C
C0002 RCL P
C0003 6
C0004 *
C0005 10
C0006 RMDR
C0007 RTN


LBL    CK      LN
 A    B68D     48
 B    49D1    153
 C    CEA4     45
             ----
Total length: 246 bytes
```

Now it works for all Valentin's examples, including 0 and 1. The program is not optimized, of course.

500,000,000,000 -> 8 (less than 3 seconds, this means about 100 seconds on the 15C)

Regards,

Gerson.

Update:

Replaced

```
A0001 LBL A
A0002 x=0?
A0003 x!
A0004 1
A0005 x<>y
A0006 x=y?
A0007 RTN
A0008 STO A
A0009 0
A0010 STO i
A0011 1
A0012 STO P
```

with

```
A0001 LBL A
A0002 STO A
A0003 1
A0004 STO P
A0005 x>=y?
A0006 RTN
A0007 0
A0008 STO i
```

Better, isn't it?

One more note: the 33S version works for N up to 999,999,999,999.

Edited: 18 Feb 2007, 8:33 a.m. after one or more responses were posted

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-33S - Fixed!

Message #10 Posted by ***Les Wright*** *on 16 Feb 2007, 8:09 a.m.,*
in response to message #9 by Gerson W. Barbosa

Quote:

The program is not optimized, of course.

A couple of step savers:

Get rid of A0009 and A0010 and put CLRVRS after A0007. It works to set the index register i to zero too (I checked).

Replace 1 STO P at A0011-12 by ISG P (since we have cleared P a few steps above). The next line, LBL B, is conveniently there to act as the NOP instruction to skip over.

That creates a modest saving of two steps. But it is a start!

Les

Edited: 17 Feb 2007, 12:33 a.m. after one or more responses were posted

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-33S - Fixed!

*Message #11 Posted by **Gerson W. Barbosa** on 16 Feb 2007, 8:31 a.m.,*
in response to message #10 by Les Wright

Hello Les,

Thanks for these and other suggestions you have made. I'd rather avoid CLRVARS, though. I won't have time to work on this any further, but I encourage anyone who's willing to optimize it. Also, I'd like to know what early calculors this and Egan's program could be ported to. They don't fit in the 33C, but they do fit in the 34C and 11C, for instance.

Actually, I liked the first part better, because I was able to come up with a formula on my own, albeit the resulting program was twice as large :-)

Thanks to you now I know what OEIS is and what it is useful for. I had to google for OEIS to get to the site. I was aware of some Sloane's sequences but I had never noticed they were all organized at OEIS.

Best regards,

Gerson.

Edited: 16 Feb 2007, 8:33 a.m.

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-33S - Fixed!

Message #12 Posted by ***Gerson W. Barbosa*** *on 16 Feb 2007, 8:48 a.m.,*
in response to message #11 by Gerson W. Barbosa

> Quote:
>
> Thanks to you now I know what OEIS is and what it is useful for.

What is the next number in the sequence?

2, 10, 12, 16, 17, 18, 19, ?

No, don't take the time to find it. Only Portuguese and Spanish-speaking people would be able to solve this :-)

Just for fun I searched at OEIS and here is is:

http://www.research.att.com/~njas/sequences/?q=2%2C10%2C12%2C16%2C17%2C18%2C19&sort=0&fmt=0&language=english&go=Search

I thought these kind of sequence would not be there...

Regards,

Gerson.

Edited: 19 Feb 2007, 8:01 a.m.

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-32SII

Message #13 Posted by ***Gerson W. Barbosa*** *on 16 Feb 2007, 11:56 a.m.,*
in response to message #10 by Les Wright

Hello again, Les!

I've just received a near-mint brown-bezel HP-32SII ($29.32 USD, shipping included!). It passes both self-tests and this one too :-)

Your suggestions accepted! Thanks!

```
A01 LBL A
A02 CLVARS
A03 STO A
A04 1
A05 STO P
A06 x>=y?
A07 RTN
B01 LBL B
B02 RCL A
B03 5
B04 XEQ D
B05 x!
B06 2
B07 LASTx
B08 RCL* I
B09 y^x
B10 *
B11 10
B12 XEQ D
B13 STO* P
B14 RCL A
B15 5
B16 /
B17 IP
B18 x=0?
B19 GTO C
B20 STO A
B21 1
B22 STO+ i
B23 RCL i
B24 4
B25 XEQ D
B26 STO i
B27 GTO B
C01 LBL C
C02 RCL P
C03 6
C04 *
C05 10
D01 LBL D
D02 /
D03 LASTx
D04 x<>y
D05 FP
D06 *
D07 RTN
```

```
LBL     CK       LN
 A     8CE2    010.5
 B     5AC4    040.5
 C     4E8D    007.5
 D     1CBB    010.5
               -----
Total length:  069.0 bytes


500,000,000,000 -> 8  (6.5 seconds)
```

Best regards,

Gerson.

Update (16 Feb) :

```
C06 XEQ D
C07 RTN
```

These lines were absolutely not necessary!

Update (18 Feb) :

Replaced

```
A01 LBL A
A02 x=0?
A03 x!
A04 1
A05 x<>y
A06 x=y?
A07 RTN
A08 CLVARS
A09 STO A
A10 ISG P
```

with

```
A01 LBL A
A02 CLVARS
A03 STO A
```

```
A04 1
A05 STO P
A06 x>=y?
A07 RTN
```

This saves 4.5 bytes. Your CLVARS suggestion kept. Your ISG suggestion was tricky for the older version though.

The working range is from 0 to 500,000,000,000.

Edited: 18 Feb 2007, 8:34 a.m.

Re: Valentine's Day Mini-Challenges !

Message #14 Posted by ***Les Wright*** *on 14 Feb 2007, 12:05 p.m.,*
in response to message #3 by Valentin Albillo

Valentin, you should be honoured.

I broke out my rarely used, mint condition 15C just for this occasion.

I paid a small fortune for the calculator, and it is beautiful, and I am afraid to use it, prefering the much more battle-scarred 11C.

Yes, you should be honoured....

Les

Re: Valentine's Day Mini-Challenges !

Message #15 Posted by ***Valentin Albillo*** *on 14 Feb 2007, 2:00 p.m.,*
in response to message #14 by Les Wright

Hi, Les:

> I am, I am ... though I have the feeling that when my solution gets posted you're gonna hate me a little for it ... :-|

Best regards from V.

Re: Valentine's Day Mini-Challenges !

Message #16 Posted by ***Les Wright*** *on 14 Feb 2007, 10:16 a.m.,* *in response to message #2 by Les Wright*

Shaved it down to 12 steps by getting rid of the initial redundant division by 5 and using CLEAR REG to make sure there is a 0 to start in the summation register. Also, that initial FIX 0 is irrelevant--the INT command acts like RND in this context:

```
LBL A
CLREG (f CLEAR REG)
STO 1
LBL 0
5
STO/ 1
RCL 1
INT
STO+ 2
x!=0? (g TEST 0)
GTO 0
RCL 2
```

I think 12 steps is getting there, eh?

Les

Edited: 14 Feb 2007, 10:16 a.m.

Re: Valentine's Day Mini-Challenges !

Message #17 Posted by ***Gerson W. Barbosa*** *on 14 Feb 2007, 10:27 a.m.,*
in response to message #2 by Les Wright

Hello Les:

A seventeen-step solution (you win!):

```
001 LBL A
002 CLEAR REG
003 STO 2
004 1
005 STO 1
006 LBL 0
007 RCL 2
008 RCL 1
009 5
010 *
011 STO 1
```

```
012 /
013 INT
014 STO+ 0
015 TEST 1
016 GTO 0
017 RCL 0
```

5,000,000,000 returns 1,249,999,998 in 14 seconds (time on Nonpareil).

Of course this still can be optimized, but lunch-break ends in one minute, and I am not supposed to be solving this kind of problem during working hours. That's not what I am paid by my employer for :-)

Regards,

Gerson.

Edited: 14 Feb 2007, 10:30 a.m.

Re: Valentine's Day Mini-Challenges !

*Message #18 Posted by **Valentin Albillo** on 14 Feb 2007, 11:16 a.m.,*
in response to message #17 by Gerson W. Barbosa

Hi, Gerson:

Gerson posted:

> *"I am not supposed to be solving this kind of problem during working hours. That's not what I am paid by my employer for :-)"*

> My, my, aren't you being a little too *"self-righteous"* ? Be careful or you might end up taking from me the *"World Prize for Hollow Self-Righteousness"* that some anonymous coward nominated me for, for (tongue-in-cheek) saying that much a few weeks ago. Though I doubt she will comply if you win ... :-)

> Thanks for your contribution, do your utmost to improve it to 10 steps or less (actually, it's easier than it seems, trust me!) and

Best regards from V.

Re: Valentine's Day Mini-Challenges !

*Message #19 Posted by **Gerson W. Barbosa** on 14 Feb 2007, 3:25 p.m.,* *in response to message #18 by Valentin Albillo*

Hello, Valentin!

> Quote:
>
> ...that some anonymous coward nominated me for, for (tongue-in-cheek) saying that much a few weeks ago.

I remember I read your original remark some weeks ago :-)

I was able to find a 10-step solution, but I relied on J-F Garnier's approach.

My first approach was summing up the integer part of N divided by successive powers of 5. For instance, for N=2007:

```
INT(2007/5)    = 401
INT(2007/25)   =  80
INT(2007/125)  =  16
INT(2007/625)  =   3
INT(2007/3125) =   0
                 ---
                 500
```

Though this works, I think a 10-step solution based on that would not be possible. Would it?

Best regards,

Gerson.

Re: Valentine's Day Mini-Challenges !

Message #20 Posted by ***Les Wright*** *on 14 Feb 2007, 8:26 p.m.,*
in response to message #19 by Gerson W. Barbosa

> Quote:
>
> Though this works, I think a 10-step solution based on that would not be possible. Would it?

I think you have discovered that it is.

The key is that INT(n/5^(k+1)) very conveniently happens to equal INT(INT(n/5^k)/5). (I am sure there is a way to prove this by induction, but I am too lazy at present.)

This means you generate the next term of the series just by dividing the previous by 5 and taking INT. Saving the "pre-INT" version of the former term and dividing that by 5 before taking INT is happily unnecessary and costs those extra steps.

I think Gerson and JF should get the gold medal and I get the silver. :)

Les

Re: Valentine's Day Mini-Challenges !

Message #21 Posted by ***Les Wright*** *on 14 Feb 2007, 8:28 p.m.,*
in response to message #20 by Les Wright

PS. I should put in a plug for the much taunted 33S.

These little routines are VERY FAST on that oft scourged machine. They are even fast if you splurge on 12 or 16 steps or so.

Long live the 33S! Everyone flash your Star Trek chevron salute!!!!

Les

Re: Valentine's Day Mini-Challenges !

Message #22 Posted by ***Gerson W. Barbosa*** *on 14 Feb 2007, 9:27 p.m.,*
in response to message #21 by Les Wright

Hello Les,

On the 33S, we can have a nine-step solution:

```
LBL A
CLVARS
LBL B
5
INT/
STO+ i
x#0?
```

```
GTO B
RCL i
```

Regards,

Gerson.

Edited: 14 Feb 2007, 9:27 p.m.

Re: Valentine's Day Mini-Challenges !

Message #23 Posted by ***Les Wright*** *on 14 Feb 2007, 10:47 p.m.,*
in response to message #22 by Gerson W. Barbosa

With its lack of labels, the 33C allows 8 steps:

```
01 CLEAR REG
02 5
03 /
04 INT
05 STO+ 0
06 x!=0?
07 GTO 02
08 RCL 0
```

And, head to head with the 11C, it is even a little faster...

Les

p.s. I misentered your code in my 11C and at times it wouldn't converge. I thought the calculator was busted, as it has been "hanging" occasionally--some internal contact stuff needs a permanent fix. Eventually I figured out I substituted x#y? for x#0? So it would converge sometimes if the leftovers on the stack were favourable, would go on forever if not. I was tearing my hair out!

Edited: 14 Feb 2007, 10:54 p.m.

Re: Valentine's Day Mini-Challenges !

Message #24 Posted by ***Kiyoshi Akima*** *on 17 Feb 2007, 1:08 a.m.,*
in response to message #2 by Les Wright

Here's a quick and dirty RPL implementation, tested on my HP48GX:

```
<<
 IF DUP 2 < THEN
  DROP 1
 ELSE
  { }
  "0313211214224233313244341014421233424113314424322102122142442343332122413110441213324311433441242231"
  -> f s
  <<
   WHILE DUP REPEAT
    DUP 5 MOD 'f' STO+ 5 / IP
   END
   f SIZE 3 + 4 MOD
   WHILE f SIZE REPEAT
    SWAP 5 * f HEAD + OVER 25 * + 1 + s SWAP DUP SUB NUM 48 -
    f TAIL 'f' STO SWAP 3 + 4 MOD
   END
   DROP DUP +
  >>
 END
>>
```

This program works for positive integers in the range 0..999999999992 inclusive, taking less than 1.7 seconds worst case (less than 0.1 seconds best case) (much faster on Emu48). I know why it doesn't work for 999999999993..999999999999; the first division by 5 rounds up to the next integer. (Actually, it also works for 999999999995..999999999997 because the division works correctly in these cases.) Since the challenge specified 500000000000 for the upper limit for the 12-digit models, I won't bother to fix it. I also see several ways to speed it up, but I'm satisfied with its current performance.

I also have a five-liner for the HP200LX, but I consider it cheating and hence won't post it here.

Re: Valentine's Day Mini-Challenges !

Message #25 Posted by ***Les Wright*** *on 17 Feb 2007, 6:40 a.m.,*
in response to message #24 by Kiyoshi Akima

Even though I have a serial cable for my 48G, it was faster to cut and paste your code to a text file, drop it onto a 49G implemetation of Emu48, and after tweaking the symbols saving it as a 49 series binary and moving to my 49G+ where it seems to be about as fast as you say at least, though I didn't put a stopwatch on it. It is quite fast and clearly meets the specifications of the challenge. Bravo!

I am intrigued by this since this is clearly a different approach than the Dresden formula on that OEIS page that the rest of us have been looking at. What is the significance of the big long string of digits?

Les

Re: Valentine's Day Mini-Challenges !

Message #26 Posted by ***Egan Ford*** *on 17 Feb 2007, 10:40 a.m.,*
in response to message #25 by Les Wright

> Quote:
>
> What is the significance of the big long string of digits?

It's a look up table.

http://www.mathpages.com/home/kmath489.htm

Re: Valentine's Day Mini-Challenges !

Message #27 Posted by ***Kiyoshi Akima*** *on 17 Feb 2007, 1:14 p.m.,*
in response to message #25 by Les Wright

> Quote:
>
> ... though I didn't put a stopwatch on it.

Neither did I. I simply wrote another program like this:

```
<<
 TICKS SWAP LF TICKS ROT - B->R 8192 /
>>
```

and let the calculator time itself. (LF is the program to compute the last non-zero digit of a factorial.) Much easier than running a stopwatch with one hand and a calculator with the other.

> Quote:
>
> What is the significance of the big long string of digits?

Would you believe it's a machine-language program? No? As Egan pointed out, it's a lookup table, taken from the MathPages reference he cited. The first loop converts the input to base-5, putting the digits into a list. The second loop goes through the digits one at a time, looking up the appropriate entry in the table. This makes the running time proportional to log(n).

I did make one change to the algorithm, dividing all the entries in the table by two rather than having the program do it each time through the loop, hence accounting for the extra DUP + at the end. Otherwise it's pretty much straight from the MathPages reference.

The HP200LX version is now down to one line of obfuscated C, which I will NOT post here.

Edited: 17 Feb 2007, 1:16 p.m.

Re: Valentine's Day Mini-Challenges !

Message #28 Posted by ***Kiyoshi Akima*** *on 18 Feb 2007, 2:22 a.m.,*
in response to message #24 by Kiyoshi Akima

My earlier analysis was totally bogus. The RPL program gets the first division wrong 40% of the time above 500,000,000,000, though it gets the right answer by chance about 25% of those times. Thus, the upper limit of the domain is 500,000,000,002.

To make up for the gaffe, I present my solution to an exercise I left for the reader in my original post: extending a binary-mode 16C program to handle decimal. It doesn't handle the 0 and 1 cases, though they can be put in rather easily. Its upper limit is (5^20)-1 (about 9.5E13), which it does in just under two minutes.

```
Word size 48, DEC, UNSIGNED


pre-store:
R21: 265161901066072
R22: 150653886932972
R23:  67185572116372
R24:  76106210809632


01 LBL A       25 x#0
02 1           26 GTO 3
03 STO I       27 RCL I
04 R dn        28 3
05 LBL 1       29 AND
06 ENTER       30 2
07 ENTER       31 1
08 5           32 +
09 RMD         33 x<>I
10 STO (i)     34 RCL (i)
11 R dn        35 x<>y
12 5           36 STO I
13 /           37 R dn
14 ISZ         38 x<>y
15 LBL 0       39 SL
```

```
16 x#0         40 RLn
17 GTO 1       41 3
18 DSZ         42 AND
19 CLx         43 1
20 LBL 2       44 +
21 5           45 LBL 3
22 *           46 DSZ
23 RCL (i)     47 GTO 2
24 +           48 SL
```

If you make the change to handle 0 and 1, you'll have to shift the constants down a register or two, which will cost digits and reduce the upper limit.

I suppose that with 48-bit registers, one could pack more than one base-5 digit into each and thus really extend the upper limit. This will be left as an exercise for the reader...

Re: Valentine's Day Mini-Challenges !

Message #29 Posted by ***Gerson W. Barbosa*** *on 14 Feb 2007, 9:50 a.m.,*
in response to message #1 by Valentin Albillo

Hello Valentin,

Here is my non-optimized solution:

```
001 LBL A
002 STO 2
003 1
004 STO 1
005 0
006 STO 0
007 LBL 0
008 RCL 2
009 RCL 1
010 5
011 *
012 STO 1
013 /
014 INT
015 STO+ 0
016 TEST 1
017 GTO 0
018 RCL 0
```

Eighteen steps! Well, 0 STO 0 can be replaced by CLR REG, thus saving one step. Even so, too long a program...

Best regards,

Gerson

P.S.: I've come to the formula by myself, analyzing the first 27 factorials and a little guessing about the rest. I am not sure if this is the most efficient one...

People without a 15C at hand, like me, might find this table useful:

```
TEST


0  x<>0
1  x>0
2  x<0
3  x>=0
4  x<=0
5  x=0
6  x<>y
7  x>y
8  x<y
9  x>=y
```

Edited: 14 Feb 2007, 9:55 a.m.

Re: Valentine's Day Mini-Challenges !

Message #30 Posted by ***J-F Garnier*** *on 14 Feb 2007, 1:31 p.m.,*
in response to message #1 by Valentin Albillo

Hi,

My 12-step solution for a HP-11C, using no memory register:

```
LBL A
0
x<>y
LBL 1
5
/
```

```
INT
+
LASTx
x#0?
GTO 1
RDN
```

Quite fast... but 12 steps ...

J-F

Re: Valentine's Day Mini-Challenges !

Message #31 Posted by ***Les Wright*** *on 14 Feb 2007, 2:19 p.m.,*
in response to message #30 by J-F Garnier

I like yours much better than mine.

I think a 12 step solution is just fine for this little challenge.

Next!

Re: Valentine's Day Mini-Challenges !

Message #32 Posted by ***Les Wright*** *on 14 Feb 2007, 2:22 p.m.,*
in response to message #30 by J-F Garnier

I think the hint is that Valentin posed the first problem SPECIFICALLY for HP15C. So the 10 step answer must rely on some unique feature of that calculator. But what????

So far, your solution, mine, and Gerson's could be programmed on most HP RPN programmables....

Les

Re: Valentine's Day Mini-Challenges !

Message #33 Posted by ***Gerson W. Barbosa*** *on 14 Feb 2007, 3:13 p.m.,*
in response to message #30 by J-F Garnier

Congratulations!

You're approach is better than mine. Using it, the following 10-step solution is possible:

```
001 LBL A
002 CLEAR REG
003 LBL 0
004 5
005 /
006 INT
007 STO+ I
008 TEST 0   ;  x#0?
009 GTO 0
010 RCL I
```

No (numbered) register!

Regards,

Gerson.

Re: Valentine's Day Mini-Challenges !

Message #34 Posted by ***Les Wright*** *on 14 Feb 2007, 5:50 p.m.,*
in response to message #33 by Gerson W. Barbosa

Gerson, I think this is it.

I was tripped up by one thing--I couldn't get my brain around the idea that floor(x)/5 is the same as floor(x/5), but of course it should be, even though I am hard pressed to prove it right now. Working with x/5 and floor(x/5) as possibly different cost those two extra steps in my routine.

Excellent work! I think we can retire this one.

Les

Re: Valentine's Day Mini-Challenges !

Message #35 Posted by ***Les Wright*** *on 14 Feb 2007, 6:07 p.m.,*
in response to message #34 by Les Wright

Quote:

> the idea that floor(x)/5 is the same as floor(x/5)

Actually, this isn't true, but somehow doing the division where you do it allows for a correct answer and keeps the code down to 10 steps....

Makes sense, I just wish I better understood why....

Les

Re: Valentine's Day Mini-Challenges !

*Message #36 Posted by **Gerson W. Barbosa** on 14 Feb 2007, 6:07 p.m.,*
in response to message #34 by Les Wright

Hello Les,

And I was tripped by my initial approach, as you can see in my reply to Valentin. But that was the pattern I was able to find then...

> Quote:
>
> Excellent work! I think we can retire this one.

Thanks, but I think the compliment should go to J-F Garnier, whose approach I followed :-) I believe your approach is not different, but I didn't have time to take a look at your solution this afternoon.

Well, I wish you good luck to you on the second part. One of those Valentin's mini-challenges is too much for the day :-)

Best regards,

Gerson.

Re: Valentine's Day Mini-Challenges !

*Message #37 Posted by **PeterP** on 14 Feb 2007, 8:59 p.m.,*
in response to message #1 by Valentin Albillo

Got home to find this new challenge - and realised that given the large number of postings, probably all is already solved. Anyway, did not read anything and started on the simpler, first on. Don't have a 15c (yet), but I believe the program below should work on most RPN's (it is conceived on a 41). The solution below assumes an empty reg 00 (and clears it for the following runs) and hence is only 9 steps. If one has to clear 00 first, this gives a 12 step solution. Given that Valentin made the challenge for the 15C I presume that the 15C has a function that gives the value of Int(Y/X) which would reduce the 5,/,Int step to one. Angel Martin's wonderful Sandbox has a function QRem which could replace those three steps with QREM,RDN. Anyway, now I will have dinner (Indian food) and then maybe try the harder one as well. Thanks for this nice pre-dinner entertainment Valentin, as always, it is much appreciated!

```
Lbl A
5
/
Int
Sto+ 00
X>0?
GTO A
Clx
X<>00
```

Re: Valentine's Day Mini-Challenges !

Message #38 Posted by ***Les Wright*** *on 14 Feb 2007, 10:52 p.m.,*
in response to message #37 by PeterP

> Quote:
>
> The solution below assumes an empty reg 00

Valentin, is this allowed? If so, we have a new winner!

Les

Re: Valentine's Day Mini-Challenges !

Message #39 Posted by ***PeterP*** *on 15 Feb 2007, 1:09 a.m.,*
in response to message #37 by PeterP

A dinner, a movie and some thinking time later I must admitt that the second challenge is a bit trickier. I can get it to work up to a certain factorial number using similar principles as for the first one (finding the total number of prime divisors, finding the possible end digit each one can produce (only four possible per prime divisor) and multiplying those out. But I am overlooking something somewhere...

Anyway, maybe I can figure something out by tomorrow, unless Valentin posts his solution already. (I can resist reading the other posts while I try to figure things out, but once Valentin posts his solution, I know the problem has been solved multiple times by the Marcus', Les', Karl's, J-F's and Gerson's of the world, who actually know what they are doing, and I just want to see the masters at work ;-))

Cheers

Peter

Part I solved ! Any takers for the 2nd half ? :-)

Message #40 Posted by ***Valentin Albillo*** *on 15 Feb 2007, 7:45 a.m.,*
in response to message #1 by Valentin Albillo

Deadline extended to next Monday, so you'll have time aplenty to try and conquer it, weekend included.

Anyway, it's not that hard at all (else it would be a full-fledged ***S&SMC(tm)***, not a Mini-Challenge !

Best regards from V.

Re: Part I solved ! Any takers for the 2nd half ? :-)

Message #41 Posted by ***Les Wright*** *on 15 Feb 2007, 8:26 a.m.,*
in response to message #40 by Valentin Albillo

What is your official 10 step solution for Part 1, Valentin?

Re: Part I solved ! Any takers for the 2nd half ? :-)

Message #42 Posted by ***Valentin Albillo*** *on 19 Feb 2007, 6:56 a.m.,*
in response to message #41 by Les Wright

Hi, Les:

Les posted:

> *"What is your official 10 step solution for Part 1, Valentin?"*

> See below for both original solutions and comments.

> Thanks for your interest and continued efforts to optimally solve this mini-challenge and

Best regards from V.

Re: Part I solved ! Any takers for the 2nd half ? :-)

Message #43 Posted by ***PeterP*** *on 15 Feb 2007, 9:42 p.m.,*
in response to message #40 by Valentin Albillo

grrr, who needs 5 anyway! very annoying...

Re: Part I solved ! Any takers for the 2nd half ? :-)

Message #44 Posted by ***PeterP*** *on 16 Feb 2007, 12:50 a.m.,*
in response to message #43 by PeterP

OKay, I give up. I'm not smart enough for a smart way, so here is a rather brute force way for my trusty old 41, with the usual help of Mr. Angel Martin.

The concept is rather straight forward: It calculates the prime-factor construction (? don't know the right english term) of n!, using the nifty little routine we developped in part 1 (where we used it for 5 only, now we use it for all primes <= n).

It then makes use of four obvious facts: 1) it removes all the 2*5 pairs, as they just give 0; 2) the last digit of p^k is identikal to the last digit of p^(k mod 4); 3)all primes end either 1,3,7 or 9 and have the same effect as 3,7,9 and hence we sum them all up in the same reg 0p and use the nifty mod 4 factor to keep all numbers small and neat and lastly 4) we do not need to calculate the powers of all primes which end with a 1, they do not affect the last digit of n!. This check (done at the start of Lbl 02) does not help with the small n, but becomes handy at the larger n, where we do get powers of 11,31, etc...

Anyway, the below program is long and takes it sweet little time on the 41 for large n (about 4min for 2007!, it scales with about sqrt(n) or less for large n. Far from the 'almost instantaneous of Valentin...) but it has the advantage that a) I was able to figure it myself and b)it works. Both of which are no small feats for little ole' me...

Anyway, thanks for Valentin for great fun and I write some comments after the listing about where I got stuck with counting all the factors of 5 for large n while trying to do this a smart way...

```
Lbl LSSMC (for Little Short and Sweet Mathc Challenge)
CLRG   (just some prep work, storing of numbers etc)
CLA
Sto 11 (this is n)
Sto 12 (this will hold the first prime to test, either n or n-1)
2
Sto 00 (store it for speed purposes, digits are reaaaal slow for 41)
Mod
X=1?
  GTO 00
1
St- 12
```

```
Lbl 00
 10
 Sto 04 (store for speed purposes)
 Rcl 12


Lbl 01 (go down from n and find all primes)
  Prime? (God, I love you Angel...)
    GTO 02
  Last X
  Rcl 00 (reg 00 is the number 2)
  -
GTO 01


Lbl 02
  Sto 10
  Rcl 04 (check if the prime ends in 1, if so, ignore)
  Mod
  X=1?
   GTO 05


CLA     (this clears Reg M fastly...)
Rcl 11
Lbl 03 (this routine should be familiar...)
  Rcl 10 (reg 10 is the current prime factor)
  /
  Int
  Sto+ M
  X>0?
GTO 03
Rcl 10
Rcl 04   (reg 04 is the number 10)
Mod
Sto+ Ind Y


Lbl 05 (we land here if we had found a prime ending in 1, which we
        can ignore)
  Rcl 10
  Rcl 00 (reg 00 is the number 2)
  -
  X=1?
    GTO E (pfhh, we are done...)
GTO 01
```

```
Lbl E
Tone 9 (almost done)
CLA
Rcl 11
Lbl 04 (this gets the power of 2)
  Rcl 00
  /
  Int
  Sto+ M
  X>0?
GTO 04


Rcl M
Rcl 05 (subtract the powers of 5)
-
4
Mod
X=0?
  4
Rcl 00
X<>Y
y^x
Sto 01 (01 keeps the relevant digits)


Rcl 03  (this, Lbl 07 and Lbl 09 collect the powers of all primes
        ending in 3,7 and 9)
x=0?
  GTO 07
4
Mod
x=0?
  4
3
x<>y
y^x
Sto* 01


Lbl 07
Rcl 07
X=0?
 GTO 09
4
Mod
X=0?
```

```
  4
7
x<>y
y^x
Sto* 01


Lbl 09
Rcl 09
x=0?
  GTO 10
4
Mod
x=0?
  4
9
x<>y
y^x
Sto*01


Lbl 10 (Horray!!)
Rcl 01
Rcl 04
Mod
Beep
Stop
```

Here is where i got stuck trying to find a smarter way: basically onbly the last digit of all 1,2,3,...n matter for the last digit of n! (lets call it ld(n!)). so one could simply count how often one gets the sequence 1*2*3*4*5*6*7*8*9, lets call that X. Knowing that 9! gives a last digit of 8 (after removing the 2*5 for the zero), the ld(n!) = ld(8^(X mod 4)). Well, almost, but not quite. The nasty thing is that the numbers ending with 5 in n (e.g. 15,25,35 for n=36) leave some remnants after the 5 is paired off with a two. So the 15 leaves a 3 (from 3*5), the 25 leaves a 5 (from 5*5) and 35 leaves a 7 (from 5*7). One has to multiply the 8^(Xmod4) with all those remnants to get the right number. And then there is additionally the factors from the full 10's (2 from the 20 and 3 from the 30). Things get really ugly when n>100, as you get another nasty 5 from the 50, that you could pair off with the 2 from the 20 to ignore, and be left with the 3*4*6*7*8*9 which gives a last digit of 8 again. So you have to count those again. Somewhere just about now my brain gave up on to keep track of all the 5's floating around that are not paired off with a 2. Yet I'm sure there must be ab even smarter way. Angel provides a nifty function that can move any number into any base. Maybe moving n into base 5 can help to avoid the above mentioned problems of keeping track of them, but it is now to late for me to think about this.

Re: Part I solved ! Any takers for the 2nd half ? :-)

*Message #45 Posted by **Egan Ford** on 16 Feb 2007, 1:41 a.m.,*
in response to message #44 by PeterP

http://home.wlu.edu/~dresdeng/papers/lnzd2.pdf, jump to page 10.

Edited: 16 Feb 2007, 3:02 a.m.

Re: Part I solved ! Any takers for the 2nd half ? :-)

Message #46 Posted by ***PeterP*** *on 16 Feb 2007, 4:50 p.m.,*
in response to message #45 by Egan Ford

Thanks! I see... Looking through the code it seems that most are based on this formula or some derivative. Is that correct? Well, if it is worth a published paper I don't feel to bad for having given up... :-)

Cheers

Peter

Re: Part I solved ! Any takers for the 2nd half ? :-)

Message #47 Posted by ***Egan Ford*** *on 16 Feb 2007, 5:16 p.m.,*
in response to message #46 by PeterP

> Quote:
>
> Thanks! I see... Looking through the code it seems that most are based on this formula or some derivative. Is that correct?

Yes. This is also interesting: http://www.mathpages.com/home/kmath489.htm

Re: Part I solved ! Any takers for the 2nd half ? :-)

Message #48 Posted by ***Gerson W. Barbosa*** *on 16 Feb 2007, 9:32 p.m.,*
in response to message #46 by PeterP

> Quote:
>
> Looking through the code it seems that most are based on this formula or some derivative.

Hello Peter,

I used the formula in this link:

http://www.research.att.com/~njas/sequences/A008904

Quoting from the site:

Quote:

```
FORMULA  The generating function for n>1 is as follows: for n = a_0 + 5 a_1 + 5^2 a_2 + ... +5^N a_N
         (the expansion of n in base-5), then the last nonzero digit of n!, for n>1, is
         6*\prod_{i=0}^N (a_i)! (2^(i a_i)) mod 10 - Greg Dresden (dresdeng(AT)wlu.edu), Feb 21 2006
```

Hand-computing for n = 2007:

First, n is converted to base 5:

2007 -> $(31012)_5$

Then, the last non-zero digit is computed as:

$[6 * 3! * 2^{4*3} * 1! * 2^{3*1} * 0! * 2^{2*0} * 1! * 2^{1*1} * 2! * 2^{0*2}] \bmod 10 =$

$[6 * 6 * 2^{12} * 2^3 * 2 * 2] \bmod 10 = 2$

Regards,

Gerson.

Re: Part I solved ! Any takers for the 2nd half ? :-)

Message #49 Posted by ***PeterP*** *on 16 Feb 2007, 9:52 p.m.,*
in response to message #48 by Gerson W. Barbosa

Thanks guys, super interesting! Encouraged by your pointers, I started to poke around in google a bit more and came accross the following book, which might be a fun read for many of us and a potential source of ideas for Valentin to keep us from doing productive work. Oops, did I say this out loud...

It is called 'Concrete Mathematics' and is a short 670 page book written by Graham, Knuth and Patashnik. It also has some 500 'examples for the interested reader' and we all know what that means... It was mentioned in a post with regards to our 'last non-zero digit of n!' problem (seems like 'Exercise 4.40 in that book also comes up with a formula...') and I found a link to the pdf here

Thanks a lot for giving me those pointers, as usually I learned a lot from you guys (And sure will from Valentin's final answer as well)

BTW - Valentin, I got myself a 71 and might start poking around in that machine as well, it seems to have a special place in your heart. Though for sure it will be a time before I feel even remotely comfortable enough to try some of your challenges on it!

Cheers and have a great weekend!

Peter

Re: Part I solved ! Any takers for the 2nd half ? :-)

Message #50 Posted by ***Valentin Albillo*** *on 19 Feb 2007, 6:53 a.m.,*
in response to message #49 by PeterP

Hi, PeterP:

PeterP wrote:

> *"I started to poke around in google a bit more and came accross the following book [...] It is called 'Concrete Mathematics' and is a short 670 page book written by Graham, Knuth and Patashnik. "*

>> Thank you very much for the link ! :-) I already had it in printed form, but I find it much more convenient to have technical books in PDF format for easy and quick reference. When extensively reading them for leisure, printed form is preferable and easier on the eyes, of course.

> *"BTW - Valentin, I got myself a 71 and might start poking around in that machine as well, it seems to have a special place in your heart. Though for sure it will be a time before I feel even remotely comfortable enough to try some of your challenges on it!"*

>> Yes, it certainly has a place in my heart, together with the HP-25, HP-67, HP-34, HP-41C, HP-11C, and HP-15C. I do like many other HP models, of course, but they do not have a *"place in my heart"* as these models do (as for the SHARPs, the only ones to have a place in my heart are the PC-1211 and the PC-1350).

As for the HP-71B, it always was my "dream machine" back in the 80's, the one model I desired the most and my natural upgrade path from the HP-41C. It had everything I wanted (except for more speed and larger display), and I was extremely happy when I finally managed to get one, as told in my ***"Long Live the HP-71B!"*** article recently published in Datafile (which, by the way, does include an extremely short HP-71B program to create and print awesome text-based 3D stereograms).

As for feeling comfortable to try my challenges, I'm sure you'll manage pretty soon: the HP-71B is truly easy to program, extremely powerful, and my challenges aren't that hard to begin with. You'll be amazed to learn that many seemingly extremely difficult or convoluted tasks can be done in 9 lines or less of HP-71B code, frequently 5 lines will suffice. This is possible thanks to its extremely advanced programming features, such as subprograms with their own local environment, parameter passing by value and by reference, user-defined multi-line functions, string arrays, string slicing, full recursivity, etc, etc. I would recommend that you get hold of and read my HP-71B-related Datafile articles, most specially ***"HP-71B Math ROM Baker's Dozen"***, ***"HP-71B Minimax Polynomial"***, and the Sudoku "trilogy", to name a few.

However, there is one thing you absolutely *must* do: you must get the HP-71B Math ROM, as soon as you can. Without it you *don't* have an HP-71B, regardless of what they said to you. The real, full, true HP-71B is an HP-71B *plus* the Math ROM. Anything else, you don't have an HP-71B, period. :-) Matter of fact, nearly all my programs, articles, and challenges absolutely require it.

"Cheers and have a great weekend!"

Thank you very much, I wish you the same.

Best regards from V.

New Topic: HP 71b - Thanks for the tips!

Message #51 Posted by ***PeterP*** *on 19 Feb 2007, 3:35 p.m.,*
in response to message #50 by Valentin Albillo

Thanks Valentin for your encouraging words. Following them I went right away to the HPCC webpage, got the index of all HP71 related files and wanted to see if I can download them or by a CD/DVD. Unfortunately Jake's only goes until 2003 and most articles, including the ones you mentioned, are not available for download. While I am not familiar with authors for the 71, I would assume that next to your articles, Wlodek's and Graeme's articles would be good gets, too? Do you have any tips on how to get a hold of those articles? Thanks for the tip on the math module, I'll see that I can find one. Do you know of any one available from our community for purchase?

Thanks and all the best, I let you know how it goes :-)

Cheers

Peter

Re: Valentine's Day Mini-Challenges !

Message #52 Posted by ***Kiyoshi Akima*** *on 15 Feb 2007, 12:50 p.m.,*
in response to message #1 by Valentin Albillo

I tried this first on the 16C since I had one handy, and came up with this 11-stepper in integer mode:

```
01 LBL A
02 0
03 x<>y
04 LBL 1
05 5
06 /
07 +
08 LASTx
09 x#0?
10 GTO 1
11 x<>y
```

Working in integer mode eliminates the need to take the integer portion of the quotient, but the lack of register arithmetic costs steps. It handles numbers up to (2^64)-1 in unsigned mode, I think. ((2^64)-1)! ends with 4,611,686,018,427,387,890 zeros. (About 27 seconds.)

I won't post my 10-step 15C version since it's essentially the same as Gerson's (I used a numbered register).

For the second part, here's a major cheat for the 16C, two steps and constant speed regardless of argument:

```
01 LBL B
02 1
```

Alas, it only works in binary. Converting this to work in decimal, as required by the challenge, will be left as an exercise for the reader :-)

Re: Valentine's Day Mini-Challenges !

Message #53 Posted by ***Valentin Albillo*** *on 15 Feb 2007, 1:52 p.m.,*
in response to message #52 by Kiyoshi Akima

Veeeery clever ! :-) Give the man a cigar !

Best regards from V.

Re: Valentine's Day Mini-Challenges !

Message #54 Posted by ***Paul Dale*** *on 15 Feb 2007, 5:06 p.m.,*
in response to message #1 by Valentin Albillo

Sadly, too slow this time on the first part :-(

Nobody has mentioned the opetion of using clear sigma instead of clear regs and sigma+ instead of the sto+ This doesn't save any steps, but is a little less destructive.

Two steps longer but avoiding all register use:

```
    001 LBL A
    002 0
    003 X<>Y
    004 LBL 0
    005 5
    006 /
    007 INT
    008 +
    009 LASTx
    010 TEST 0    ;  x#0?
    011 GTO 0
    012 X<>Y
```

- Pauli

Re: Valentine's Day Mini-Challenges !

Message #55 Posted by ***Miguel Toro*** *on 15 Feb 2007, 9:48 p.m.,*
in response to message #1 by Valentin Albillo

Hello,

For the 15C, just a little optimisation without need of any register, assuming that the stack is cleared before entering the number:

```
001 LBL A
002 5
```

```
003 /
004 INT
005 +
006 LAST X
007 TEST 0
008 GTO A
009 X<>Y
```

Regards,

Miguel

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #56 Posted by ***Egan Ford*** *on 15 Feb 2007, 11:31 p.m.,*
in response to message #1 by Valentin Albillo

> Quote:
>
> ---
>
> ## The Name-Your-Model (15C) Mini-Challenge
>
> You must write a program for your HP calc which, given as input a positive integer N between 0 and 5,000,000,000 (for 10-digit models) or between 0 and 500,000,000,000 (for 12-digit models), it will compute and output the rightmost digit of N! *disregarding all trailing zeros*.
>
> ```
>          0 ->  1
>          1 ->  1
>         13 ->  8  (10 sec)
>         69 ->  4  (15 sec)
>       1000 ->  2  (20 sec)
>       2007 ->  2  (30 sec)
>    1234567 ->  4  (35 sec)
> 4567890123 ->  8  (55 sec)
> ```
>
> ---

This is just my first draft, I was shooting for readability. I'll work on speed and size later, perhaps another beats me to it.

15C: Least Significant Non-Zero Digit of n! (n > 1)

```
1 LBL A
2 CLEAR REG
3 STO 3
```

```
 4 1
 5 STO 2
 6 LBL 0
 7 RCL 3
 8 TEST 4
 9 GTO 1
10 5
11 /
12 FRAC
13 5
14 *
15 STO 4
16 x!
17 2
18 RCL 1
19 RCL 4
20 *
21 Y^X
22 *
23 1
24 0
25 /
26 FRAC
27 1
28 0
29 *
30 STO* 2
31 RCL 3
32 5
33 /
34 INT
35 STO 3
36 RCL 1
37 1
38 +
39 4
40 /
41 FRAC
42 4
43 *
44 STO 1
45 GTO 0
46 LBL 1
47 RCL 2
48 6
49 *
50 1
```

```
    51 0
    52 /
    53 FRAC
    54 1
    55 0
    56 *
    57 RTN
```

Edited: 16 Feb 2007, 11:48 a.m. after one or more responses were posted

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #57 Posted by ***Egan Ford*** *on 16 Feb 2007, 12:05 a.m.,*
in response to message #56 by Egan Ford

Hmmm... The domain for the above is 2 - 9,999,999,999.

UPDATE: Opps, not true. My perl prototype was limited to 10 digit numbers for all variables, I assumed the port to the 15C would also function. Perl unlike the 15C will return a FRAC for n > 5,000,000,000 / m. When attempting to mod 5 numbers > 5,000,000,000 the 15C always returns 0.

Edited: 16 Feb 2007, 4:38 p.m. after one or more responses were posted

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #58 Posted by ***Gerson W. Barbosa*** *on 16 Feb 2007, 12:07 a.m.,*
in response to message #57 by Egan Ford

Congratulations, Egan!

Can you please post some more examples, so I can test my program? So far I can say its domain is narrower, but I want to check the results.

Thanks,

Gerson.

Edited: 16 Feb 2007, 12:12 a.m.

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #59 Posted by ***Egan Ford*** *on 16 Feb 2007, 12:12 a.m.,*
in response to message #58 by Gerson W. Barbosa

```
1324420984 -> 2
1756073772 -> 2
6105143989 -> 2
6283630118 -> 8
 346030996 -> 8
1938887450 -> 8
7741346755 -> 4
1748471318 -> 8
7764025134 -> 2
1540062188 -> 2
9999999999 -> 2
```

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #60 Posted by ***Gerson W. Barbosa*** *on 16 Feb 2007, 12:25 a.m.,*
in response to message #59 by Egan Ford

```
1324420984 -> 2   Ok!
1756073772 -> 2
6105143989 -> 2   Ok!
6283630118 -> 8   Ok!
 346030996 -> 8   Ok!
1938887450 -> 8
7741346755 -> 4   Ok!
1748471318 -> 8
7764025134 -> 2
1540062188 -> 2   Ok!
9999999999 -> 2
```

Thanks and

Good-night!

Gerson.

Update: It's ok for all examples now, thanks to your suggestion. Please see the fixed version. Thanks!

Edited: 16 Feb 2007, 7:56 a.m.

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #61 Posted by ***Gerson W. Barbosa*** *on 17 Feb 2007, 12:01 p.m.,*
in response to message #57 by Egan Ford

> Quote:
>
> Perl unlike the 15C will return a FRAC for n > 5,000,000,000 / m. When attempting to mod 5 numbers > 5,000,000,000 the 15C always returns 0.

Keen observation! This explains why the upper limit on the HP-32SII is 500,000,000,000 and 999,999,999,999 on the HP-33S. The latter has the built-in RMDR function, not subject to rounding errors like the MOD routine on the first.

Regards,

Gerson.

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #62 Posted by ***Egan Ford*** *on 17 Feb 2007, 12:41 p.m.,*
in response to message #61 by Gerson W. Barbosa

> Quote:
>
> ... This explains why the upper limit on the HP-32SII is 500,000,000,000 ...

I am sure our challenger had that in mind. I see two practical solutions for this problem.

Older calcs (15C, 34C, 33C, 12C, 11C, etc...):

http://home.wlu.edu/~dresdeng/papers/lnzd2.pdf, page 10.

Newer calcs (71B, 48GX, 50G, etc...):

http://www.mathpages.com/home/kmath489.htm

Both solutions use n base 5 limiting the upper boundary for many models by half. But with a bit a clever work as you have already demonstrated can get around this. Perhaps the 5,000... upper limits were a clue.

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #63 Posted by ***Gerson W. Barbosa*** *on 17 Feb 2007, 2:13 p.m.,*
in response to message #62 by Egan Ford

The problem with the MOD routine is not exactly due to rounding errors, as I stated. HP calculators have two or three guard digits, the problem is the result is rounded to the number of digits on the display, otherwise the FRAC-based MOD routine would work, at least for n MOD 4, 5 or 10. The HP-42S version, with the 5-step MOD routine, works flawlessly on Free42 Decimal.

Edited: 17 Feb 2007, 2:14 p.m.

"MOD" and "Rmdr"

Message #64 Posted by ***Karl Schneider*** *on 17 Feb 2007, 4:22 p.m.,*
in response to message #63 by Gerson W. Barbosa

Hi, Gerson --

Quote:

> The problem with the MOD routine is not exactly due to rounding errors, as I stated. HP calculators have two or three guard digits, the problem is the result is rounded to the number of digits on the display, otherwise the FRAC-based MOD routine would work...

I belive that the "Rmdr" function on the HP-33S is "MOD" by a different name:

http://www.hpmuseum.org/cgi-sys/cgiwrap/hpmuseum/archv014.cgi?read=53359#53359

That's not to say there couldn't be some difference in the domain of the function.

Quote:

> The HP-42S version, with the 5-step MOD routine, works flawlessly on Free42 Decimal.

I didn't look at the example of code, but the HP-42S has MOD built in.

It's difficult to implement a robust and efficient MOD function as a user program on a calculator, even though the calculations are fairly simple. Built-in machine-code routines work far better. Every calculator ought to have one, for reduction of arguments to trigonometric functions.

-- KS

Re: "MOD" and "Rmdr"

Message #65 Posted by ***Gerson W. Barbosa*** *on 17 Feb 2007, 5:32 p.m.,*
in response to message #64 by Karl Schneider

Hello Karl,

> Quote:
>
> I belive that the "Rmdr" function on the HP-33S is "MOD" by a different name

Thanks for the link. I remember I had read that thread in another occasion. I haven't tried RMDR with negative arguments as you did, but to me RMDR stands for remainder in an integer division, that is, same as MOD. By the way, that's why I chose the 33S in the first place, not because it is my favorite calculator, but because of RMDR and INT/ (integer division) availability.

> Quote:
>
> I didn't look at the example of code, but the HP-42S has MOD built in.

Thanks for reminding me. It was hidden in the catalog and I had forgotten about it. Anyway, that was only a test to see how the program would perform on Free42 (instantaneously for any N). I did not listed the HP-42 version because it is essentially the same as the HP-33S version, now that the built-in MOD is being used (72-bytes longs, 46 lines). Of course, the domain remains extended to 999,999,999,999. As MOD, like RMDR on the HP-33S, has no constraints the program should work on the physical HP-42 also. A version you might find interesting is the one for the HP-33C, as it has no factorial built-in.

> Quote:
>
> Built-in machine-code routines work far better. Every calculator ought to have one, for reduction of arguments to trigonometric functions.

I realized this when writing the trig program for the HP-12CP. The errors increased as the arguments grew larger. While walking home thinking about the problem I found the solution: a 360 MOD routine based on INT rather than FRAC. It worked!

Best regards,

Gerson.

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #66 Posted by ***Les Wright*** *on 16 Feb 2007, 2:04 a.m.,*
in response to message #56 by Egan Ford

For several of the modular arithmetic divisions where you say FACT, you must mean FRAC, as it the HP41 factorial command?

Les

Edited: 16 Feb 2007, 2:33 a.m. after one or more responses were posted

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #67 Posted by ***Egan Ford*** *on 16 Feb 2007, 2:13 a.m.,*
in response to message #66 by Les Wright

Doh! Thanks, fixed.

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #68 Posted by ***Les Wright*** *on 16 Feb 2007, 2:42 a.m.,*
in response to message #67 by Egan Ford

The mod 4 reduction of the i in the exponent is very clever. I don't understand basic number theory well enough to have picked up on that and I am still not clear how it works.

The product series on the OEIS page only works for n > 1, so I think that any routine needs and exit to return the correct answer of 1 for 0! and 1!

I was going to work on this myself but took a nap first. Looks like other folks were thinking along the lines I was. I didn't anticipate the problem of the accumulated product blowing up enough so that taking mod 10 of it would just return zero.

What a fun problem!

Les

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #69 Posted by ***Egan Ford*** *on 16 Feb 2007, 3:00 a.m.,*
in response to message #68 by Les Wright

Quote:

> The mod 4 reduction of the i in the exponent is very clever. I don't understand basic number theory well enough to have picked up on that and I am still not clear how it works.

2^x mod 10, n=1,2,3,... is cyclical, i.e. 2,4,8,6,2,4,8,6,...

e.g.

2^x mod 10, n=1,5,9,... is always 2.

Edited: 16 Feb 2007, 8:41 a.m. after one or more responses were posted

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

*Message #70 Posted by **Les Wright** on 16 Feb 2007, 4:18 a.m.,*
in response to message #69 by Egan Ford

You mean 2^n in these expressions, right? We are talking about exponentiation?

But I do understand the principle--the mod 4 cyclic nature of the exponentiation of 2 in mod 10 means that every fourth i will map to the same mod 10 equivalence class.

It that about right? Very clever!

At the beginning of code you seem to test if n is lt zero. Bit curious about that. Shouldn't be the test for n lt 2, and default to an output of 1 for input of 0 and 1?

Les

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

*Message #71 Posted by **Egan Ford** on 16 Feb 2007, 9:07 a.m.,*
in response to message #70 by Les Wright

> Quote:
>
> You mean 2^n in these expressions, right? We are talking about exponentiation?

Yes. I should have use ^ or **, in my haste I lost the 2nd * in my example. Late night. Fixed.

Quote:

> But I do understand the principle--the mod 4 cyclic nature of the exponentiation of 2 in mod 10 means that every fourth i will map to the same mod 10 equivalence class.
>
> It that about right? Very clever!

Yes, that was the idea.

Quote:

> At the beginning of code you seem to test if n is lt zero. Bit curious about that. Shouldn't be the test for n lt 2, and default to an output of 1 for input of 0 and 1?

The product calculating loop must also convert n to base 5. To do that you mod 5 to get the next lest significant digit (for base 5), then n becomes int(n/5), that process must continue while n > 0. The check is at the top of a loop to see if n has been completely converted to base 5.

In my laziness I omitted an early check for n < 2. I was going to add that to the optimized solution. BTW, n FRAC 0 > should error out.

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #72 Posted by ***Egan Ford*** *on 16 Feb 2007, 11:17 a.m.,*
in response to message #56 by Egan Ford

There is a bug in the 15C solution above with mod 4, e.g. 625 and 626 fail. I omitted the 6 * because 6*n mod 10 = n mod 10 when n is even, but with the mod 4 added to support larger n it is possible to get 1 from the mod 10 of the product.

Gerson's 33S solution with mod 4 didn't omit the 6 * and should not have a problem.

Edited: 16 Feb 2007, 11:35 a.m.

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #73 Posted by ***Gerson W. Barbosa*** *on 16 Feb 2007, 12:12 p.m.,*
in response to message #72 by Egan Ford

Quote:

> Gerson's 33S solution with mod 4 didn't omit the 6 * and should not have a problem

I get 6 for both 625 and 626, which is correct. But if my version is fully working, the credit goes to you!

Regards,

Gerson

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

*Message #74 Posted by **Egan Ford** on 16 Feb 2007, 4:33 p.m.,*
in response to message #56 by Egan Ford

Updated 15C code with domain of 0-5,000,000,000. Complete, smaller, but a bit slower than previous code.

If you want this to go faster, then inline the mods, unroll the loop 4 times, lose the i mod 4 and hardcode i=0,1,2,3 for the 4 loops.

```
             0 ->  1  (01 sec)
             1 ->  1  (01 sec)
            13 ->  8  (11 sec)
            69 ->  4  (16 sec)
          1000 ->  2  (24 sec)
          2007 ->  2  (24 sec)
       1234567 ->  4  (41 sec)
    4567890123 ->  8  (66 sec)
    5000000000 ->  4  (64 sec)
```

Code:

```
    1 LBL A
    2 CLEAR REG
    3 1
    4 STO 2
    5 TEST 9 x>=y
    6 RTN
    7 X<>Y
    8 LBL 0
    9 TEST 4
   10 GTO 1
   11 ENTER
   12 ENTER
   13 5
```

```
  14 GSB 2
  15 x!
  16 LSTx
  17 RCL 1
  18 *
  19 2
  20 X<>Y
  21 Y^X
  22 *
  23 1
  24 0
  25 GSB 2
  26 STO *2
  27 X<>Y
  28 5
  29 /
  30 INT
  31 RCL 1
  32 1
  33 +
  34 4
  35 GSB 2
  36 STO 1
  37 X<>Y
  38 GTO 0
  39 LBL 1
  40 RCL 2
  41 6
  42 *
  43 1
  44 0
  45 LBL 2
  46 /
  47 LSTx
  48 X<>Y
  49 FRAC
  50 *
  51 RTN
```

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #75 Posted by ***Egan Ford*** *on 17 Feb 2007, 10:21 a.m.,*
in response to message #74 by Egan Ford

Same as above but with inline mod, loop unrolling, and some constant expressions pre-evaluated. Time is almost cut in half. Draft, more can be done.

```
            0 ->  1  (01 sec)
            1 ->  1  (01 sec)
           13 ->  8  (06 sec)
           69 ->  4  (10 sec)
         1000 ->  2  (14 sec)
         2007 ->  2  (13 sec)
      1234567 ->  4  (24 sec)
   4567890123 ->  8  (37 sec)
   5000000000 ->  4  (35 sec)
```

Code:

```
 1 LBL A
 2 CLEAR REG
 3 1
 4 STO 1
 5 TEST 9 x>=y
 6 RTN
 7 X<>Y
 8 LBL 0
 9 TEST 4 #loop 0
10 GTO 1
11 ENTER
12 ENTER
13 5
14 /
15 FRAC
16 5
17 *
18 x!
19 1
20 0
21 /
22 FRAC
23 1
24 0
25 *
26 STO *1
27 X<>Y
28 5
29 /
30 INT
31 TEST 4 #loop 1
32 GTO 1
33 ENTER
34 ENTER
35 5
```

```
36 /
37 FRAC
38 5
39 *
40 x!
41 LSTx
42 2
43 X<>Y
44 Y^X
45 *
46 1
47 0
48 /
49 FRAC
50 1
51 0
52 *
53 STO *1
54 X<>Y
55 5
56 /
57 INT
58 TEST 4 #loop 2
59 GTO 1
60 ENTER
61 ENTER
62 5
63 /
64 FRAC
65 5
66 *
67 x!
68 LSTx
69 2 #i=2
70 *
71 2
72 X<>Y
73 Y^X
74 *
75 1
76 0
77 /
78 FRAC
79 1
80 0
81 *
82 STO *1
```

```
 83 X<>Y
 84 5
 85 /
 86 INT
 87 TEST 4 #loop 3
 88 GTO 1
 89 ENTER
 90 ENTER
 91 5
 92 /
 93 FRAC
 94 5
 95 *
 96 x!
 97 LSTx
 98 3 #i=3
 99 *
100 2
101 X<>Y
102 Y^X
103 *
104 1
105 0
106 /
107 FRAC
108 1
109 0
110 *
111 STO *1
112 X<>Y
113 5
114 /
115 INT
116 GTO 0
117 LBL 1
118 RCL 1
119 6
120 *
121 1
122 0
123 /
124 FRAC
125 1
126 0
127 *
128 RTN
```

Edited: 17 Feb 2007, 10:22 a.m.

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #76 Posted by ***Gerson W. Barbosa*** *on 17 Feb 2007, 11:53 a.m.,*
in response to message #74 by Egan Ford

> Quote:
>
> Updated 15C code with domain of 0-5,000,000,000.

The domain can be extended up to 9,999,999,999 by changing the MOD routine:

```
    45 LBL 2
    46 X<>Y
    47 STO 3
    48 X<>Y
    49 /
    50 LSTx
    51 X<>Y
    52 INT
    53 *
    54 CHS
    55 RCL+ 3
    56 RTN
```

```
7777777777 ->  4  (87 sec)
```

Previously returned 2 in 70 seconds. A bit slower, but the MOD routine can still be optimized.

Regards,

Gerson.

Re: Valentine's Day Mini-Challenges ! -- part 2 for 15C

Message #77 Posted by ***Egan Ford*** *on 17 Feb 2007, 12:19 p.m.,*
in response to message #76 by Gerson W. Barbosa

It is only a problem for the first itteratoin of the loop, so this could be inlined and only used for a[0]. That may help with the speed at the cost of a few more lines of code.

Re: Valentine's Day Mini-Challenges ! -- extended domain and HP-41 & HP-12C versions

Message #78 Posted by ***Gerson W. Barbosa*** *on 18 Feb 2007, 6:55 p.m.,*
in response to message #77 by Egan Ford

Looks like I made a mistake when I said

The domain can be extended up to 9,999,999,999 by changing the MOD routine...

Replacing the MOD routine solves *half* the problem. The other problem is the division by 5 being rounded up and giving an error of one unit in the last significant digit for some numbers greater than 5,000,000,000, as explained by Valentin in his *Original Solutions and Comments*. As an example, the HP-41 program below has not an extended domain, despite the built-in MOD function. On the other hand, if run on an HP-12C Platinum the second program below should give correct answers for arguments up to 9,999,999,999.

Regards,

Gerson.

```
------------------------------------------------


HP-41 version


01>LBL "LNZD"
02 STO 00
03 0
04 STO 01
05 E^X
06 STO 02
07 X<>Y
08 X<=Y?
09 GTO 02
10>LBL 00
11 RCL 00
12 5
13 MOD
14 FACT
15 2
16 LASTX
17 RCL 01
18 *
19 Y^X
20 *
```

```
21 10
22 MOD
23 ST* 02
24 RCL 00
25 5
26 /
27 INT
28 X=0?
29 GTO 01
30 STO 00
31 RCL 01
32 1
33 +
34 4
35 MOD
36 STO 01
37 GTO 00
38>LBL 01
39 RCL 02
40 6
41 *
42 10
43 MOD
44 RTN
45>LBL 02
46 X<>Y
47 RTN
               HP-41CX 2819S*****


               0 ->  1  (00.9 sec)
               1 ->  1  (00.9 sec)
              13 ->  8  (03.3 sec)
              69 ->  4  (04.5 sec)
            1000 ->  2  (06.7 sec)
            2007 ->  2  (06.7 sec)
         1234567 ->  4  (11.7 sec)
      4567890123 ->  8  (18.6 sec)
      5000000000 ->  4  (17.6 sec)


      5000000003 ->  8  wrong, should be 4!


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


HP-12C version
```

```
01 STO 0
02 0
03 STO 1
04 e^x
05 STO 2
06 x<>y
07 x<=y?
08 GTO 60
09 RCL 0
10 5
11 /
12 LSTx
13 x<>y
14 FRAC
15 *
16 n!
17 2
18 LSTx
19 RCL 1
20 *
21 y^x
22 *
23 1
24 0
25 /
26 LSTx
27 x<>y
28 FRAC
29 *
30 ST0* 2
31 RCL 0
32 5
33 /
34 INTG
35 x=0?
36 GTO 49
37 STO 0
38 RCL 1
39 1
40 +
41 4
42 /
43 LSTx
44 x<>y
45 FRAC
46 *
```

```
47 STO 1
48 GTO 09
49 RCL 2
50 6
51 *
52 1
53 0
54 /
55 LSTx
56 x<>y
57 FRAC
58 *
59 GTO 00
60 x<>y
61 GTO 00


                  HP-12C CN048*****


               0 R/S ->  1  (01 sec)
               1 R/S ->  1  (01 sec)
              13 R/S ->  8  (07 sec)
              69 R/S ->  4  (10 sec)
            1000 R/S ->  2  (16 sec)
            2007 R/S ->  2  (15 sec)
         1234567 R/S ->  4  (28 sec)
      4567890123 R/S ->  8  (44 sec)
      5000000000 R/S ->  4  (42 sec)


      5000000003 R/S ->  8  wrong, should be 4!
------------------------------------------------
```

Edited: 18 Feb 2007, 6:58 p.m.

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-33C

*Message #79 Posted by **Gerson W. Barbosa** on 16 Feb 2007, 7:39 p.m.,*
in response to message #1 by Valentin Albillo

Hello Valentin,

Here is a version for the modest HP-33C (n=0 and n=1 not handled) :

```
01  STO 0
02  CLx
03  STO 1
04  1
05  STO 2
06  RCL 0
07  5
08  GSB 44
09  STO 3
10  x^2
11  5
12  /
13  e^x
14  INT
15  2
16  RCL 3
17  RCL 1
18  *
19  y^x
20  *
21  1
22  0
23  GSB 44
24  STO* 2
25  RCL 0
26  5
27  /
28  INT
29  x=0
30  GTO 39
31  STO 0
32  1
33  STO+ 1
34  RCL 1
35  4
36  GSB 44
37  STO 1
38  GTO 06
39  RCL 2
40  6
41  *
42  1
43  0
44  /
45  LAST x
46  x<>y
```

```
47  FRAC
48  *
49  RTN



            13 GSB 01 ->  8 (10 s)
            69 GSB 01 ->  4 (15 s)
          1000 GSB 01 ->  2 (24 s)
          2007 GSB 01 ->  2 (25 s)
       1234567 GSB 01 ->  4 (45 s)
    4567890123 GSB 01 ->  8 (72 s)
```

Best regards,

Gerson.

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-33C (full specified range)

Message #80 Posted by ***Gerson W. Barbosa*** *on 18 Feb 2007, 2:20 p.m.,*
in response to message #79 by Gerson W. Barbosa

Hello again, Valentin!

Here is an optimized solution for the HP-33C. The HP-33C version is particularly interesting as this calculator has no built-in factorial function, as you are aware of. This improved version handles N=0 and N=1.

```
01  x=0
02  1
03  STO 0
04  0
05  STO 1
06  e^x
07  STO 2
08  x=y
09  RTN
10  RCL 0
11  5
12  GSB 45
13  STO 3
14  x^2
15  5
16  /
```

```
17  e^x
18  INT
19  2
20  RCL 3
21  RCL 1
22  *
23  y^x
24  GSB 42
25  STO* 2
26  RCL 0
27  5
28  /
29  INT
30  x=0
31  GTO 40
32  STO 0
33  RCL 1
34  1
35  +
36  4
37  GSB 45
38  STO 1
39  GTO 10
40  RCL 2
41  6
42  *
43  1
44  0
45  /
46  LAST x
47  x<>y
48  FRAC
49  *


             0 GSB 01 ->  1 ( 1 s)
             1 GSB 01 ->  1 ( 1 s)
            13 GSB 01 ->  8 (10 s)
            69 GSB 01 ->  4 (15 s)
          1000 GSB 01 ->  2 (24 s)
          2007 GSB 01 ->  2 (25 s)
       1234567 GSB 01 ->  4 (44 s)
    4567890123 GSB 01 ->  8 (70 s)
```

Lines 33-35 is borrowed code from Egan Ford. Well, I had borrowed his MOD 4 idea here, why not borrow also the more elegant code? :-)

Regards,

Gerson.

Edited: 19 Feb 2007, 6:53 a.m. after one or more responses were posted

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-33C (full specified range)

Message #81 Posted by ***Valentin Albillo*** *on 19 Feb 2007, 5:48 a.m.,*
in response to message #80 by Gerson W. Barbosa

Hi, Gerson !

First of all, thanks a lot for your continued efforts and interest in this mini-challenge, much appreciated.

Also, your results are truly worthwhile, most specially taking into account the very modest HP models you've been providing solutions for. Using an HP-71B is one thing, providing an efficient solution using an HP-33C is quite another.

However, may I point out what I think is a typo in your listing above, which perhaps you would deem appropriate to correct, namely:

```
  [...]
10  5
11  GSB 45
03  STO 3
14  x^2
15  5
  [...]
```

Seems to me that you're *missing* steps 12 and 13 above, and have pasted instead step 03 by mistake, a typical copy-paste oversight. Please correct if possible.

Best regards from V.

Re: Valentine's Day Mini-Challenges ! Part 2 - HP-33C (full specified range)

Message #82 Posted by ***Gerson W. Barbosa*** *on 19 Feb 2007, 7:39 a.m.,*
in response to message #81 by Valentin Albillo

Hello Valentin,

Thanks for pointing out the mistake! Fixed!

When I noticed the program was about 50 steps long, I thought it would be easy to port it to the HP-33C. Then I remembered it lacked factorial... I had two factorial routines that could in principle be inlined in placed of n!, after an adaptation to save the stack register X:

```
01 x=0                01 x=0
02 e^x                02 e^x
03 ENTER              03 STO 0
04 ENTER              04 STO/ 0
05 1                  05 STO* 0
06 -                  06 1
07 x=0                07 -
08 GTO 12             08 x=0
09 *                  09 GTO 11
10 LAST x             10 GTO 05
11 GTO 05             11 RCL 00
12 Rv
```

The first is my own factorial routine using only the stack. The second is the one in your excellent SF short-story Time Voyager, to which I added the first two lines as 0! would also be needed. Unfortunately, despite they are short they wouldn't fit. But, as only factorials of 0 through 4 are necessary I tried a simple curve-fitting, thus saving 6 or 7 precious steps:

```
14  x^2
15  5
16  /
17  e^x
18  INT
```

Thanks for your always interesting mini-challenges, solutions and comments.

Best regards,

Gerson.

Edited: 19 Feb 2007, 2:54 p.m.

Valentine's Day MC: My Original Solutions and Comments ! [LONG]

Message #83 Posted by ***Valentin Albillo*** *on 18 Feb 2007, 1:13 p.m.,*
in response to message #1 by Valentin Albillo

Hi, all:

First of all, thank you so much for the overwhelming interest you took in this humble mini-challenge of mine ! There's been an incredible number of posts, which, as was bound to happen, finally solved and nailed both parts up to their excruciatingly minor details.

I've been impressed with the many extremely clever solutions posted for such a variety of machines, and the incredible amounts of optimization to polish the resulting routines to the max. Also, the theoretical base for it all hasn't been neglected either, so I'm really sure that apart from the enjoyment of producing a working routine, there was also a lot of didactic math to be enjoyed and fruitfully assimilated.

My original solutions and comments follow:

Part I: Given a positive integer N from 0 to 5,000,000,000, write a routine to output the number of trailing zeros of N!

My original solution is the following 10-step RPN routine for the HP-15C:

```
01  LBL A
02  CLREG
03  LBL 0
04   5
05   /
06  INT
07  STO+ 0
08  TEST 0  (X<>0?)
09  GTO 0
10  RCL 0
```

which doesn't make any assumptions whatsoever about stack's or registers' contents and which simply implements the well-known formula (where [ ] stands for integer part):

```
    Number of trailing zeros of N! = [N/5] + [N/5^2] + [N/5^3] + ...
```

which is a *finite* sum because eventually the integer part of the quotient becomes zero for sufficiently large powers of 5. The reason for the upper limit being 5,000,000,000 is the fact that from 5,000,000,003 onwards, [N/5] may be off the true value by one unit. For instance, the HP-15C computes [5000000003/5] as 1000000001, instead of the correct value 1000000000, because internally the result is 1000000000.6, which gets rounded upwards when delivered to the user by the division routine.

This routine, with the trivial change of **TEST 0** for the equivalent **X<>0?**, will also work in many other classic RPN models. For those which don't have labels, an additional step can be saved. It is also possible to use register **I** instead of register **0**, and there are several other minor variations. As I said beforehand, this part was actually very easy and just an appetizer for the second part.

Part II: Given a positive integer N from 0 to 5,000,000,000 (for 10-digit models) or to 500,000,000,000 (for 12-digit models), write a routine to output the last nonzero digit of N!

Again, I've been awed by the clever routines posted to solve this part of the challenge for assorted HP models, including HP-15C versions. This indeed goes to prove that nothing short of sheer impossibility will deter most of you from getting a working solution, then refine it as if driven by the motto: *"Shorter, Faster !"*. Hats off to your skills and perseverance.

My original solution for the HP-71B is the following 6-liner:

```
1  DESTROY ALL @ OPTION BASE 0 @ DIM D(17),A$(3)[25] @ READ A$
2  DATA 0626422428448466626488682,0288424668482266288486442
3  DATA 0424428488468666424482622,0882426648422866882484462
4  INPUT "N=";N @ IF N<2 THEN DISP 1 @ END ELSE M=0 @ S=0
5  D(M)=MOD(N,5) @ N=N DIV 5 @ IF N THEN M=M+1 @ GOTO 5
6  FOR I=M TO 0 STEP -1 @ T=5*S/2+1+D(I) @ S=VAL(A$(MOD(I,4))[T,T]) @ NEXT I @ DISP S
```

which simply uses a suitably constructed lookup table (as duplicated by Kiyoshi Akima in his RPL version) where the indexes depend on both the successive digits of N when represented in base 5, and the value extracted from the table in the previous cycle . My routine dimensions a couple of arrays, one to store the individual digits of N in base 5, the other to store the lookup table, which contains four strings of digits, each string composed of 5 blocks of 5 consecutive final nonzero digits each, then proceeds as follows:

- Line 1 dimensions the arrays and reads the lookup data.
- Lines 2 and 3 are simply the data for the lookup table.
- Line 4 prompts the user to input the value of N, deals with the special cases of N being 0 or 1, then initializes some variables.
- Line 5 converts N to base 5, storing each individual digit in array D( ) and keeping a tally of the number of base-5 digits.
- Finally, line 6 simply uses each base-5 digit in turn to serve as part of an index which gets us from block to block till the final digit is extracted from the lookup data string and output.

Running time is proportional to the number of digits of N in base 5, which means it's proportional to the log5 of N. For any input from 0 to 500,000,000,000 the running time is negligible, as less than 20 base-5 digits are involved. As it was the case with Part I, values over 5E11 can give wrong results as, for instance, INT((5E11+3)/5) is returned as 1E11+1 instead of the correct 1E11.

Well, that's all for now. Thanks again for making this humble mini-challenge a great success of participation, both quantity- and quality-wise. I sincerely hope you enjoyed it and found the effort worthwhile !

Best regards from V.

Re: Valentine's Day MC: A few modest insights

Message #84 Posted by ***Karl Schneider*** *on 18 Feb 2007, 5:04 p.m.,*
in response to message #83 by Valentin Albillo

Hi all --

I may not develop and post solutions to the simpler parts of Valentin's challenges, mainly because others "beat me to it" so quickly -- especially those in Europe who see them right away and are off work and free to peruse them.

The first part in particular was an enlightening, yet easily accessible puzzle. For the benefit of those who didn't "plunge in" or follow the threads, here is some modest commentary:

> Quote:
>
> ---
>
> Part I: Given a positive integer N from 0 to 5,000,000,000, write a routine to output the number of trailing zeros of N!
>
> My original solution is the following 10-step RPN routine for the HP-15C:
>
> ```
> 01  LBL A
> 02  CLREG
> 03  LBL 0
> 04   5
> 05   /
> 06  INT
> 07  STO+ 0
> 08  TEST 0  (X<>0?)
> 09  GTO 0
> 10  RCL 0
> ```
>
> which doesn't make any assumptions whatsoever about stack's or registers' contents and which simply implements the well-known formula (where [ ] stands for integer part):
>
> Number of trailing zeros of N! = $[N/5] + [N/5^2] + [N/5^3] + \dots$
>
> ---

Well, I'd never seen the formula! :-)

Thinking about the problem, it's reasonably obvious that the number of 10's that can be extracted from all of the terms in the factorial gives the number of trailing zeroes. Each matched pair of 2 and 5 comprises a 10, and it's also evident that there is an abundant surplus of 2's in the terms. Thus, the problem becomes one of finding an efficient method to "count fives".

It's easy to see the pattern in, for example, 132 factorial. The terms that include 5 are the following:

```
term:     5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130
# of 5's: 1  1  1  1  2  1  1  1  1  2  1  1  1  1  2  1  1  1  1  2   1   1   1   1   3   1
```

The best way to sum up the 5's is to determine by division how many terms are a multiple of five, add to that the number of terms that are also a multiple of five squared (25), and so forth. The iterative loop program to do this is quite straightforward, as shown. The fractional remainders of division are discarded, because they are not of interest.

An indirect way to pose the problem is, "How many *significant digits* does the factorial of a given number have? This, of course, would be the number of digits in the factorial minus the number of trailing zeroes. Posing the problem that way would restrict its domain, though...

Thanks to Valentin and best regards,

-- KS

Edited: 18 Feb 2007, 10:03 p.m. after one or more responses were posted

Trailing zeroes...

Message #85 Posted by ***Gene*** *on 18 Feb 2007, 5:46 p.m.,*
in response to message #84 by Karl Schneider

This was an example I used when I did the virtual classroom training for the HP graphing calculators.

On the 49g+/50g, I would ask "How many zeroes are at the end of 100! ?"

The answer was to factor the item and see the lower of the number of 2's or 5's. That would be the number of zeroes.

Of course, that's easier to see in the EquationWriter than for me to program. :-)

Re: Valentine's Day MC: A few modest insights

Message #86 Posted by ***Valentin Albillo*** *on 19 Feb 2007, 4:47 a.m.,*
in response to message #84 by Karl Schneider

Hi, Karl:

Karl posted:

"Well, I'd never seen the formula! :-)"

It's pretty well known, believe me. I was first aware of this topic as a teenager. You can find it here, for instance:

Ask Dr. Math - The Number of Zeros in a Factorial"

"An indirect way to pose the problem is, "How many significant digits does the factorial of a given number have?"

Nope. You're getting the wrong interpretation of "*significant digits"* in your reposing of the problem. The usual definitions of these terms are:

"The digits of the decimal form of a number beginning with the leftmost nonzero digit and extending to the right to include all digits warranted by the accuracy of measuring devices used to obtain the numbers. Also called significant figures."

"Those digits in a number that add value to the number. For example, in the number 00006508, 6508 are the significant digits."

So you see, the trailing, i.e., rightmost zeros of the number but still at the *left* of the (implied) decimal point are pretty significant. They wouldn't be if they were the leftmost digits at the left of the decimal point, or the rightmost digits at the right of the decimal point. But being the rightmost digits at the *left* of the (implied) decimal point makes them significant, in spades.

After all, would you tolerate that someone paid you $12 instead of $1200 on the grounds that the suppressed zeros aren't to be considered as "significant digits", as per your initial proposal ? :-)

Thanks for your kind words, your unfailing interest, and most specially your always interesting and didactic comments, and

Best regards from V.

Re: Valentine's Day MC: A few modest insights

Message #87 Posted by ***Giancarlo (Italy)*** *on 19 Feb 2007, 7:47 a.m.,*
in response to message #86 by Valentin Albillo

What an interesting thread!
I've got used to select my favourite threads and "print" them as a PDF file
using a "virtual PDF printer" (one of many freewares available on the net:)

Well, this very thread of Valentin's is currently 53 pages long, and doesn't seem to be ending soon, hopefully!
By the way, anyone remembering the longest thread in the HPMuseum forum?
Best regards.
Giancarlo

Edited: 19 Feb 2007, 7:48 a.m.

Re: Valentine's Day MC: A few modest insights

Message #88 Posted by ***Valentin Albillo*** *on 19 Feb 2007, 8:26 a.m.,*
in response to message #87 by Giancarlo (Italy)

Hi, Giancarlo:

Giancarlo posted:

> *"What an interesting thread!"*
>
> > Thank you very much, I'm truly glad so many people did find it worthwhile to invest their time in solving these mini-challenges in so many different HP models and continually optimizing and improving to achieve the utmost performance even for the humblest of them. That's the HP way: striving for excellence, nothing less will do ! :-)
>
> *"I've got used to select my favourite threads and "print" them as a PDF file [...] Well, this very thread of Valentin's is currently 53 pages long"*
>
> > Well, it's sort of funny because I did the same thing, just after your post, and it came up to 83 pages, not 53. Are you sure you aren't losing the right part of deeply nested messages ? Aren't your lines badly truncated at many places ? :-)
> >
> > I suggest you check and, if necessary, print in landscape format and even landscape plus 2 pages per "printed" page, or in a smaller font. Else, you're probably badly losing contents.
>
> *"By the way, anyone remembering the longest thread in the HPMuseum forum?"*
>
> > I'm pretty sure the archives are full of eVay garbage threads much, much longer than this minichallenge's one.
>
> Thanks for your comments and worthwhile ideas, and

Best regards from V.

Re: Valentine's Day MC: A few modest insights

Message #89 Posted by ***Giancarlo (Italy)*** *on 19 Feb 2007, 4:26 p.m.,*
in response to message #88 by Valentin Albillo

Hi Valentin.

> Quote:
>
> Are you sure you aren't losing the right part of deeply nested messages ? Aren't your lines badly truncated at many places ? :-)

As usual, I immediately got your advise and checked, but there are no bad
truncations even in the most nested posts - the virtual PDF printer seems to do the right work...
But I prefer to think of it as to how wonderfully Dave Hicks designed his site!
Hmmm... you just reminded me that I could use another freeware (Fine Print, did you hear about it?)
which allows to print up to 8 pages of a document on a sheet of paper...
Warmest regards.
Giancarlo

Edited: 19 Feb 2007, 4:35 p.m.

Significant digits -- well, yes and no...

Message #90 Posted by ***Karl Schneider*** *on 19 Feb 2007, 9:45 p.m.,*
in response to message #86 by Valentin Albillo

Hi, Valentin --

I hadn't fully thought through my reference to the string of digits preceding the trailing zeroes in a calculated interger (n!) as the "significant digits", and I stand corrected.

Since the factorial result is an exact number, the trailing zeroes should indeed be considered significant, although not absoutely necessary for exactitude. Scientific notation would generally be more convenient for *expressing* the exact result for large n!, identified as such:

40! = "exactly 8.15915283247897734345611269596115894272 x 10^{47}"

is preferable to

40! = "815915283247897734345611269596115894272000000000"

> Quote:
>
> ---
>
> So you see, the trailing, i.e., rightmost zeros of the number but still at the left of the (implied) decimal point are pretty significant.
>
> They wouldn't be if they were the leftmost digits at the left of the decimal point, or the rightmost digits at the right of the decimal point.
>
> But being the rightmost digits at the left of the (implied) decimal point makes them significant, in spades.
>
> ---

Are you referring to these factorial calculations in particular, or to numbers in general? If the latter is the case, I can't fully agree with these statements, and would refer to the definitions you quoted:

> *"The digits of the decimal form of a number beginning with the leftmost nonzero digit and extending to the right to include all digits warranted by the accuracy of measuring devices used to obtain the numbers. Also called significant figures."*
>
> *"Those digits in a number that add value to the number. For example, in the number 00006508, 6508 are the significant digits."*

The operative phrases are "warranted by the accuracy" and "add value".

The trailing zeroes to the left of radix point (implied or not) are significant only if they represent measured or calculated values that are accurate to the stated digits. The same holds for zeroes to the right of the radix point, even if no non-zero digits follow them.

Leading zeroes *on either side* of a decimal point are never significant, as implied by the quoted definitions.

A distance of 40 km, accurate to the nearest meter, could be written properly as "40.000 km", "40000. meters" or "4.0000 x 10^4 m" with five significant digits. It would be improper to write "40000000. mm", though, as that would imply eight significant digits. Changing the measurement scale cannot change the number of significant digits, even if extra zeroes are needed as placeholders for correct magnitude.

When I calculate $\sin^{-1}$ (0.5) or $\cos^{-1}$ (0.5) on an HP-42S and get 30.0000000000 and 60.0000000000 respectively as the results, I accept that answers correct to 12 significant digits were returned.

A couple more links from a Dr. Peterson at "Ask Dr. Math":

Significant Digits and Zero (1999)

Significant Digits and Irrational Numbers (2005)

His first response states that trailing zeroes are in general not significant, but *could* be. His second response stated, "Any exact number (not a measurement, but a known number such as pi or sqrt(2)) is considered to have infinitely many significant digits..."

Best regards,

-- KS

Edited: 19 Feb 2007, 10:11 p.m.

Re: Significant digits -- well, yes and no...

Message #91 Posted by ***Dave Shaffer*** *on 20 Feb 2007, 5:56 p.m.,*
in response to message #90 by Karl Schneider

"When I calculate sin-1 (0.5) or cos-1 (0.5) on an HP-42S and get 30.0000000000 and 60.0000000000 respectively as the results, I accept that answers correct to 12 significant digits were returned."

I'd be a bit careful about this statement, too!

If the input (0.5) is significant to only one digit (as implied), then the output is significant to only one digit, too: 30 or 60 would be correct (at least in my physics classes!), with the realization that the trailing zero is NOT significant.

I'd argue that you need arcsin(0.500000000000) to give a result (30.0000000000) with 12 significant digits.

Re: Significant digits -- well, yes and no...

Message #92 Posted by ***Karl Schneider*** *on 20 Feb 2007, 11:55 p.m.,*
in response to message #91 by Dave Shaffer

Hi, Dave --

> Quote:
>
> *"When I calculate sin^{-1} (0.5) or cos^{-1} (0.5) on an HP-42S and get 30.0000000000 and 60.0000000000 respectively as the results, I accept that answers correct to 12 significant digits were returned."*
>
> I'd be a bit careful about this statement, too!

> If the input (0.5) is significant to only one digit (as implied), then the output is significant to only one digit, too: 30 or 60 would be correct (at least in my physics classes!), with the realization that the trailing zero is NOT significant.
>
> I'd argue that you need arcsin(0.500000000000) to give a result (30.0000000000) with 12 significant digits.

The input received by any calculator is always assumed by the machine to be *exact*; no trailing zeroes need be entered. If the actual value of the input is not exactly the value entered, it is the user's responsibility to appropriately round the result returned.

The HP-42S (or any Pioneer series) accepts 12-digit input. Because of binary-coded decimal (BCD), these inputs are exactly-represented internally, as well.

The answers returned by Pioneers are *accurate to 12 digits*, all of which are significant on that basis. Full accuracy cannot always be claimed for the KinHPo 33S (due to some faulty algorithms), or for the 10-digit pre-Pioneer models (perhaps due to abbreviated algorithms).

My favorite examples:

HP-42S: sin (3.14159265358 rad) = 9.79323846264 x 10^{-12} -- correct result to 12 significant digits

HP-41: sin (3.141592653 rad) = 5.9 x 10^{-10} -- correct result to 2 significant digits

HP-32SII: cos (89.9999999 deg) = 1.74532925199 x 10^{-9} -- correct result to 12 significant digits

KinHPo 33S: cos (89.9999999 deg) = 1.74532000000 x 10^{-9} -- incorrect result to 6 significant digits...

Best regards,

-- KS

Edited: 21 Feb 2007, 4:05 a.m. after one or more responses were posted

Re: Valentine's Day MC: A few modest insights, n! digits - part 3, anyone?

Message #93 Posted by ***Egan Ford*** *on 19 Feb 2007, 11:25 p.m.,*
in response to message #84 by Karl Schneider

> Quote:

> An indirect way to pose the problem is, "How many *significant digits* does the factorial of a given number have? This, of course, would be the number of digits in the factorial minus the number of trailing zeroes. Posing the problem that way would restrict its domain, though...

Number of digits n! for the 15C. Domain 1 - 10^10/12 because of n*12. Because of n*log(n) rounding errors large numbers may be over by one. Quick and dirty first draft.

```
          1 ->  1
         13 ->  10
         69 ->  99
       1000 ->  2568
       2007 ->  5759
    1234567 ->  6984221
  833333333 ->  7072103563 (should be 1 less)
```

Code:

```
 1 LBL A
 2 STO 1
 3 4
 4 *
 5 1/x
 6 RCL 1
 7 +
 8 5
 9 *
10 2
11 SWAP
12 /
13 1
14 2
15 RCL 1
16 *
17 +
18 1/x
19 RCL 1
20 -
21 1
22 e^x
23 LOG
24 *
25 2
26 PI
27 *
28 RCL 1
29 *
```

```
30 LOG
31 2
32 /
33 +
34 RCL 1
35 LOG
36 RCL 1
37 *
38 +
39 INT
40 1
41 +
42 RTN
```

Re: Valentine's Day MC: A few modest insights, n! digits - part 3, anyone?

Message #94 Posted by ***Valentin Albillo*** *on 20 Feb 2007, 4:43 a.m.,*
in response to message #93 by Egan Ford

Hi, Egan:

This 19-step routine for the HP-15C exactly reproduces your results:

```
01  LBL A
02  STO I
03   1
04  EXP
05   /
06  LOG
07  RCL* I
08  RCL I
09  RCL+ I
10  PI
11   *
12  SQR
13  LOG
14   +
15   1
16   +
17  INT
18  X=0?
19   1
```

and further it has no N*12 to restrict range. If the lower limit of the range were 2 instead of 1, the last 2 steps can be omitted for a 17-step solution.

I don't claim that this is the shortest or optimum, as I didn't spend more than 5 minutes with it, but it's still 23 steps shorter than the one you posted. How's that for an improvement :-)

Best regards from V.

Re: Valentine's Day MC: A few modest insights, n! digits - part 3, anyone?

Message #95 Posted by ***Egan Ford*** *on 20 Feb 2007, 10:32 a.m.,*
in response to message #94 by Valentin Albillo

The above was the log of existing gamma code I had for the 12C. Dumping most of the 3rd term as you discovered only affects n = 1.

What's left is the log of Stirling's approximation. Accurate enough for integer work. Nice catch.

Edited: 20 Feb 2007, 10:55 a.m.

Re: Valentine's Day MC: A few modest insights, n! digits - part 3, anyone?

Message #96 Posted by ***Gerson W. Barbosa*** *on 20 Feb 2007, 8:51 p.m.,*
in response to message #94 by Valentin Albillo

> Quote:
>
> If the lower limit of the range were 2 instead of 1, the last 2 steps can be omitted for a 17-step solution.

Hello Valentin,

I would suggest a slight modification in your code. This won't make it shorter, which I believe is almost impossible, but the lower limit will be 0:

```
01  LBL A
02  STO I
03   1
04  TEST 9   (x>=y)
05  RTN
06  EXP
07   /
08  LOG
09  RCL* I
10  RCL I
11  RCL+ I
12  PI
```

```
    13   *
    14  SQR
    15  LOG
    16   +
    17   1
    18   +
    19  INT
```

Best regards,

Gerson.

Re: Valentine's Day MC: A few modest insights, n! digits - part 3, anyone?

Message #97 Posted by ***Valentin Albillo*** *on 20 Feb 2007, 7:59 a.m.,*
in response to message #93 by Egan Ford

Hi again:

I forgot to include the corresponding HP-71B version, which is simply this one-liner which implements a user-defined function:

```
1  DEF FNF(N) @ IF N<2 THEN FNF=1 ELSE FNF=INT(N*LGT(N/EXP(1))+LGT(2*PI*N)/2+1)
```

Let's see it in action against your posted examples:

```
>FNF(1);FNF(13);FNF(69);FNF(1000);FNF(2007);FNF(1234567);FNF(833333333)


      1  10  99  2568  5759  6984221  7072103562
```

As you can see, it perfectly reproduces them all, except the last value is the correct value 7072103562, not 7072103563.

Best regards from V.

Re: Valentine's Day MC: A few modest insights, n! digits - part 3, anyone?

Message #98 Posted by ***Gerson W. Barbosa*** *on 20 Feb 2007, 2:58 p.m.,*
in response to message #93 by Egan Ford

Quote:

Number of digits n! for the 15C.

The digits themselves for the HP-42S:

```
This is a simple adaptation of an HP-41C program in this book (ISBN: 84-267-0506-5), at page 88:
```

FARRANDO BOIX, Ramón: *109 Programas para Ordenadores Personales y Calculadoras*. Ed. Marcombo. Barcelona, 1983.

```
00 { 184-Byte Prgm }
01>LBL "FAC"
02 FIX 00
03 CF 29
04 CF 12
05 CLRG
06 "NUM="
07 PROMPT
08 STO 07
09 1
10 STO 09
11>LBL 02
12 RCL 07
13 1E3
14 /
15 1
16 +
17 STO 05
18>LBL 03
19 RCL 06
20 2
21 -
22 STO 01
23 9
24 STO 06
25>LBL 06
26 RCL IND 06
27 RCL 05
28 IP
29 *
30 RCL 02
31 +
32 STO 03
33 1E10
34 /
35 IP
36 STO 02
37 1E10
38 *
39 +/-
40 RCL 03
41 +
42 STO IND 06
43 RCL 03
44 X=0?
45 GTO 05
46>LBL 04
47 1
48 STO+ 06
49 GTO 06
50>LBL 05
51 RCL 01
52 RCL 06
53 X>Y?
54 GTO 07
55 GTO 04
56>LBL 07
57 ISG 05
58 GTO 03
59 BEEP
60 "FACTORIAL OF "
61 |-""
62 ARCL 07
63 AVIEW
64 ADV
65 SF 12
66 1
67 STO- 06
68 RCL 06
69 STO 03
70 8E-3
71 STO+ 03
72>LBL 00
73 CLA
74 RCL IND 03
75 X=0?
76 GTO 08
77 9
78 RCL IND 03
79 LOG
80 IP
81 -
82 0
83 X=Y?
84 GTO 01
85>LBL 09
86 ARCL ST X
87 DSE ST Y
88 GTO 09
89>LBL 01
90 ARCL IND 03
91 AVIEW
92 DSE 03
93 GTO 00
94 STOP
95>LBL 08
96 "0000000000"
97 AVIEW
98 DSE 03
99 GTO 00
100 END
```

```
        Notes:


     1) Flag 12: double-width printing
     2) Flag 29: digit separator mark
     3) [ALPHA] [ENTER] [ENTER] to enter line 61
     4) Lines 04 and 64 added to the original program
     5) Constants 1E10 (lines 33 and 37) 9 (line 77) replace 1E7 and 6 in the original program
     6) String "0000000000" replaced "0000000"
     7) To check results without a printer, add R/S after line 91





             FACTORIAL OF 69         FACTORIAL OF 100


             0171122452              0093326215
             4281413113              4439441526
             7246833888              8169923885
             1272839092              6266700490
             2705448935              7159682643
             2036939364              8162146859
             8040923257              2963895217
             2797541406              5999932299
             4742400000              1560894146
             0000000000              3976156518
                                     2862536979
                                     2082722375
              (197 sec)              8251185210
                                     9168640000
                                     0000000000
                                     0000000000



                                      (410 sec)
```

I haven't tested the domain on the HP-42S as it takes almost seven minutes for 100!. On Free42 (ARM, 312 MHz) the upper limit is 999 but it takes about 2 minutes for 999! . I do agree the HP-50G is a better option in this case: only 30 seconds for 999! in exact mode! :-)

Gerson.