

HP Forum Archive 10

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

A little HP-15C challenge

Message #1 Posted by *Ex-PPC member* on 16 Jan 2003, 11:20 a.m.

Hi, everybody.

To cheer the forum up a little, here's a nice challenge I've come up with for all you fans of the awesome Voyager Series' HP-15C calculator. Take yours from the safe and use your time-proven ingenuity to try and solve this challenge:

Initial conditions:

1. Assume a newly reset machine (i.e.: no specific angular or display mode, no specific contents in any register or program memory)
2. Start with the value of PI (3.141592654) displayed in the X-register

The Challenge:

You must try and get the value $\text{PI}/10$ (i.e: 0.3141592654) in the X-register, subject to these

Conditions:

1. you cannot use *any* numbered registers (R00 thru R65)
2. you cannot use number-entry keys (0,1,...9, ., EEX, CHS)
3. you cannot use arithmetic operators (+, -, *, /)
4. you must try to *minimize* the byte-count for the operations performed

If you find it too hard, I suggest you first try to solve it subject to a subset of conditions 1,2,3,4 above. For the full challenge (all four conditions simultaneously met), unlikely as it seems, there's a solution in as few as *two bytes* !

The first solution posted will get an autographed jpeg of my 'hero' HP-15C :-)

[By the way, does your solution work for values other than PI, such as e (=2.718281828...) or the square root of 5 ? ;-)]

Wondering about it: I'm stuck.

Message #2 Posted by **Vieira, Luiz C. (Brazil)** on 16 Jan 2003, 3:18 p.m.,
in response to message #1 by Ex-PPC member

I'm stuck.

What makes the HP15C elected for the task instead of the HP11C? Complex Mode, Matrices, SOLVER and Numerical Integration.

Let's verify initial conditions: Pr Error and [f][PI].

Cannot be Matrices, because the simple access to any element demands matrix creation and number entering. But if we create a 1X1 matrix with [f][MATRIX]1... Oops, R0 and R1 are used. SOLVER and Numerical integration demand programming and number entering as either starting roots or upper and lower limits, despite the fact that the program itself is byte consuming.

Activating complex mode uses at least 1 byte ([f][Re\diamondIm] or [f][I]) unless you burn the two bytes with [g][CF][8]... Oops, numbers not allowed.

Based in the fact that there is a two-bytes solution, this leads us to a single one-byte instruction to solve the problem! Well, this is true only if I am correct in my reasoning, otherwise everything else in the HP15C, under the conditions above, is also available in the HP11C.

I want to know how to do it!

Still stuck.

Re: Wondering about it: I'm stuck.

Message #3 Posted by **Katie Wasserman** on 16 Jan 2003, 5:28 p.m.,
in response to message #2 by Vieira, Luiz C. (Brazil)

Here's a 3 byte solution that will work on almost any RPN calculator for any number:

ENTER, %, SQRT

Re: Wondering about it: I'm stuck.

Message #4 Posted by **Dave Shaffer** on 17 Jan 2003, 6:07 p.m.,
in response to message #3 by Katie Wasserman

Why does Katie's solution work?

I don't have a manual handy, but the operation of the % key seems very strange (at least in RPN). As near as I can tell from trying several examples on a '41CX, if you ENTER a number, then enter a second number, and then do "%", you get the product of the two numbers times 0.01. I wouldn't call that a per cent operation. (Given that this is what happens, then taking the square root of the 0.01 factor gives Ex-PPC's desired factor of 0.1) If the second number you enter happens to be 100, then the % operation makes sense - but not otherwise. Am I missing some fundamental math here?!?! I would have expected the % key to give what per cent the X register is of Y (or vice-versa).

Re: Wondering about it: I'm stuck.

Message #5 Posted by **Vieira, Luiz C. (Brazil)** on 17 Jan 2003, 10:07 p.m.,
in response to message #4 by Dave Shaffer

Hi;

if you compute X% of X, you have:

$$\begin{array}{r} x \times x \\ \text{---} \\ 100 \end{array}$$

or

$$\begin{array}{r} x^2 \\ \text{---} \\ 100 \end{array}$$

Think of 10% of 10:

$$\begin{array}{r} 10 \times 10 \\ \text{---} \\ 100 \end{array}$$

or

$$\begin{array}{r} 10^2 \\ \text{---} = 1 \\ 100 \end{array}$$

The square root of X% of X is:

$$x$$

--
10

It took me a while to figure it out, too. A very good demonstration of knowledge from our dear contributor.

Thank you, Katie.

Cheers.

Re: A little HP-15C challenge

Message #6 Posted by **Stephan (Germany)** on 16 Jan 2003, 3:49 p.m.,
in response to message #1 by Ex-PPC member

Long solution:

Starting with PI in x register

$x \diamond y 10^x 10^x g \%$

Re: A little HP-15C challenge

Message #7 Posted by **Gene** on 16 Jan 2003, 4:51 p.m.,
in response to message #6 by Stephan (Germany)

First stop:

Assuming PI is in display.

ENTER SIGMA+ $10^x \%$

four steps, but I haven't given up.

Re: A little HP-15C challenge

Message #8 Posted by **Vieira, Luiz C. (Brazil)** on 16 Jan 2003, 4:56 p.m.,
in response to message #7 by Gene

Hi, Gene;

Ex-PPC asked not to use registers... and [Sigma+] uses them. Just a reminder. Anyway, if he did that, he already knows he can't.

Cheers.

Re: A little HP-15C challenge

Message #9 Posted by **OJM** on 17 Jan 2003, 11:04 a.m.,
in response to message #8 by Vieira, Luiz C. (Brazil)

One answer is:

STO RAN#

RCL RAN#

I think this is unique to the HP15C in how it stores a seed number for the next random number generation. It will do this for any number, not just Pi. For example, 5.235 in the x-register becomes 0.5235.

Re: A little HP-15C challenge

Message #10 Posted by **Ex-PPC member** on 17 Jan 2003, 11:26 a.m.,
in response to message #9 by OJM

Congratulations !! :-)

This is the the **exact** answer to my challenge.

I've always found it somewhat curious that such infrequently used instructions as STO #RANDOM and RCL #RANDOM are **one byte** in length, instead of two-byte. I guess this is because all actual two-byte instructions in the set are of the 'prefix' kind, i.e: STO+, STO-, CF n, SF n, INTEGRATE a, SOLVE b, etc., so there were no candidates to use these two remaining one-byte codes.

By the way, STO #RANDOM and RCL #RANDOM do **not** divide by 10 numbers in the range [0..1), they come out unaltered. So, this internal register can be perfectly used to store constants in that range, thus saving a valuable register (numbered or allocated from the pool) or 7 program bytes, at no cost ! This works fine as long as no random numbers are needed in the routine. I've used this technique in some of my programs for the 15C, and sometimes it was the *only* way to fit the program in the available memory :-)

Your correct entry is the first one which meets all four requirements, while minimizing the byte count, so I will be *very* pleased to send you the 'prize' via e-mail, if only I had some e-mail address of yours ... :-)

Re: A little HP-15C challenge

Message #11 Posted by **OJM** on 17 Jan 2003, 11:51 a.m.,
in response to message #10 by Ex-PPC member

Keep the picture - send me the calc! ;-)

Re: A little HP-15C challenge

Message #12 Posted by **Nenad Vulic (Croatia)** on 18 Jan 2003, 2:19 a.m.,
in response to message #10 by Ex-PPC member

I don't have a solution, but analysing the above proposals (regardless of that weird store/recall random#) I must say:

Katie, you are simply the best!!

IMHO, the solution she has given is the most elegant one...

Hear! Hear!

Message #13 Posted by **Larry Corrado** on 18 Jan 2003, 7:55 a.m.,
in response to message #12 by Nenad Vulic (Croatia)

I agree with Nenad Vulic: Katie's solution was the most elegant.

I tried to do it by subtracting 1 from the log, but couldn't do it within the constraints of the problem.

LC

I agree!

Message #14 Posted by **Vieira, Luiz C. (Brazil)** on 18 Jan 2003, 7:56 a.m.,
in response to message #12 by Nenad Vulic (Croatia)

;-)

(Hi, Nenad!)

Re: I agree!

*Message #15 Posted by **Katie Wasserman** on 18 Jan 2003, 12:54 p.m.,
in response to message #14 by Vieira, Luiz C. (Brazil)*

Thank you all!

That's high praise, coming for the likes for such HP fanatics as yourselves, really!

-Katie

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



Go back to the main exhibit hall