



HP Forum Archive 15

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

New HP-15C mini-challenge !

Message #1 Posted by [Valentin Albillo](#) on 12 Dec 2005, 6:55 a.m.

Hi all,

While waiting for the incoming S&SMC#13, those of you fortunate enough to own an HP-15C (physical or emulated/simulated) might want to check any alleged 15C-*guru* abilities with this [15C-specific](#) mini-challenge:

Write a routine as short and fast as possible to find integer solutions (a,b,c) of:

$$a^2 + b^2 = c^N$$

for any given integer exponent N >= 2

Your routine must assume that the stack has been loaded like this upon calling it:

```
Z: some control value
Y: some control value
X: N
```

where the control values are *arbitrary user-supplied integer values* that your routine must use somehow to generate solutions so that, for different control values, different solutions are usually generated, in a repeatable way. Depending on your algorithm, they might be initial ranges for a search, or "seed" values for some computation or whatever, as long as they ensure variety and provide some control on the solutions produced.

Your routine must find out exactly *one* solution (which should depend on the user-supplied control values and N, of course), and must output the resulting integer values (a, b, c). Of course, you should aim to produce non-trivial solutions, at least for the vast majority of control values even if certain combinations do occasionally result in some trivial solution.

A sample run for N=5 would be something like this:

```
p1 [ENTER] p2 [ENTER]
5 [A] -> 1121
[R/S] -> 404
[R/S] -> 17
```

where p₁ and p₂ are some control values that your algorithm uses to produce the solution (1121, 404, 17). Indeed we have:

$$1121^2 + 404^2 = 1256641 + 163216 = 1419857 = 17^5$$

Specifying different control values, q_1 and q_2 should produce a different solution, say:

$$223191^2 + 119512^2 = 145^5$$

i.e.: $49814222481 + 14283118144 = 64097340625$

Remember that your routine must work for arbitrary integer values of $N \geq 2$, not just 5 (within the HP-15C accuracy and range limitations, of course). In a few days I'll post my own solution, which is a small, HP-15C-specific **12-step routine** (not counting LBL A or RTN). Let's see yours !

Note: Although both this mini-challenge and my solution are HP-15C-specific, solutions for other models are welcome, of course.

Best regards from V.

Re: New HP-15C mini-challenge !

Message #2 Posted by [hugh steers](#) on 13 Dec 2005, 7:05 p.m.,
in response to message #1 by Valentin Albillo

right, as a longtime fan of the 15c. here's my first stab at 19 steps. i am wasting space adjusting loops and stack which is annoying.

```

LBLA
STO 0
DSE 0
RD
I
ENTER
ABS
X^2
R/S
X<>Y
ENTER
ENTER
ENTER
LBL 1
*
CHS
DSE 0
GTO 1
RTN

```

examples

2 ENTER 3 ENTER 2 ENTER LBL A gives $c = 13$ R/S to give 5 and (i) 12. Therefore $13^2 = 5^2 + 12^2$

3 ENTER 4 ENTER 5 ENTER LBL A gives first $c = 25$ R/S to give 1875 and (i) 2500 so therefore $25^5 = 1875^2 + 2500^2$ (claim).

:-)

Re: New HP-15C mini-challenge !

Message #3 Posted by **PeterP** on 14 Dec 2005, 12:31 a.m.,
in response to message #1 by Valentin Albillo

Valentin, you truly have me hooked! Having joined the HP community only very recently I'm right now going through all the historical SSMCs. Just finished #6 (only 5 more to go :-() when I saw this mini-challenges and could not resist. My attempt is not for a HP 15C (don't own one, but appreciated the "hint"...), but for my trusty 41, not very short (20 lines without the Lbl) yet relatively fast as it does the (some?) trick in one run without any loops and works with Size 000.

Entry: n, enter, seed1, enter, seed2. Output: C in x, a in y, b in z. with $C^n = a^2 + b^2$

Here is the listing

```
Lbl ZZ
R-P
Rc1 Z
St/Z
1/x
y^x
P-R
Int
Incx
x<>y
Int
Incx
x<>y
R-P
Sto Z
R/^ (RollUp)
St* Z
y^x
P-R
R/^
x^2
End
```

Edited: 14 Dec 2005, 12:34 a.m.

What is "Incx"? [NT]

Message #4 Posted by **Karl Schneider** on 14 Dec 2005, 1:51 a.m.,
in response to message #3 by PeterP

No Text

X => X + 1 [NT]

Message #5 Posted by [Valentin Albillo](#) on 14 Dec 2005, 4:35 a.m.,
in response to message #4 by Karl Schneider

Best regards from V.

Re: New HP-15C mini-challenge !

Message #6 Posted by [hugh steers](#) on 14 Dec 2005, 6:03 a.m.,
in response to message #3 by PeterP

hi peter,

intriguing! can you explain Z here. what does that do on a 41? thanks.

Re: New HP-15C mini-challenge !

Message #7 Posted by [PeterP](#) on 14 Dec 2005, 9:50 a.m.,
in response to message #6 by hugh steers

Z is just the stack register Z. 41 has no innate complex abilities (if that's what you thought it might be), unless you use Angel's wonderful rom. Here the P-R's and R-P's do the best they can (actually, I can...) to cope for this shortcoming of the 41 vs the 15.

Cheers

Peter

Re: New HP-15C mini-challenge !

Message #8 Posted by [Valentin Albillo](#) on 14 Dec 2005, 6:21 a.m.,
in response to message #3 by PeterP

Hi, PeterP:

PeterP wrote:

"Valentin, you truly have me hooked!"

Why, thank you very much. That's precisely why I do take the trouble to concoct them in the first place.

"I'm right now going through all the historical SSMCs. Just finished #6 (only 5 more to go :-)"

You mean "6 more to go", right ?. Or did you miss "S&SMC#12: Squaring cubes !" ? ... :-)

"when I saw this mini-challenges and could not resist."

There are a lot of them, most for the HP-15C, but I'm not counting.

I'll comment on your solution and all the others within a day or two. Thanks again for your interest and

Best regards from V.

Edited: 14 Dec 2005, 6:23 a.m.

Re: New HP-15C mini-challenge !

Message #9 Posted by **PeterP** on 14 Dec 2005, 9:42 a.m.,
in response to message #8 by Valentin Albillo

... already did #12, even posted a solution i believe...It's what got me into this "lovely mess" in the first place! ;-)

Incoming S&SMC#13 will have a 'prize' ...

Message #10 Posted by **Valentin Albillo** on 14 Dec 2005, 9:48 a.m.,
in response to message #9 by PeterP

... which perhaps you'd find interesting, so if you feel like getting it, get ready and sharpen your HP-calc programming skills *and* mathematical ingenuity, you'll need them both.

Thanks for your kind comments and best regards from V.

Re: New HP-15C mini-challenge !

Message #11 Posted by **Karl Schneider** on 14 Dec 2005, 12:38 a.m.,
in response to message #1 by Valentin Albillo

Valentin --

Interesting little math problem. I have a few observations:

1. Setting Y and Z as the upper and lower limits of a and b will bound the search range of the solution (sort of like SOLVE).
2. All possible values of c could then be determined from Y and Z.
3. Even values of c require that a and b are both even, or both odd.
4. Odd values of c require exactly one odd value of either a or b.
5. If the loop-counter functions ISG or DSE for a and b are used directly, the maximum value of 999 would preclude your first solution.
6. Your second solution requires 11 significant digits for both terms, outside the display range of the HP-15C. I'm sure there's a workaround.

Without some applicable mathematical theorem that would make the problem simple, I don't have a solution to present...

(But, as I was writing this, Peter apparently did!)

-- KS

Edited: 14 Dec 2005, 12:48 a.m.

Re: New HP-15C mini-challenge !

Message #12 Posted by [Eamonn](#) on 14 Dec 2005, 5:51 p.m.,
in response to message #1 by Valentin Albillo

Hi Valentin,

I have to admit that I did not solve the math part of this challenge, so I cheated on this part by looking at the solutions provided by others. After that I understood the basic approach, so I had a go at implementing a program for the HP-15C. Here is a 12 line (not including LBL & RTN) solution for the HP-15C:

```
LBL A
FIX 0
Rv
I
ENTER
ABS
X^2
R/S
X<>Y
R^
Y^X
R/S
Re<>Im
RTN
```

The program displays results using FIX 0 display mode. This is done so that the program always displays an integer. Note that the program can be shortened if the HP-15C is configured for FIX 0 mode before running. The FIX 0 command in the program can be removed, resulting in an 11 step solution.

Additionally, the two R/S commands and the Re<>Im command are included for more user friendly output. If they are omitted, then we get the following eight line program (again, not including the LBL A and RTN lines). The result for 'a' is displayed when the program finishes, 'b' can be displayed by entering Re<>Im and 'c' can be displayed by entering X<>Y.

```
LBL A
Rv
I
ENTER
ABS
X^2
X<>Y
R^
Y^X
RTN
```

Here are some sample results (12 line program)

Test run 1

```
-----
1 ENTER 4 ENTER 5 fA
---> 17 (=c)
R/S
---> 1121 (=b)
R/S
---> 404 (=a)
```

ie. $404^2 + 1121^2 = 17^5$

Test run 2

```
-----
8 ENTER 9 ENTER 5 fA
---> 145 (=c)
R/S
---> -119512 (=b)
R/S
---> -223191 (=a)
```

ie $(-223191)^2 + (-119512)^2 = 145^5$

Test run 3

```
-----
1 ENTER 12 ENTER 5 fA
---> 145 (=c)
R/S
---> 102241 (=b)
R/S
---> 231612 (=a)
```

ie $231612^2 + 102241^2 = 145^5$

Full credit goes to Hugh and Peter for figuring out what needed to be done to solve the challenge, and of course to Valentin for the mini-challenge itself.

Eamonn.

Re: New HP-15C mini-challenge !

*Message #13 Posted by [hugh steers](#) on 14 Dec 2005, 9:10 p.m.,
in response to message #12 by Eamonn*

Most Excellent! doh! i completely missed the y^x idea.

so, read this and weep Valentin :-)) only 10 steps (not counting LBL A + RTN), all answers positive and not even 15c specific.

LBL A
 RD
 ->P
 X^2
 R/S
 SQRT(X)
 R^
 Y^X
 ->R
 R/S
 X<>Y
 RTN

the idea is based on both Eamonn and Peter plus my own original hacks and in the best of humor.

example 1 ent 7 ent 5 lbl a, gives c=50 then r/s gives 17,500 r/s gives 2,500. therefore $50^5 = 17500^2 + 2500^2$.

happy hacking!

Edited: 14 Dec 2005, 9:25 p.m.

Re: New HP-15C mini-challenge !

Message #14 Posted by **PeterP** on 14 Dec 2005, 10:37 p.m.,
 in response to message #13 by hugh steers

You guys are amazing!

I was thinking of the relationship between nth root in complex space and $X^2 + Y^2$. However your tricks are way above me. Would you mind to explain old Foolish me what the math behind your tricks is? maybe for 10 lines a line-by-line analysis would not be too long...

Thanks!

Cheers

Peter

Re: New HP-15C mini-challenge !

Message #15 Posted by **hugh steers** on 15 Dec 2005, 10:18 a.m.,
 in response to message #14 by PeterP

hi peter,

my original idea is based on a plan for generating the solutions by repeated multiplies. the sum of two squares is closed under multiplication;

$(a^2+b^2)(c^2+d^2) = (ac - bd)^2 + (ad + bc)^2$ ie another sum of two squares.

so it occurred to me that, given any starting a & b , let $N = a^2 + b^2$ then we must have $N^2 = (a^2+b^2)(a^2+b^2)$ and it must also be the sum of two squares by the above formula and you can keep going to generate any N^m as required in terms of two squares.

i also noticed that the above is the same way complex numbers multiply.

$$(a+ib)(c+id) = (ac-bd) + i(ad + bc)$$

except that i was getting negative real parts. i wanted positive ones. so i changed the sign each time and carried on multiplying. it was you that made me realise i didn't need to do this and should keep the negative cases – in which case there was no need for a loop to multiply, just use y^x . the other shocker is that the complex number treatment can also be done in polar coordinates. in $(r,\theta)^n = (r^n, n*\theta)$. which means the 15c complex mode is not strictly required.

and then i had a very cheeky idea. my original plan of negating the real part of a complex number in polar terms means $(r,\theta) \rightarrow (r, \pi - \theta)$. so therefore,

```
(1 times) (r, theta)
(2 times) (r^2, 2*theta) -> (r^2, pi - 2*theta)
(3 times) (r^3, pi - 2*theta + theta) = (r^3, pi - theta) -> (r^3, theta)
```

so the theta remains the same for odd powers! so we can get away with $(r,\theta)^n = (r^n, \theta)$ when n is odd, but only for the purposes of retaining two squares – not in general.

However, my cheeky idea was too cheeky because even powers went wrong. Eamonn has applied the fixup for them. However, just doubling theta would also fix it. but i cant see a trick for that.

my complex number (re)attempt came out as; LBL A, RD, I, ENT, R^, Y^X, R/S R<=>I, R/S, RD, ABS, X^2, RTN

which, when i compare it to Eamonn's original, is really the same! so that's not a new answer. i was hoping to somehow squeeze it down a bit more by not using the complex number mode and using instead a R->P trick.

Re: New HP-15C mini-challenge !

Message #16 Posted by **PeterP** on 15 Dec 2005, 11:17 a.m.,
in response to message #15 by hugh steers

Thanks Hugh, that makes sense, very neat indeed!

I was thinking of a thing about the n th power diophant equation $X^2 + Y^2 = Z^n$ I read once. It states something like the integer solution has to be a n th root of $(X+iY)$, lets call it $(a + ib)$, with integer (a, b) . So I "forced" an integer solution by taking the n th root and making the solution integer via "INT" to create a solution $(a + ib)$. $C = a^2 + b^2$, and X and Y are the coordinates of $(a + ib)^n$.

I'm wondering if the complex abilities of the 15c combined with your wizardry would maybe yield a shorter than 12 step solution..

stack loaded with either seed1/seed2/n or n/seed1/seed2 depending on how the 15c needs it loaded for the n th root calc.

n th root -----> yielding (a, b)

```

Int
x<>y
Int      -----> yielding integers (a,b)
nth power -----> yielding X,Y in the real and imaginary parts
LastX C
Abs or Norm -----> yielding C (as a^2 + b^2)

```

(I realize we'll need someway of recouping n after the root operation for the nth power...)

Please accept my apologies if the above is a) wrong b)stupid or c)all of the above. As I stated earlier, you and Eannon (and Valentin) are way above me, just trying to play with the big guys :-)

Cheers

Peter

Re: New HP-15C mini-challenge !

Message #17 Posted by **Eamonn** on 14 Dec 2005, 10:50 p.m.,
in response to message #13 by hugh steers

Hi Hugh,

Based on the comments in your post, I'm sure that you realize that your program does not always return integer solutions for a and b. For example, using an example from an earlier post, with p1=2, p2=3 and N=2, your program returns 13 for c, but 10.8167 for a and 7.2111 for b.

However, this can be fixed with the addition of four more program steps, as shown below. This leads to a 14 line solution that will work on many HPs. Testing with the above inputs returns c=13, a=5 and b=12.

Best Regards,

Eamonn.

```

LBL A
RD
->P
X^2
R/S
SQRT(X)
R^
Y^X
X<>Y  <- Here are the additional program steps
LASTx  <-
*      <-
X<>Y  <-
->R

```

R/S
X<>Y
RTN

Edited: 14 Dec 2005, 10:51 p.m.

Re: New HP-15C mini-challenge !

*Message #18 Posted by [hugh steers](#) on 15 Dec 2005, 9:03 a.m.,
in response to message #17 by Eamonn*

hi eamonn,

yes i knew that some answers were not integers, although i thought it was down to numerical inaccuracy of the machine. non-small input numbers give off-integral results, for example.

looks like its wrong for some small inputs too, which means it can't be called a solution. drat!

thanks for your fixup.

:-)

Edited: 15 Dec 2005, 9:04 a.m.

My original solution and comments

*Message #19 Posted by [Valentin Albillo](#) on 15 Dec 2005, 4:50 p.m.,
in response to message #1 by Valentin Albillo*

Hi all,

Thanks to all of you who posted solutions or comments to my 15C mini-challenge. Hugh Steers gave an early solution employing the correct approach, and then Eamonn almost nailed my original solution. I'm very happy to see the high level of the contributions posted, and I'm sure they enjoyed a lot discovering and implementing the underlying math necessary for the optimum solution.

As for the math itself, Hugh already explained it in detail so no need to repeat it here. As he said, it all depends on the fact that the product of integers which are the sum of two squares is itself the sum of two squares (closure) and there's an exact correspondence with the product of complex numbers. From this, it's trivial to demonstrate that positive integer powers of integers representable as the sum of two squares can themselves be represented likewise, and complex exponentiation will produce the required components at once.

This beats any other approach (such as searching) hands down, and results in an extremely short program, which further produces a solution extremely quickly, even if the computed values of a, b, c are such that squaring and adding them would exceed the 10 digits available.

For the record, my original solution was:

```

LBL A
Rdn
I
ABS
LASTX
Rup
Y^X
R/S
Re<>Im
R/S
CF 8
X<>Y
X^2
RTN

```

which, not counting LBL and RTN, is 12 steps long, as stated in my original post. However, it uses CF 8 to cancel complex mode, which isn't a requirement, only good housekeeping manners. Removing it brings the solution down to 11 steps, and Hugh Steers came up with virtually the same solution after some editing of his first attempt.

A couple of nice solutions:

$$999^2 + 1432^2 = 145^3$$

$$223191^2 + 119512^2 = 145^5$$

That's all. Hope you enjoyed this mini-challenge, in view of the quality of the contributions I certainly did, so thanks to all of you who posted and/or were interested and

Best regards from V.

Note: the incoming S&SMC#13 won't be such a piece of cake but it will be highly (and unexpectedly) astounding from a mathematical point of view, and further it'll include a prize (well, sort of) for those people who succeed in solving it. Stay tuned.

Re: My original solution and comments

Message #20 Posted by **PeterP** on 16 Dec 2005, 1:09 a.m.,
in response to message #19 by Valentin Albillo

Valentin,

thanks for this, I learned a lot. Also from Hugh, Eamonn and the others, many of which I started to appreciate a lot in my ongoing pursuit in solving the old SSMC's (starting at #12 I'm now down to #4...) Unfortunately my knowledge and capabilities in math are not advanced enough - Valentin almost always presents like a wizard a solution of the old "as one can easily see" form. And calling this mini-challenge a piece of cake scared me already for SSMC13, I might have to sit that one out...

Anyway, inspired by all the brain here, I loaded the fabulous 41z mod from Angel Martin onto my 41 (I think you Valentin also contributed there a program, right?) and tried a modified version of my original solution.

If one requires integer starting points, not too large, it appears that the problem is solvable in no less than 6(!) steps (not counting the lbl and rtn) using the theorem I mentioned above. The solution below works fine on my 41 and should work with very little translation on a 15c as well.

Limitations: a) for even moderately high n it quickly produces large solutions, potentially running into rounding errors. b) it almost always produces negative a, b which still fulfill the initial request, yet are a bit less pretty.

I'm sure you guys can find more limitation, yet I thought I throw it in the mix, given its "length".

Input : n enter, intseed1 enter, intseed2 enter

Output: C, R/S then yields a in x, b in y (one could display them together as well with ZView, yet I find it more convenient to have the solution in x and y)

```
Lbl aa
Znorm
R/S -----> that is C
RDN
LastZ
RollUp
Z^x
Rtn -----> a in x-reg, b in y-reg
```

Cheers

Peter

PS: a final, but huge, thanks goes to Dave for letting us have so much fun through his forum!

Edited: 16 Dec 2005, 1:12 a.m.

Re: My original solution and comments

Message #21 Posted by [Ángel Martín](#) on 16 Dec 2005, 4:44 a.m.,
in response to message #20 by PeterP

Thanks Peter for the praise. Glad to see that the 41Z is put to work in here! (still one of my favorite modules if not the most).

Regards, ÁM

BTW this alternative listing saves you one step, albeit it's the same size in bytes:

```
Lbl aa
ZNORM
R/S -----> that is C
LASTZ
RCL Z
```

Z^X

Rtn -----> a in x-reg, b in y-reg

Edited: 16 Dec 2005, 4:54 a.m.

Re: Wow, down to 5 Steps! [NT]

*Message #22 Posted by **PeterP** on 16 Dec 2005, 8:36 p.m.,
in response to message #21 by Ángel Martin*

Cheers

Peter

PS (I use all of your fabulous modules a lot...)

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



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