

Problem 3: HMMM Power!

 courses.edx.org/courses/course-v1:HarveyMuddX+CS005x+2T2016/courseware/76469178cd4f467c9527a3fe617e3762/96ada6e8

For this problem, you'll create a new HMMM program!

Your program for this problem should first ask for **two** nonnegative numbers from the user. Then, the program should compute the result obtained when you raise the first input to the power of the second input. Finally, it should print that result and then `halt`.

Use the factorial example we went over in class as a guide to how this power program could work. Here is that factorial example from that first HMMM lecture:

```
00 read r1          # get # from user to r1
01 setn r2 1        # put our result into r2
02 jeqzn r1 07      # jump to line 7 if r1 == 0
03 mul r2 r2 r1     # make r2 = r2 * r1
04 addn r1 -1       # make r1 = r1-1
05 jumpn 02         # jump back to line 2
06 nop
07 write r2         # write out the result, r2
08 halt
```

This is not a solution—remember that you'll need to edit this code!

Hints!

- Use the existing `read` statement to input the base.
- Add another `read` statement to input the power, say, into `r2`.
- *Keep 1 as your initial result value (aka "base case")!* Perhaps use `r3`.
- Then **test** to see if you're finished. You're probably finished when the power is 0!
- If you're not finished, you need to multiply once, reduce the remaining powers, and loop!

For this problem, you may assume that both inputs `n` will always be at least 0. 0 to the 0 power should result in 1. Here are a couple runs' worth of sample input and output:

```
Enter number: 2
Enter number: 5
32
```

```
Enter number: 42
Enter number: 1
42
```

```
Enter number: 42
Enter number: 0
1
```

```
Enter number: 0
Enter number: 0
```

Remember—you should have a comment of one line or more for every line of code that you write in order to explain what it is doing. Also, test your program carefully, including the "edge cases" when one or both inputs are zero.