Mario

Implement a program that prints out a double half-pyramid of a specified height, per the below.

```
$ ./mario
Height: 4
   #  #
  ##  ##
 ###  ###
####  ####
```

Specification

- Write, in a file called `mario.py` in `~/pset6/mario/more/`, a program that recreates these half-pyramids using hashes (`#`) for blocks, exactly as you did in Problem Set 1, except that your program this time should be written in Python.
- To make things more interesting, first prompt the user with `get_int` for the half-pyramid's height, a positive integer between `1` and `8`, inclusive. (The height of the half-pyramids pictured above happens to be `4`, the width of each half-pyramid `4`, with a gap of size `2` separating them).
- If the user fails to provide a positive integer no greater than `8`, you should re-prompt for the same again.
- Then, generate (with the help of `print` and one or more loops) the desired half-pyramids.
- Take care to align the bottom-left corner of your pyramid with the left-hand edge of your terminal window, and ensure that there are two spaces between the two pyramids, and that there are no additional spaces after the last set of hashes on each row.

Usage

Your program should behave per the example below.

```
$ ./mario
Height: 4
   #  #
  ##  ##
 ###  ###
####  ####
```

Testing

While `check50` is available for this problem, you're encouraged to first test your code on your own for each of the following.

- Run your program as `python mario.py` and wait for a prompt for input. Type in `-1` and press enter. Your program should reject this input as invalid, as by re-prompting the user to type in another number.
- Run your program as `python mario.py` and wait for a prompt for input. Type in `0` and press enter. Your program should reject this input as invalid, as by re-prompting the user to type in another number.
- Run your program as `python mario.py` and wait for a prompt for input. Type in `1` and press enter. Your program should generate the below output. Be sure that the pyramid is aligned to the bottom-left corner of your terminal, and that there are no extra spaces at the end of each line.

```
#  #
```

- Run your program as `python mario.py` and wait for a prompt for input. Type in `2` and press enter. Your program should generate the below output. Be sure that the pyramid is aligned to the bottom-left corner of your terminal, and that there are no extra spaces at the end of each line.

```
 #  #
##  ##
```

- Run your program as `python mario.py` and wait for a prompt for input. Type in `8` and press enter. Your program should generate the below output. Be sure that the pyramid is aligned to the bottom-left corner of your terminal, and that there are no extra spaces at the end of each line.

```
       #  #
      ##  ##
     ###  ###
    ####  ####
   #####  #####
  ######  ######
 #######  #######
########  ########
```

- Run your program as `python mario.py` and wait for a prompt for input. Type in `9` and press enter. Your program should reject this input as invalid, as by re-prompting the user to type in another number. Then, type in `2` and press enter. Your program should generate the below output. Be sure that the pyramid is aligned to the bottom-left corner of your terminal, and that there are no extra spaces at the end of each line.

```
 #  #
##  ##
```

- Run your program as `python mario.py` and wait for a prompt for input. Type in `foo` and press enter. Your program should reject this input as invalid, as by re-prompting the user to type in another number.
- Run your program as `python mario.py` and wait for a prompt for input. Do not type anything, and press enter. Your program should reject this input as invalid, as by re-prompting the user to type in another number.

Execute the below to evaluate the correctness of your code using `check50`. But be sure to compile and test it yourself as well!

```
check50 cs50/problems/2020/fall/sentimental/mario/more
```

Execute the below to evaluate the style of your code using `style50`.

```
style50 mario.py
```

This problem will be graded only along the axes of correctness and style.

How to Submit

1. Download your `mario.py` file by control-clicking or right-clicking on the file in CS50 IDE's file browser and choosing **Download**.
2. Go to CS50's Gradescope page (https://www.gradescope.com/courses/157004).
3. Click "Problem Set 6: Sentimental (Mario More)".
4. Drag and drop your `mario.py` file to the area that says "Drag & Drop". Be sure it has the correct filename!
5. Click "Upload".

You should see a message that says "Problem Set 6: Sentimental (Mario More) submitted successfully!" You won't see a score just yet, but if you see the message then we've received your submission!