Preamble A narcissistic number (see https://en.wikipedia.org/wiki/Narcissistic_number page) is a number that is the sum of its own digits each raised to the power of the number of digits (e.g. 153 = 1^3 + 5^3+3^3). Please note that we have not explicitly discussed the use of the floor() and ceil() functions which you would need for this assignment.

Instructions

- Write a function called **narcissistic_number** that finds all the "narcissistic" numbers between **A** and **B**. Hint: Use division (and the **mod** command) and the **floor** command (use the help command to find out what it does) to access individual digits of your integer.
- The inputs to your function are **A** and **B**, where

(you can assume that the user will call the function properly – making sure that the inputs are set to satisfy this constraint).

- Your function should count the total number of narcissistic numbers you found, then sum them all up. Then your function should assign these results to the variables **narci_count** and **narci_sum** respectively.
- Your function should return **narci_count** as the first output, and return **narci_sum** as the second output. Return both variables as zero if you don't find any narcissistic numbers in the given range.

Write a function named **my_diamonds** with three inputs: **shape1** as the first kind of character in a pattern, **shape2** as the second kind of character in a pattern, and lastly, **height** as the total number of lines in the pattern (which is assumed to be an even number). Make sure your function can generate the pattern in the following example. Example: when **shape1** = '_' , **shape2** = '$', and **height** = 8, the pattern is:

```
___$$___
__$$$$__
_$$$$$$_
$$$$$$$$
$$$$$$$$
_$$$$$$_
__$$$$__
___$$___
```

Print the pattern like shown below ***using loops***. Do not worry about the digits, and do not try to align the numbers well as long as the pattern of the numbers is correct.

```
1
21
321
4321
54321
654321
7654321
87654321
```

The galaxy is at peace and the Guardians finally have some time for themselves. In a constant game of "Who is smarter?", Star-Lord is perennially trying to one up his raccoon teammate at programming challenges. In this process, he comes up with an idea to display various patterns on a screen using a **for loop**. However, he can't finish it and needs your help fixing up the code! He is supposed to generate this pattern, which is the combination of two right triangles:

```
**********
_********_
__******__
___****___
____**____
```

However, his current code only generates these two right triangles:

```
*
**
***
****
*****
```

and

```
____*
___**
__***
_****
*****
```

He doesn't know how to 'flip' his triangles. He has given us his code in the solution template. Click 'test' to see what the result looks like.

Please edit this code to:

- flip the first right triangle pattern and make it look like this

```
*****
****
***
**
*
```

- flip the second pattern and make it look like this

```
*****
_****
__***
___**
____*
```

- After you complete the two triangles above, write some code to combine them and generate the upside-down triangle pattern that Star-Lord was supposed to make.