Homework #4

Submission instructions:

1. You should submit your homework in the NYU Classes system.
2. For this assignment, you should turn in 8 '.py' files. Name your files:

    'YourNetID_hw4_q1.py', 'YourNetID_hw4_q2.py', 'YourNetID_hw4_q3.py',

    'YourNetID_hw4_q4.py', 'YourNetID_hw4_q5.py', 'YourNetID_hw4_q6a.py',

    'YourNetID_hw4_q6b.py', 'YourNetID_hw4_q7.py'.

Question 1

Write **two versions** of a program that reads a positive integer `n`, and prints the first `n` even numbers.

   a) In the first program, use a `while` loop.
   b) In the second program, use a `for` loop.

For example, one execution would look like this:
Please enter a positive integer: 3
2
4
6

Question 2
Ask the user to input a positive integer n, and print a textual image of an hourglass made of $2n$ lines with asterisks.
For example, if n=4, the program should print:

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

Question 3:
Print out power table with 5 rows and 10 columns. Value of the power table at row i, column j is j^i.
The columns should be spaced by a tab.
Expected output:

```
>>> ============================== RESTART ==============================
>>>
1    2    3    4    5    6    7    8    9    10

1    4    9    16   25   36   49   64   81   100

1    8    27   64   125  216  343  512  729  1000

1    16   81   256  625  1296 2401 4096 6561 10000

1    32   243  1024 3125 7776 16807 32768 59049 100000

>>>
```

Question 4:
Ask user to input a positive integer n, and print a triangle of numbers aligned to the right, where the first line contains the number 1. The second line contains the numbers 1,2. The third line contains 1,2,3. And so on.
For example if n=5, the program should print:

```
    1
   12
  123
 1234
12345
```

Question 5:
Ask user to input a positive integer n, and print all of the numbers from 1 to n that have more even digits than odd digits.

For example, if n=30, the program should print:

2
4
6
8
20
22
24
26
28

Question 6:

Write two versions of a program that **reads a sequence of positive integers from the user**, calculates their **geometric mean**, and print the geometric mean.

Note: In mathematics, geometric mean of a dataset $\{a_1, a_2, a_3 \dots, a_n\}$ is given by: $\sqrt[n]{a_1 \cdot a_2 \cdot a_3 \cdots a_n}$.

For example, the geometric mean of 2, 9 and 12 is equal to 6 ($\sqrt[3]{2 \cdot 9 \cdot 12} = 6$).

Your two versions of the program should read the integer sequence in two ways:

a) First read the length of the sequence

For example, an execution would look like:

Please enter the length of the sequence: 3
Please enter your sequence:
1
2
3
The geometric mean is: 1.8171

b) Keep reading the numbers until 'done' is entered.

For example, an execution would look like:

Please enter a non-empty sequence of positive integers, each one in a separate line. End your sequence by typing done:
1
2
3
done
The geometric mean is: 1.8171

Question 7:
Implement a number guessing game. The program should randomly choose an integer between 1 and 100 (inclusive), and have the user try to guess that number.

Implementations guidelines:
1. The user can guess at most 5 times.
2. Before each guess the program announces:
   - An updated guessing-range, taking in to account previous guesses and responses.
   - The number of guesses that the user has left.
3. If the user guessed correctly, the program should announce that, and also tell how many guesses the user used.
4. If the user guessed wrong, and there are still guesses left, the program should tell the user if the number (it chose) is bigger or smaller than the number that the user guessed.
5. If the user didn't guess the number in all of the 5 tries, the program should reveal the number it chose.
6. Follow the execution examples below for the exact format.

Execution example 1:
I thought of a number between 1 and 100! Try to guess it.

Range: [1, 100], Number of guesses left: 5
Your guess: 15
Wrong! My number is bigger.

Range: [16, 100], Number of guesses left: 4
Your guess: 34
Wrong! My number is smaller.

Range: [16, 33], Number of guesses left: 3
Your guess: 23
Congrats! You guessed my number in 3 guesses.

Execution example 2:
I thought of a number between 1 and 100! Try to guess it.

Range: [1, 100], Number of guesses left: 5
Your guess: 15
Wrong! My number is bigger.

Range: [16, 100], Number of guesses left: 4
Your guess: 50
Wrong! My number is smaller.

Range: [16, 49], Number of guesses left: 3
Your guess: 3
Wrong! My number is bigger.

Range: [16, 49], Number of guesses left: 2
Your guess: 34
Wrong! My number is smaller.

Range: [16, 33], Number of guesses left: 1
Your guess: 20
Out of guesses! My number is 25