CMPSC-132: Programming and Computation II
Fall 2018

Lab #7

Due Date: 10/05/2018, 11:59PM

Instructions:

- The work in this lab must be completed alone and must be your own. Do not copy code from online sources. That is considered plagiarism.
- Use the starter code provided on this CANVAS assignment. Do not change the function names or given started code on your script
- The file name must be LAB7.py (incorrect name files will get a -1 point deduction)
- **A doctest is provided as an example of code functionality. Getting the same result as the doctest does not guarantee full credit. You are responsible for testing your code with enough data as possible.**
- Each function must return the output (Do not use print in your final submission otherwise, you will get a -1 pt deduction)
- Do not include test code outside any function in the upload. Remove all your testing code before uploading your file. Do not include the input() function in your submission.

Goal

[10 pts] In the starter code, there is a function called *triangle* that calls the function *recursive_triangle*(*n*,*n*) once. For this exercise you must write the recursive function *recursive_triangle*(*x*, *n*) that **returns** a string with the LAST *x* lines of a right triangle of base and height *n*.

- *recursive_triangle*(*x*, *n*) must be a recursive function, otherwise, no credit is given
- *recursive_triangle*(*x*, *n*) must return the pattern as ' ***\n   **\n   *'. As shown in the example, you can use the print method during testing to check if your pattern is correct.
- Don't modify anything in the triangle function. If your recursive function is correct, calling *triangle*(*n*) should return the complete right triangle.

EXAMPLES:

```
>>> triangle(4)
'****\n ***\n  **\n   *'
>>> print(triangle(4))
****
 ***
  **
   *

>>> recursive_triangle(2,4)
'  **\n   *'
>>> print(recursive_triangle(2,4))
  **
   *
```

This is the output that will be graded, make sure your function returns the value in this form

You can call the print method on your function to see the triangle pattern

```
>>> triangle(6)
'******\n *****\n  ****\n   ***\n    **\n     *'
>>> print(triangle(6))
******
 *****
  ****
   ***
    **
     *
>>> recursive_triangle(3,6)
'   ***\n    **\n     *'

>>> print(recursive_triangle(3,6))
   ***
    **
     *
```

Notes:

- Output like '   ***\n   **\n   *\n' is also valid
- The doctest includes \\n instead of \n in order to keep consistent leading whitespace

Deliverables:

- Submit your code in a file name LAB7.py to the Lab7 CANVAS assignment before the due date