CSC 242: Introduction to Computer Science II

Lab exercise set 7

See D2L for due date

Logistics

In order to receive full credit for the lab, you must attend the session, remain in the lab for at least 45 minutes, and submit a file that contains solutions to all these exercises.

You are encouraged to work in groups on lab exercises. If you do work with someone, you must include the name(s) of your collaborator(s) at the top of the file you submit. For more information about collaboration policies in this class, see the Academic Integrity policy posted on the web site.

If you complete the lab exercise early, please read Chapter 10 in the textbook and work on the 6th assignment. You are only allowed to work on assignments with at most two-other people, either directly or indirectly, who must be formally identified as part of your assignment submission. Please see the assignment description and the course Academic Integrity policy for the full description of the process you must follow when completing assignments. If you need additional help on the assignment, please ask the teaching assistant.

Exercises

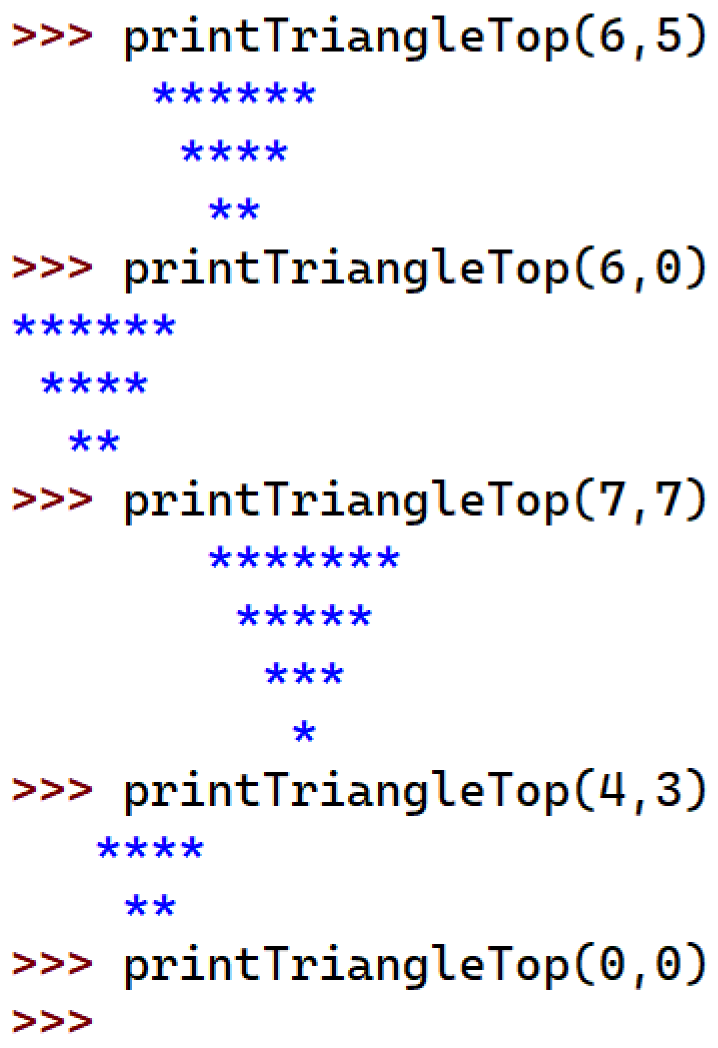
Please remember that you are not allowed to consult online resources when completing lab exercises. If you have questions, please either ask the teaching assistant or contact me by email.

Begin the lab by downloading the template file (**csc242lab7.py**) from the D2L site. It contains the headers for the functions you will write. Do not modify the function names or parameters in the template file. The functions written for this assignment must be recursive and **must not** use global variables. You are also **not allowed** to use loops. In some cases certain built-in functions are restricted in your solutions so carefully read each question. Solutions that don't follow these guidelines will not earn full credit, even if they produce the correct results in all cases.

1. Take the code snippet below and paste it into Python Tutor (A copy is in the lab file). Analyze its behavior and discuss how it works with a lab partner. Pay careful attention to the call stack as shown in Python Tutor. http://www.pythontutor.com

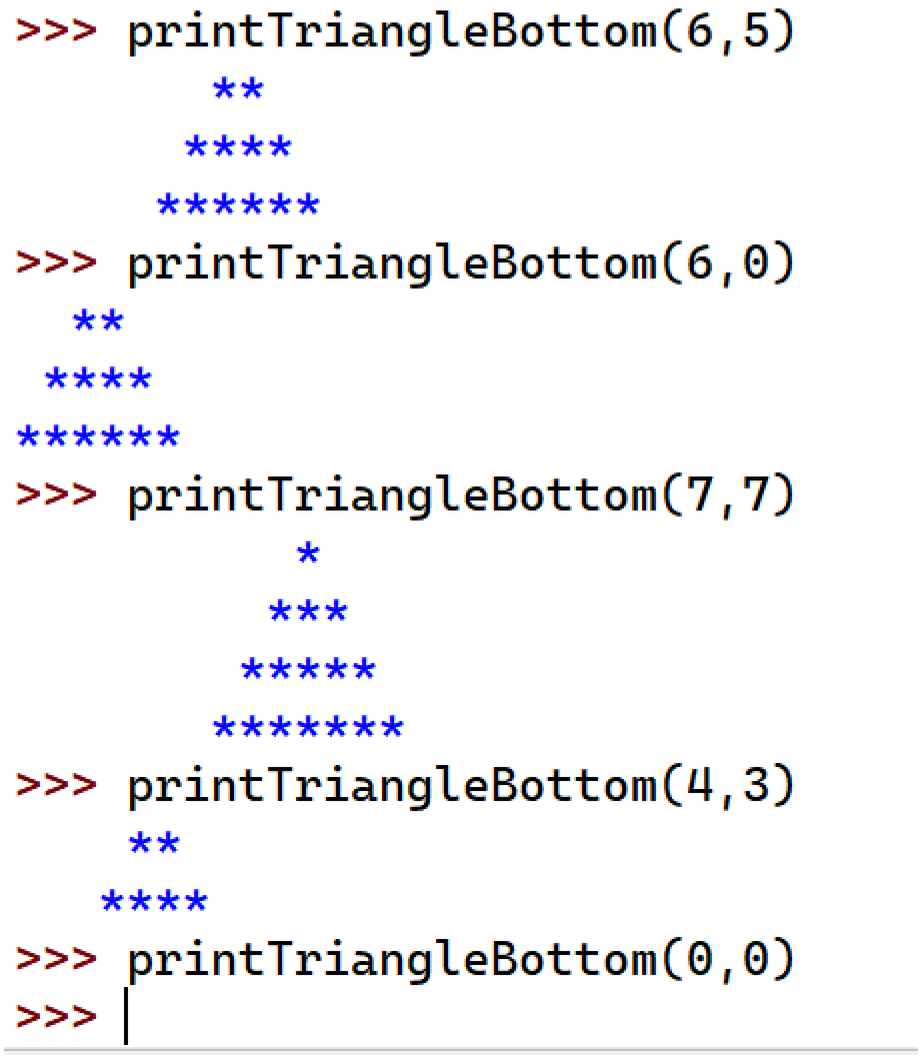
```
def printTriangleManually3():
  print((' ' * 2) + ('*' * 2))

def printTriangleManually2():
  print((' ' * 1) + ('*' * 4))
  printTriangleManually3()

def printTriangleManually():
  print((' ' * 0) + ('*' * 6))
  printTriangleManually2()

printTriangleManually()
```

2. Once you understand the snippet in part 1, you will take the concept and apply it in a more flexible manner using recursion. Write a recursive function **printTriangleTop**() that takes two integers as parameters and **prints** a triangle of asterisks based on those parameters. The first integer represents the maximum number of asterisks to be printed in the first row of the triangle pattern seen in the examples below. The second integer represents the indentation used for the first line of the pattern. If the first parameter is negative or zero, nothing will be printed. You are allowed to assume that the second parameter will be non-negative. You are allowed to use string multiplication (e.g. c * n for a character c and an integer n) and concatenations, but no other string functions are allowed. The following shows several sample runs of the function:



```
>>> printTriangleTop(6,5)
     ******
      ****
       **
>>> printTriangleTop(6,0)
******
 ****
  **
>>> printTriangleTop(7,7)
       *******
        *****
         ***
          *
>>> printTriangleTop(4,3)
   ****
    **
>>> printTriangleTop(0,0)
>>>
```

3. Write a recursive function **printTriangleBottom**() that takes two integers as parameters and **prints** a triangle of asterisks based on those parameters. The first integer represents the maximum number of asterisks to be printed in the bottom row of the triangle pattern seen in the examples below. The second integer represents the indentation used for the first line of the pattern. If the first parameter is negative or zero, nothing will be printed. You are allowed to assume that the second parameter will be non-negative. You are allowed to use string multiplication (e.g. c * n for a character c and an integer n) and concatenations, but no other string functions are allowed. The following shows several sample runs of the function:



```
>>> printTriangleBottom(6,5)
       **
      ****
     ******
>>> printTriangleBottom(6,0)
  **
 ****
******
>>> printTriangleBottom(7,7)
          *
         ***
        *****
       *******
>>> printTriangleBottom(4,3)
    **
   ****
>>> printTriangleBottom(0,0)
>>> 
```

Submitting the exercises

You must submit your solutions to the exercises using the lab 7 dropbox on the D2L site. Submit only a single Python file (e.g. **csc242lab7.py**) with each of the completed functions and classes for the lab exercises in it. Submissions after the deadline listed above will be automatically rejected by the system. See the syllabus for the grading policy.

Grading

The lab session is worth 10 points. If you complete the lab exercises before the end of the lab session, please work on the fifth assignment. Remember that the rules for collaboration on assignments is different from labs. Please review the Academic Integrity policy for more information. If you have questions about the assignment, please ask the teaching assistant for help.