CSC 243: Python for Programmers

Homework 8

Reading

Make sure you have read Chapter 10 in **Introduction to Computing using Python: An Application Development Focus, Second Edition** by Ljubomir Perković. When we finish with recursion, we will use whatever time we have left to cover inheritance (Sections 8.5 and 8.6). Make sure you have read those as well.

Logistics

If you want to do the assignment on your own computer, you will need to download and install Python. There is a link for downloading Python under Content -> Course Information on D2L.

In this class programming assignments may be completed in consultation with up to two other classmates. You must identify the classmates with whom you collaborate in the comment box when you submit on D2L. You must also list their names at the top of the assignment. The total number of collaborators on any assignment **may not exceed two other people**. Please see the Collaboration Guidelines document found under Content -> Course Information on D2L for more details on what is or is not allowed when working on homework.

Remember that **everyone** submitting the assignment must be able to explain **all** of the code submitted, regardless of the type of collaboration that occurred. Anyone submitting code they cannot explain is violating the Academic Integrity policy and will earn a 0 on the assignment.

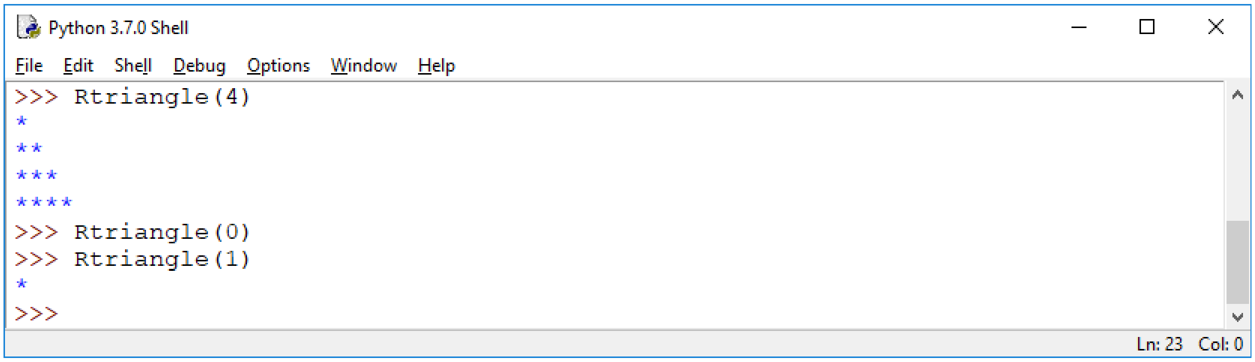
RECURSION REQUIREMENT

All functions must be implemented using recursion. You will not receive any credit for solutions that are not recursive. You may not use any kind of loop in your solution.

Assignment

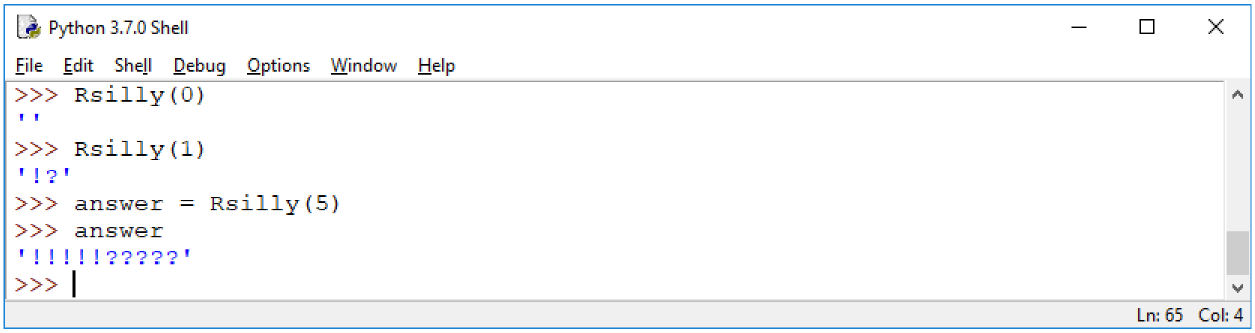
To begin the assignment, download the **HW8.py** template found on the D2L site. You will need to complete the implementation as described below.

1. Write a recursive function called **Rtraingle** that takes a single non-negative integer as an argument and displays to the screen a triangle made up of asterisks as illustrated below. (Remember that you can use the '*' operator on strings and numbers.)



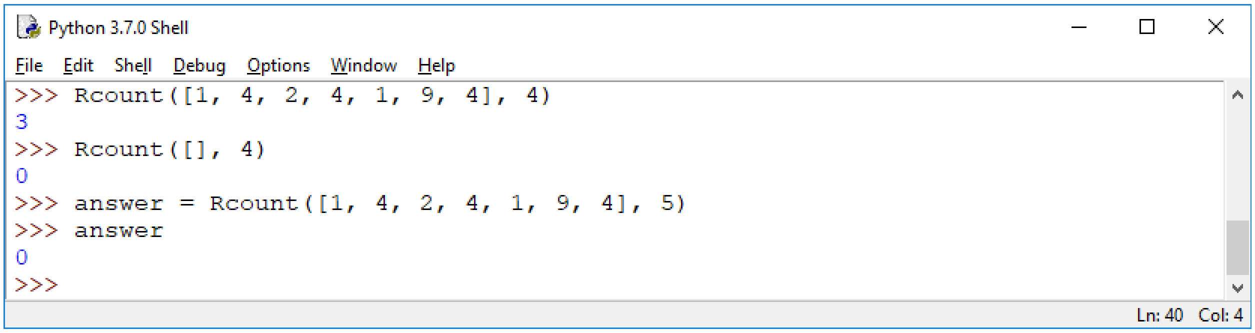
```
>>> Rtriangle(4)
*
**
***
****
>>> Rtriangle(0)
>>> Rtriangle(1)
*
>>>
```

2. Write a recursive function called **Rsilly** that that takes a single non-negative integer n as an argument and returns a string that contains n exclamation points (!) followed by n question marks (?). **IMPORTANT:** You may not use the * operator on strings. You may only use the + operator.



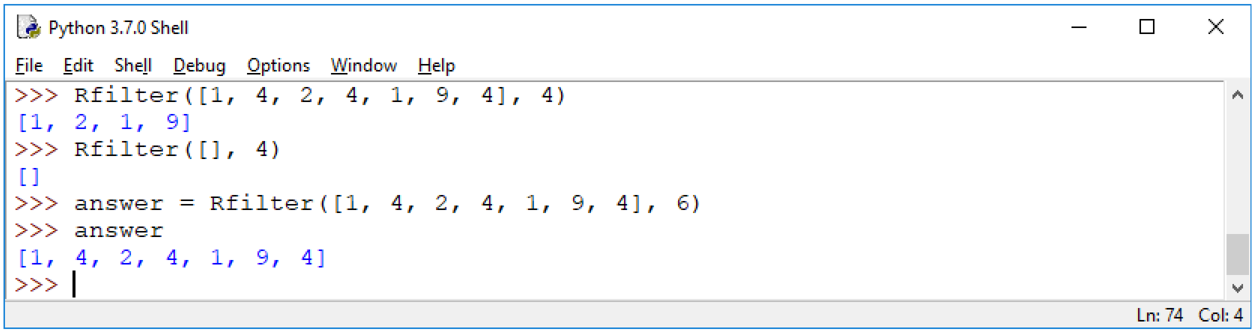
```
>>> Rsilly(0)
''
>>> Rsilly(1)
'!?'
>>> answer = Rsilly(5)
>>> answer
'!!!!!?????'
>>> 
```

3. Write a recursive function called **Rcount** that takes a list of numbers and a number as parameters and returns how many times the number appears in the list. Below are a couple of examples of how it works.
IMPORTANT: You may not use the built in count function.



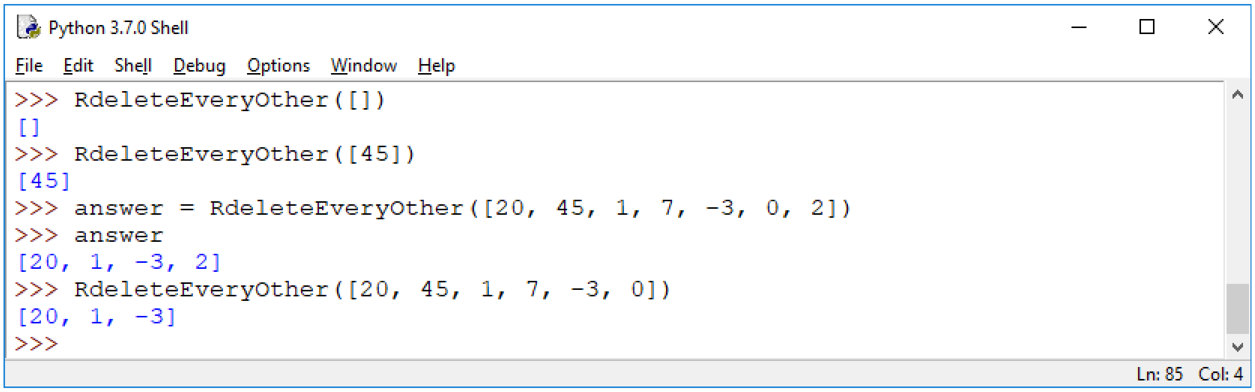
```
>>> Rcount([1, 4, 2, 4, 1, 9, 4], 4)
3
>>> Rcount([], 4)
0
>>> answer = Rcount([1, 4, 2, 4, 1, 9, 4], 5)
>>> answer
0
>>>
```

4. Write a recursive function called **Rfilter** that takes a list of numbers and a number as parameters and returns a new list that is identical to the old list but with all occurrences of the given number removed. The function must not modify the original list in any way. You may not use the built in functions remove or pop, nor the del operator. Below are a couple of examples of how it works.



```
>>> Rfilter([1, 4, 2, 4, 1, 9, 4], 4)
[1, 2, 1, 9]
>>> Rfilter([], 4)
[]
>>> answer = Rfilter([1, 4, 2, 4, 1, 9, 4], 6)
>>> answer
[1, 4, 2, 4, 1, 9, 4]
>>> 
```

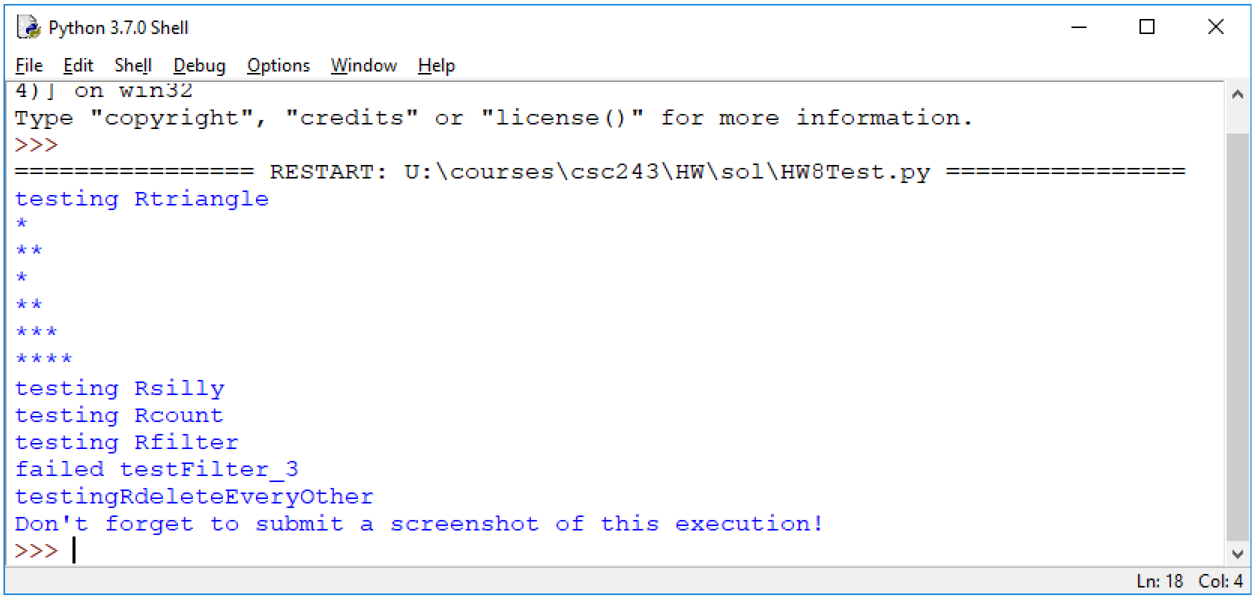
5. Write a recursive method called **RdeleteEveryOther** that takes a list as an argument and returns a new list that contains every other element from the original list, beginning with the first item in the original list. In other words, the new list looks like a copy of the original list with all the elements in the odd numbered positions removed. (Remember that the numbering begins with 0.)



```
>>> RdeleteEveryOther([])
[]
>>> RdeleteEveryOther([45])
[45]
>>> answer = RdeleteEveryOther([20, 45, 1, 7, -3, 0, 2])
>>> answer
[20, 1, -3, 2]
>>> RdeleteEveryOther([20, 45, 1, 7, -3, 0])
[20, 1, -3]
>>>
```

Testing

Included in the dropbox is a file called HW8Tests. You are required to run that test file on your HW8.py file, and submit a screenshot of the result. Remember that to run HW8Test.py it must be in the same folder as HW8.py. Open HW8Test.py in python and then hit F5 or select Run Module from the Run menu. A successful execution of HW8Test.py should look like:



```
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
================ RESTART: U:\courses\csc243\HW\sol\HW8Test.py ================
testing Rtriangle
*
**
*
**
***
****
testing Rsilly
testing Rcount
testing Rfilter
failed testFilter_3
testingRdeleteEveryOther
Don't forget to submit a screenshot of this execution!
>>> 
```

Submitting the assignment

You must submit the assignment using the HW8 folder of the dropbox on D2L. Submit your HW8.py file as well as a screenshot of running HW8Test.py. Failure to submit the screenshot will result in a 10 point deduction to your score. Submissions after the deadline will be automatically rejected by the system and will receive a grade of 0.

Grading

Each of the five problems is worth 20 points for a total of 100 points.