CSC 242: Introduction to Computer Science II

Lab exercise set 8

Due Wednesday, May 20th, 11:59 pm

Logistics

These exercises should be completed during the lab 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 to the D2L site.

If you complete the lab exercise early, please review Chapter 10 in the textbook and work on the eighth assignment. You must remain in the lab for at least 45 minutes to earn full credit. 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 pledge 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 (**csc242lab8.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. Write a **recursive** function **extractStr**() that takes an arbitrarily nested list as a parameter and **returns** a string that consists of the concatenation of all strings found in the list. Note that the list may contain any Python type, not just strings. Recall that you can determine whether an item is a string by writing type(item) == str and whether an item is a list by writing type(item) == list. The function should not use any loops. The only list functions you are allowed to use are len(), indexing (lst[i] for an integer i), or slicing (lst[i:j] for integers i and j). You should use string concatenation in writing your solution. If the list does not contain any strings or is empty the function should return the empty string. The following shows several sample runs of the completed function:

```
>>> extractStr(['a','b','c','d','e'])
'abcde'
>>> extractStr([1.1,'a',2,'b',[],'c'])
'abc'
>>> extractStr([])
''
>>> extractStr([[[[[['1']]]]],['2'],[['3'],'4'],'5'])
'12345'
>>>
```

1. Write a **recursive** function **totalNumericValue** () that takes an arbitrarily nested list as a parameter and **returns** a number that consists of the total of all the numbers in the nested list. Note that the list may contain any Python type, not just strings. Recall that you can determine whether an item is a int by writing type(item) == int and whether an item is a list by writing type(item) == list. The function should not use any loops. The only list functions you are allowed to use are len(), indexing (lst[i] for an integer i), or slicing (lst[i:j] for integers i and j). If the list does not contain any numbers or is empty the function should 0. The following shows several sample runs of the completed function:

```
>>> lst=[1,2,[3,4,[5,6]],7,8,[[[[[9]]]]]]
>>> totalNumericValue(lst)
45
>>> totalNumericValue([])
0
>>> totalNumericValue([1,2,3,4])
10
```

Submitting the exercises

You must submit your solutions to the exercises using the lab 8 dropbox on the D2L site. Submit only a single Python file (e.g. **csc242lab8.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 eighth assignment. Remember that the rules for collaboration on assignments is different from labs. Please review the Academic Integrity pledge for more information. If you have questions about the assignment, please ask the teaching for help.