

CS 1110, LAB 09: RECURSION

<http://www.cs.cornell.edu/courses/cs1110/2017sp/labs/lab09.pdf>

First Name: _____ Last Name: _____ NetID: _____

Getting Credit: Deadline: the first 10 minutes of (your) lab **two weeks from now (Tue Apr 11 or Wed Apr 12)**, due to Spring Break.

The checking-off procedure is the same as before.¹

As usual, create a new directory on your hard drive for this lab's files. Then, download into that new directory the files you need for lab 09; get them packaged in a single zip file from the Labs section of the course web page, <http://www.cs.cornell.edu/courses/cs1110/2017sp/labs> .

Read the function specifications in `lab09.py`, and look at the test cases for each such function given in `lab09_test.py` to make sure you understand what each function is supposed to do.

Note that the third and optional fourth function are the most “naturally” recursive², whereas the first two are just as amenable to for-loops as to recursion. However, we're still asking you to write recursive solutions for the first two so that you get practice with the “mechanics” of recursion.

Finish the implementations of the **the first three** functions in `lab09for.py`. *You must use recursion in each one.* You can optionally also complete the fourth function, for more practice.

You may test your code using `lab09_test.py`. If the test cases there don't look complete, feel free to add your own.

Course authors: E. Andersen, D. Gries, L. Lee, S. Marschner, C. Van Loan, W. White

¹ In case you've forgotten, here's a reminder: Show this handout and/or your code to a staff member either (a) during your lab 09 session, (b) in consulting hours listed at <http://www.cs.cornell.edu/courses/cs1110/2017sp/about/staff.php> up to the day **before** your next scheduled lab section, or (c) in the first 10 minutes of (your) next scheduled lab (Tue Apr 11 or Wed Apr 12). Beyond that time, the staff have been instructed not to give you credit. Labs are graded on effort, not correctness. We just want to see that you tried all the exercises, and to clarify any misunderstandings or questions you have.

²Nested lists are made up of items that can themselves be lists; that's a recursive data-structure.