CISC106H - Fall 2017
Lab 2, due Sunday 9/24

Do not use accumulate unless you are answering the bonus question.

1. CHANGE PARTNERS. Find zero, one or two partners. If you had a partner last week, work with someone else this week. This is very important to your well-being in class and as a person (think about why). Go to the TA and let the TA know your names and that you will be working together for this week-long lab.

Problems for submission:

For the following problems, use only list tools discussed in class: +, slicing, function calls[1]. For each one, write at least three good assertEqual tests.

2. Write a function **sum_list** of one parameter to sum the elements of a list.
3. Write a function **concatenate**that takes two list parameters and returns the concatenation.
4. Write a function of one parameter that reverses a list, e.g.
   reverse([1,4,'a']) → ['a',4,1]
5. Write a function of one parameter to sum the elements in a nested list, e.g.
   nested_sum([1, 3, [4, [], [2,1]]]) → 11
6. Write a recursive function of two parameters that extracts the nth element, e.g.
   my_nth(1, [2,3,4,5]) → 3
7. Write a function of one parameter that reverses a nested list of arbitrary depth, e.g.
   nested_reverse([1, 3, [4, [], [2,[1]]]]) → [[[[1], 2], [], 4], 3, 1]
8. Write a recursive function of three parameters that replaces the first instance of an element, e.g.
   replace(2, 3, [1,2,3,4]) → [1, 3, 3, 4]
9. Write a recursive function of two parameters to sum the elements of a list, where the second parameter is a state variable.
   sum_list_iter([1,2,3], 0) → 6
10. Write a function **remove_all** of two parameters that removes all instances of the first parameter from the list.
11. Write a function of two parameters that removes all instances of the first parameter from a nested list.
    nested_remove(2, [1, 2, [4, [2], [2,1]]]) → [1, [4, [], [1]]]

[1]There are tricks you can use to avoid learning stuff and still meet requirements; think about what you are doing, and how it may affect your performance on the exam. In particular, don't write the reverse function using slicing backwards.

12. Not for credit: how many of the functions above can you rewrite using accumulate? (Do not write the initial versions above in terms of accumulate.)

Assignment Submission:

Make sure both names of your programming pair are in each file.

Have your python functions in a single .py file, with all the relevant assertEqual tests in the same file.

Submit on Canvas: One python file containing code and tests. Follow the TA's instructions if they vary from this document.