CMPSC-132: Programming and Computation II
Spring 2020

Lab #1

Due Date: 01/24/2020, 11:59PM

Read the instructions carefully before starting the assignment. Make sure your code follows the stated guidelines to ensure full credit for your work.

Instructions:

- The work in this lab must be completed alone and must be your own.
- **Download the starter code file from the LAB1 Assignment on Canvas. Do not change the function names or given started code on your script.**
- A doctest is provided as an example of code functionality. Getting the same result as the doctest does not guarantee full credit. You are responsible for debugging and testing your code with enough data, you can share ideas and testing code during your recitation class. As a reminder, **Gradescope should not be used to debug and test code**!
- Each function must return the output (**Do not use print in your final submission**, otherwise your submissions will receive a -1 point deduction)
- **Do not include test code outside any function in the upload. Printing unwanted or ill-formatted data to output will cause the test cases to fail. Remove all your testing code before uploading your file (You can also remove the doctest). Do not include the input() function in your submission.**

Goal:

[2.5pts] Write the function *common*(*list1*, *list2*) that takes two lists, *list1* and *list2*, and **returns** (not prints) a new list containing only the elements that are common between the two lists in descending order (without duplicates). Make sure your code works on lists of different sizes. Your code should not mutate the given lists. **Hint**: reverse sorting could be useful here

[2.5pts] Write the function *connect*(*list1*, *list2*, *k*) that takes two lists, *list1* and *list2*, and a non-negative integer *k* that is less than or equal to the length of *list1*. It **returns** (not prints) a new list containing the first *k* elements of *list1*, then all elements of *list2*, then the remaining elements of *list1*. **Hint**: slicing could be helpful here

[5pts] Write the function *countWords*(*document*), where *document* is the name of a *txt* file that contains multiple strings. This function **returns** (not prints) a dictionary whose keys are included words (in lowercase), and values their word counts. You can assume the string doesn't have any contraction words such as can't, don't, isn't, etc. The starter code already contains the code to read the txt file, just make sure the file is in the same directory as your .py file.
Hint: Remove punctuation and numbers before counting the words,

Deliverables:

- Submit your code with the file name LAB1.py to the Lab1 GradeScope assignment before the due date

Notes:

- Verify your code is returning and not printing the output, otherwise, your code will return None and the test cases will fail
- To provide input to your functions, include –i as an argument in the command line to run an interactive Python session (`python –i LAB1.py`). If you are using a main function with input(), make sure to remove it before submitting
- A doctest is provided in the starter code. To run it, type `python -m doctest -v LAB1.py`
- Mac users type python3