

In this task, you will practice using files in Python.

Goal: After completing this task, you should know and be able to use:

- Files and different file modes (writing, reading, etc.) in Python.

Task: In this assignment, you will write the program *Word Statistics*. As usual, the outputs must match the example exactly, including when writing to a file.

There are no requirements for error handling this time. You can assume that the filenames provided are correct.

Word Statistics: An old statistics agency has hired you as a consultant. One of their main assignments is to count the occurrences of words in files and documents. Since the people working there are getting older, they find it increasingly difficult to keep track of this in their heads while reading. Your job is to write a Python program to do this work for them. Their needs are as follows:

- A subroutine that takes a filename and reads all the words from the corresponding file, storing them in a suitable data structure along with a counter for each word that tracks how many times it appears in the file. Then, return the variable containing the results.
- A subroutine that takes a filename and the results from the previous subroutine and writes all the results, sorted lexicographically, to the file corresponding to the given filename, one result per line. If the file already contains content, it should be overwritten.
- A subroutine that takes the filename we want to store the results in, the results from the first function, and the name of the file from which we fetched the words. The subroutine then determines which word in the results has the highest occurrence to write the filename the word came from, the word, and how many times it appears in the file corresponding to the result filename. If multiple words share the top spot, the word that is "lexicographically smallest" (the first one you encounter when the words are sorted) should be chosen. If results are already stored in this file, the new result should be added to a new line. See the example output for the format.

Main Program: In the main program, we will ask the user which file to search through, which file to save the full results to, and make all the necessary calls to the subroutines. The last file, which only contains the words with the highest occurrence (one per run), should be named *most_frequent.txt*, but this can easily be changed by modifying the main program.

TIPS!

- You can create empty files in the terminal if needed by navigating to your lab folder and typing: `touch filename.txt`
- We do not need to handle case sensitivity, and we do not need to remove commas and periods: "Hello" and "hello" are different words, as are "hello,", "hello." and "hello".
- You can sort lists in one of the following ways: