**FILM SCRIPT ANALYSIS**

You are asked to design a Python program (in a single script file) to read a text file that contains a film script (or any other long text, like a book or etc.) and print out the frequency of the words contained within the text. Your program should remove stop words - words such as “the”, “a”, “is” etc. that do not have specific meaning.

The default name of the text files the program use, should be named as “word\_1.txt” and “word\_2.txt” and it should be stored in global variables named “FILE\_NAME\_1” and “FILE\_NAME\_2”. Also, a user can give a file name or two file names as arguments. At program start, your program should take input from the user, if it is given, and check if the given input files, or the initially declared text files in the code, exist and readable for the continuation of the program.

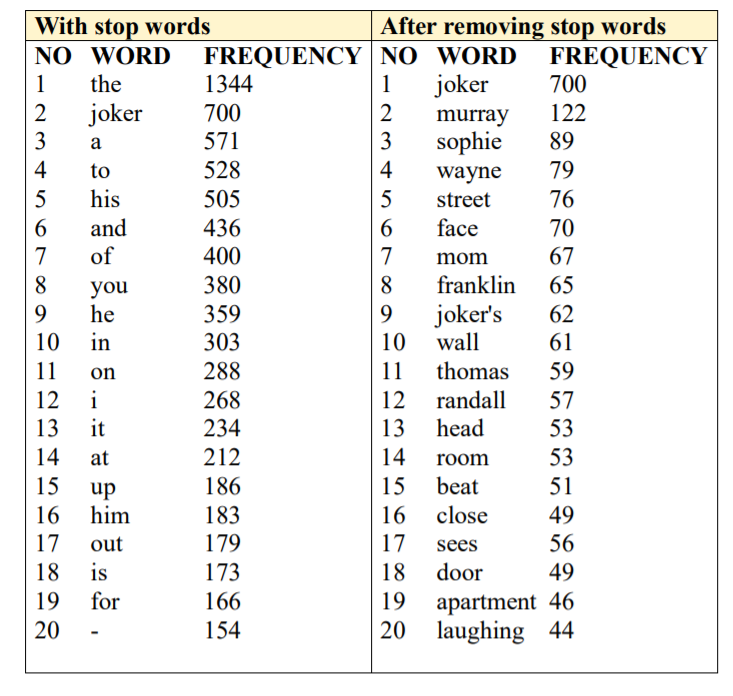
If the user gives two file names as arguments, your program will list the frequency of common words in both scripts.

It is accepted as bonus for your code to be able to download, the script of the film given by the user, from IMSDB website (“The Internet Movie Script Database (IMSDb),” n.d.). Another bonus is displaying the most frequent words in bar chart (“Bar chart,” 2019) or tag cloud (“Tag cloud,” 2019) visualization formats. You can use external libraries or other tools to implement these features, however, you are required to explain in detail the tool you have used and how did you use it

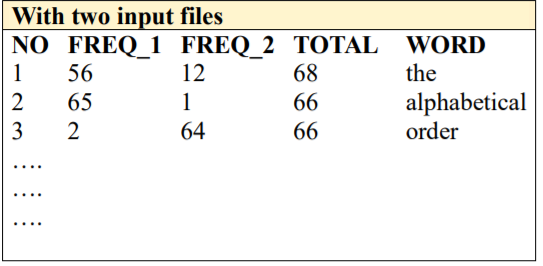
The limit of the printed words should be 20 (meaning the top 20 words with the highest frequency should be printed) and it should be adjustable within the source code via a global variable named “PRINT\_LIMIT”. If you encounter words with same frequency, you should print them out in alphabetical order.

You should explain your algorithms, details of your programming process through comments in the source code. Also you are required to prepare a report including program structure, details of the libraries and tools you used, your developing process, use-cases and results.

EXAMPLE USE CASE AND RESULTS For the first use case of the program, we use Joker (2019) film script (“Joker Script at IMSDb.,” n.d.), we will get the following results in Python console, assuming “PRINT\_LIMIT” is 20 (you can use NO, FREQUENCY and WORD ordering in table below if you wish):



The result above has been retrieved from (“Word Frequency Counter - Count Word Occurrences - Online - Browserling Web Developer Tools,” n.d.). You can use this tool to double check your results. As you can see most common words are the stop words, when they are not filtered. Second use case of the program requires two different text inputs and the corresponding output should have the following structure (given data below is dummy, meaning not real or usable):



The results should be ordered according to the total frequency (FREQ\_1 + FREQ\_2), and if the frequencies are the same, alphabetical order should be used.

**REFERENCES**

Bar chart. (2019). In Wikipedia. Retrieved from <https://en.wikipedia.org/w/index.php?title=Bar_chart&oldid=925835611>

Joker Script at IMSDb. (n.d.). Retrieved December 9, 2019, from <https://www.imsdb.com/Movie%20Scripts/Joker%20Script.html>

Tag cloud. (2019). In Wikipedia. Retrieved from <https://en.wikipedia.org/w/index.php?title=Tag_cloud&oldid=929224555>

The Internet Movie Script Database (IMSDb). (n.d.). Retrieved December 9, 2019, from <https://www.imsdb.com/>

Word Frequency Counter - Count Word Occurrences - Online - Browserling Web Developer Tools. (n.d.). Retrieved December 9, 2019, from

https://www.browserling.com/tools/wordfrequency