

Top Common Words

Matthew's Stats

Time Taken: 40 minutes

Files: 6

Lines of Code: 243 including blank lines and comments

What to Submit

- A zip file containing
 - Your .cpp and .h files that make up your solution
 - A CMakeLists.txt file that will generate an executable named **TopCommonWords** from your .cpp and .h files
 - In your CMakeLists.txt in the add_executable line, make sure you have `add_executable(TopCommonWords your_.cpp_files your_.h_files)` so the correctly named executable gets built
- Make sure to zip the files you want to submit and **NOT** the folder that contains the files. Submitting the folder with the files will cause your program to fail to build.

Restrictions and Requirements

- No global variables may be used
- Your submission must contain at least 2 or more .cpp files and one or more .h files
- Your submission must have at least 3 user defined functions in it in addition to main

Description

Write a program that displays the top N most occurring words in a file along with the number of times the word appeared.

Additional Details

- Words should be displayed from most commonly occurring to least commonly occurring
- A word is considered to be 1 or more consecutive alphanumeric characters
- Case does not matter when counting words
 - HELLO and hello are to be considered the same word
 - When displaying the most commonly occurring words they should all be displayed in lowercase
- When counting a word all leading and trailing non-alphabetical, non-numeric characters should be removed for a more accurate count

- For example
 - hello
 - hello,
 - hello.
 - hello;
 - !!\$#%hello<>?/
- Are all considered to be the same word
- The complete list of special characters is: ,.:;"'|\!@#\$\$%^&*()_+~{}<>?/~`
- If multiple words tie for most commonly occurring they should all be displayed
 - These words should be displayed in alphabetical order
- You should ignore the following words when counting the most common occurring words because they are so frequent and aren't interesting
 - a, an, and, in, is, the
- If there are fewer than N unique occurrences of a word all words should be displayed
 - For example if there were 5 unique words in a file but the user asked to display the top 10 words then only the top 5 will be displayed as there are only 5 words in the file

Input

- All input will be valid

Command Line Arguments

- First Argument: The path to the file
 - Required
- Second Argument: N, the number of top words to find
 - Will always be an integer greater than or equal to 1
 - Optional. If not given N should default to 10

Hints

- When opening the file to read from it make sure to use only an ifstream and not an fstream. This is because you only have read permissions on the files on Kodethon and opening a file with an fstream requires both read and write permissions. Since you don't have write permissions attempting to open a file with an fstream in testing will cause you to fail with weird behavior.
- The [algorithm library](#) contains many useful functions for helping to solve this problem
- You will find [maps](#) to be incredibly useful in solving this problem
 - By default a map will sort the values in ascending order. You can change this by providing a comparator function. [This example](#) shows how to do that.

Examples

- Input has been underlined so that you can differentiate between what is input and what is output
 - You do not have to underline anything
- Assume that shake_it_off.txt contains the lyrics to Taylor Swift's song "Shake it Off" which can be found here: [shake_it_off.txt](#)
- I've also provided an example executable named ExampleTopCommonWords that can be run by doing `./ExampleTopCommonWords path_to_file num_words_to_find`
 - It is only guaranteed to run on Kodethon and may not run on your personal computer

Example 1

```
./TopCommonWords shake_it_off.txt 5
1.) These words appeared 78 times: {shake}
2.) These words appeared 70 times: {i}
3.) These words appeared 44 times: {off}
4.) These words appeared 21 times: {gonna}
5.) These words appeared 15 times: {break, fake, hate, play}
```