

Homework #9 – Counting & Sorting

DUE: November 12 by 11:59:59 PM

Assigned: November 5

Background

This program will open a file and read its contents, then output: a word count, the longest word, and the average word length.

Assignment Requirements

- The name of your source code file shall be `hw09.cpp`
- The program will not take any input from the keyboard
- The file to be read will always be named `input`
- Words will be split by any character in the following string: " \t\n\r.!?,"
- Words will be have its ends stripped of the following characters: " !?, . - ; ; ' () [] { } @ # \$ % ^ & * _ + = / < > ` ~ \" \t \n \r "
 - If one of these characters appear in the middle of a word, like the apostrophe in a word like "don't", it must be left alone
 - Numbers will not be tested
 - Empty strings must not be counted
- The word count is a whole number
- The longest word must be enclosed in double-quotation marks
- The average word length must always show one decimal place
- Input files are provided, including the file that you will graded against
 - A sample run of `input.GRADE` will not be shown
 - Attribution for the file `input.GRADE`: Edgar Allan Poe
- Only submit your source code

Sample Run

This sample run uses the file `input.SAMPLE`, renamed to `input` (attribution: Frank Herbert)

➤ `./a.out`

There were 61 words read.

The longest word is "little-death"

The average word length is 3.9 letters.

Hints

- When we deal with files, there are some housekeeping rules we should keep in mind
 - After opening a file, confirm that you opened it
 - Close the file as soon as you're done with it
- Like previous homeworks, functions are not required, but are still an excellent idea
- Create your own test cases where word counts and averages are easy to calculate
- Not knowing the size of the inputs ahead of time should rule out static storage

- Familiarizing yourself with <string> and <vector> capabilities can be beneficial
- Familiarizing yourself with some of the other libraries we recently covered can also be beneficial

Reminders

- Be sure to include a comment block at the top of every file with the required information
 - Refer to the General Homework Requirements handout on Blackboard
- Provide meaningful comments
 - If you think a comment is redundant, it probably is
 - If you think a comment is helpful, it probably is
 - Remember that you are writing comments for other programmers, not people who know nothing (obligatory Jon Snow) about coding
 - Comments are more helpful when they explain why, not what or how
- There will be no extensions

Preparing and Submitting

- Your code must be able to compile and run on the EECS lab machines
 - You are responsible for testing your code
 - “But it runs fine on my machine!” will **not** earn you any points
- Submit **ONLY** your source code file
- Homework submission will be handled exclusively through Blackboard