CS 280
Spring 2025

Short Assignment 2
Simple Counting Tool

February 3rd, 2025

Due Date: Friday, February 7, 2025, 23:59
Total Points: 7

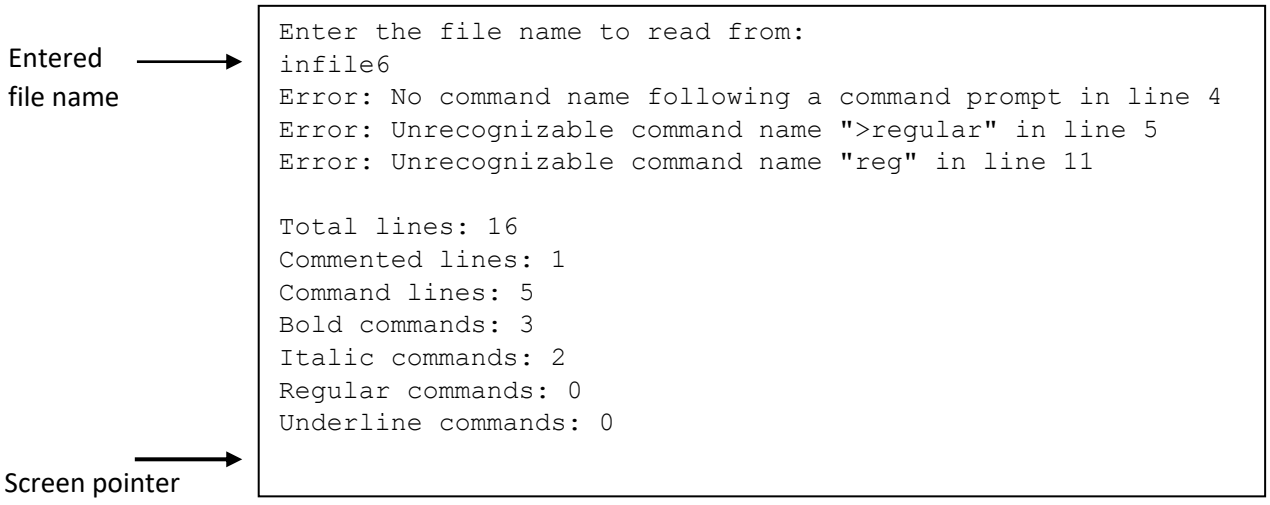
Write a C++ program that acts like a simple counting tool for collecting information from textual files of documents prepared for a simple word processing software. An input file for the simple word processor includes three types of data: commented lines, command lines and general text of the document to be formatted. A commented line is recognized by the sequence of characters "//" at the beginning of the line, and would be skipped by the word processor. A command line is recognized by the sequence of characters '>>' at the beginning of a line, followed by a command name. The simple word processor software includes four commands only for formatting textual documents. These are "bold", "italic", "underline", and "regular". Those commands would affect formatting the text of the document in the following lines, till the specification of the next command. Only one command name is considered in a command line, where all other names (or of that fact words) following the first one would be ignored by the tool. A short form of the commands' names can be recognized by the first two characters of the command name. That means the commands "bold", "italic", "underline", and "regular" can be recognized by their short forms as "bo", "it", "un", and "re", respectively. All command names or their short forms are case insensitive. The tool should generate an error message, if the command line prompt characters are not followed by a command name, or their short forms. An error message should be printed out followed by the line number. See the example below for the format of the error message.

Write a C++ program for the simple counting tool that reads lines from a file until the end of file. The program should prompt the user for the file name to read from. The program should open the file for reading, and if the file cannot be opened, it should print the message "CANNOT OPEN THE FILE ", followed by the filename, and then exit. If the input file is empty, print out on a new line the message "The File is Empty." and then exit. The simple tool should collect data about the total number of lines read from the file, the number of commented lines, the number of command lines, and the number of bold, italic, underline, and regular commands. An example of an input file and the expected results after processing the file are shown below.

Given the following input file contents,

Line number	File contents
1	3456 george 10.25
2	>>bold
3	
4	>>
5	>>>regular
6	// 4321 staci 12.7
7	>> ITALIC
8	67899 smith 9643.45
9	
10	
11	>>Reg
12	>> IT
13	>> Bold
14	278 hello -
15	>> BO
16	+654
17 →	
End of File marker	

the generated results are as follows:



Notes:

1. **The example assumes that the file name is entered from the keyboard.**
2. There are 17 lines in this input file.
3. The screen pointer is at a new line after displaying the results.
4. You have to apply the same format in order to have exact match.

Hints:

1. Download the zipped file for the test cases from Canvas. These are the test cases you will be graded against on your submission to Vocareum. Use the test cases to test your implementation. Note that case 1 is not included in the set. Your program will be checked against a file name, infile1, that does not exist.

2. There are 6 test cases, case1 through case6. Each test case file includes a file name similar to what you would type from the keyboard. Case1 includes a non-existing file, infile1, while case2-case6 files include infile2 through infile6 for existing files. Expected correct outputs are included in the files case1.correct-case6.correct. The actual input files are infile2-infile6.

3. If you want to look at the input for one of the test cases, use the linux "cat" command. The cases are in the directory $LIB/public/RA_Spring2025/SA2. You can, for example, look at infile3 by saying "cat $LIB/public/ RA_Spring2025/infile3", and you can look at the expected output by saying "cat $LIB/public/ RA_Spring2025/case3.correct".

Submission Guidelines

1. Please name your file as "RAx_firstinitial_lastname.cpp". Where, "firstinitial" and "lastname" refer to your first name initial letter and last name, respectively, and "x" refers to the short assignment number (e.g., 1, 2, etc). Uploading and submission of your program is via Vocareum. Follow the link on Canvas for SA 2 Submission page.

2. **Submissions after the due date are accepted with a fixed penalty of 25%. No submission is accepted after Sunday, February 9, 2025, 11:59 pm.**

Grading Table

Testing Cases	Points
Case 1: File cannot be opened	1.0
Case 2: Empty File	1.0
Case 3: All whitespace	1.0
Case 4: Test file "infile4"	1.0
Case 5: Test file "infile5"	1.0
Case 6: Test file "infile6"	1.0
Compiles Successfully	1.0
Total	7