CS 280
Spring 2022
Recitation Assignment 2
September 15, 2022

Due Date: Monday, September 19, 2022, 23:59
Total Points: 8

Write a C++ program that acts like a simple counting tool for collecting information from textual files of documents prepared for a simple scripting language interpreter. An input file for the simple scripting interpreter includes two types of data, commented lines and command lines. A commented line is recognized by the '%' at the beginning of the line, and would be skipped by the script interpreter. A valid command line is a one which starts with one of the four possible commands. Only one valid command per line is considered, followed by possibly one or more arguments. The simple scripting language includes the commands "run" for executing file(s), "print" for printing one or more files, and "copy" for copying a file to another file, and "delete" to delete one or more files. The simple counting tool should collect data about the total number of lines read from the file, the number of commented lines, the number of valid command lines, the number of invalid command lines, and the number of "run", "print", "copy" and "delete" commands. Note, a line should be skipped if it is empty or consists of white space characters only. In that case, the line is not counted as either valid or invalid command line.

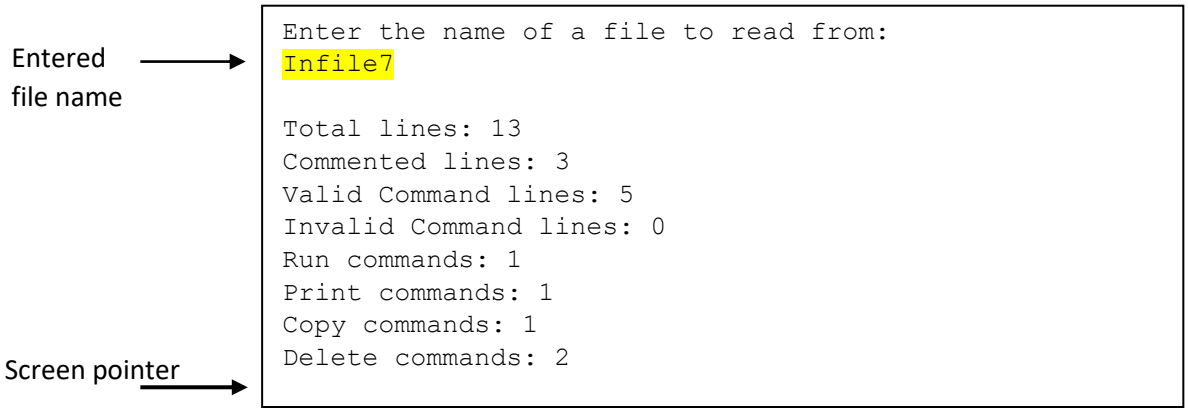
Write a C++ program for the simple counting tool that reads lines from a file until the end of file. The program should prompt the user for the file name to read from. The program should open the file for reading, and if the file cannot be opened, it should print the message "File cannot be opened ", followed by the filename, and exit. The program should consider the first word of a non-commented line as a command. If no word is found, or a word is not recognized as one of the four commands, an error message should be printed out for an invalid command line, followed by the its line number. See the example below for the format of the error message.

After reading the contents of the input file, the program should print out the total number of lines, the number of commented lines, the number of command lines, the number of invalid command lines, and the number of "run", "print", "copy" and "delete" commands. An example of an input file and the expected results after processing the file are shown below. Given the following file contents,

Line number	**File contents**
1	run prog1
2	%
3	delete
4	% execute next command
5	delete myfile
6	
7	
8	copy file1 file2
9	
10	

```
11                          print myfile
12
13                          % terminate script
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 13 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. You can use *get()* or *getline()* methods for reading from the input file. However, it is recommended that you use *getline()* to read an entire line into a string.

2. A string can be treated like an array of characters.

3. 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.

4. There are 7 test cases, case1 through case7. 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 infile7 for existing files. Expected correct outputs are included in the files case1.correct-case7.correct. The actual input files are infile2-infile7.

5. 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_Fall22/RA2. You can, for example, look at infile3 by saying "cat $LIB/public/ RA_Fall22/infile3", and you can look at the expected output by saying "cat $LIB/public/ RA_Fall22/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 recitation assignment number (e.g., 1, 2, etc). Your program Submission is to Vocareum environment. Follow the link of Recitation Assignment 2 on Canvas in the Modules or Assignments pages to connect to the current assignment on Vocareum.
2. **Submissions after the due date are accepted with a fixed penalty of 25% from the student's score. No submission is accepted after Wednesday September 21st, 2022, 11:59 pm.**

Grading Table

Testing Cases	Points
Case 1: File cannot be found	1.0
Case 2: Empty File	1.0
Case 3: All whitespace	1.0
Case 4: Commented lines only	1.0
Case 5: Invalid Command lines only	1.0
Case 6: Command lines only	1.0
Case 7: Mixed comments & command lines	1.0
Compiles Successfully	1.0
Total	8