CSE-165 Lab 03

Points: 100

Write a separate .cpp file for each of the following tasks.

Zip all your .cpp files together and submit the zip file to CatCourses.

---

1. **Character Matcher (10 points)**

Write a program that keeps reading in strings of varied sizes. If an input string has length greater than 1, store it in a vector. If an input string has length 1 (a single character), output the strings stored in your vector that have the first letter matching the input character. Keep doing this until you read the string "quit", at which point the program ends.

Example Input: The user inputs the words "Apple", "Orange", "Banana", "Apricot" and then the character "a"

Output:
Apple
Apricot

---

2. **Character Repeater (10 points)**

Write a simple run-length encoder: you will read a sequence of pairs containing a character and a number. For each pair (C,N), output the character C a total of N times without spaces. When a pair has number 0, print a newline. If the number is -1, the program ends.

Example Input: The user inputs "h" and then "3"

Output: hhh

CSE-165 Lab 03

Points: 100

3. **Integer Accumulator (20 points)**

Write a program that keeps reading in integer values. If an input value is positive, store it in a vector. If an input value is negative, remove all existing values in your vector with the same *absolute* value. When 0 is read, output the number of entries remaining in the vector and the sum of all entries, then stop.

Example Input: The user inputs the sequence: ***1 2 3 4 5 6 -5 0***

Output: 5 16

---

4. **Vector of Vectors of Strings (20 points)**

Write a program where you will have a vector V where each entry is a pointer to a vector of strings. Your program will then read input strings. For each string, if the number of characters in the string is N, then add it to the string vector in entry V[N-1]. (Be sure to allocate the string vector in each entry as needed.) The input strings will have a maximum of 10 characters so you can initialize V with 10 entries. Do not add repeated entries. Stop when string "quit" is read. String "quit" should not be processed. Then output the contents of each V entry in order from V[0] to V[9], separated by spaces within the same V entry and by a new line when switching to the next entry. Skip empty entries.

Example Input:
Enter the input: A
Enter the input: Box
Enter the input: Cube
Enter the input: to
Enter the input: happy
Enter the input: hello
Enter the input: computer
Enter the input: mouse
Enter the input: happiness
Enter the input: quit

Output:
A
to
Box
Cube
Happy hello mouse
(empty line)
(empty line)
computer
happiness
(empty line)

---

5. **Counting the Characters and Strings in V (20 points)**

Extend the program in exercise 4 in the following way: after all input strings are read, you will output for each non-empty entry of V the number of letters in that entry and the number of strings in that entry.

Example Input: Same as question 4

Output:
1 1
2 1
3 1
4 1
15 3
8 1
9 1

---

6. **Adding and Searching Strings (20 points)**

Write a program where you keep reading input strings, and then: if the number of characters is four or more, you will store the string in a vector; if the number of characters is less than or equal to three, you will find the strings which begin with the input string in the vector of strings, and then you will output the found strings. Stop when string "quit" is read. String "quit" should not be processed.

CSE-165 Lab 03

Points: 100

Example Input 1:
Hello
HelloWorld
HelloHarry
Car
Careless
Hel
quit

Output:
Hello
HelloWorld
HelloHarry

Example Input 2:
Careless
HelloWorld
Hel
Hello
HelloHarry
Car
quit

Output:
HelloWorld
Careless