

Assignment #6 (50 points; due 11/29/2017 @ 11:59 P.M.)

For all Programs:

For each program make sure and include the following comments at the top (do this on all homework assignments from now on – this is required):

```
// Your Name
// CS 1, Section #
// Assignment #, Problem #
// Summary of the program
```

Then, within the program, you will add pseudocode as appropriate to describe the steps of the program. This is in order to get in the habit of writing pseudocode and documenting your code. PSEUDOCODE IS REQUIRED FOR ALL PROGRAMS.

Email your source code and pseudocode by the due date to eambrosio@elcamino.edu. Make sure and have the subject of your e-mail be “CS 1, Section # XYZ (either 120 or 121), Assignment #1 – Your Name”. Each of your C++ file should be in this format:

FirstInitialLastName-CS1-Section#-Assignment#-Problem#.cpp

For example, it'll look similar to this:

EAmbrosio-CS1-121-A6-P1.cpp

This will be the standard format for turning in all homework assignments. The e-mail must be received by 11:59 P.M. on the due date.

This one program addresses Chapter 10.

String Processing

Write a program that manipulates a string entered by the user.

The program should start by asking the user to enter a word, a sentence, or a string of numbers. Store whatever the user enters into a C++ string. (In other words, you will be using the string class provided by C++ by adding #include <string>.)

The program should then display the following menu:

USE THIS MENU TO MANIPULATE YOUR STRING

-
- 1) Inverse String
 - 2) Reverse String

- 3) To Uppercase
- 4) Jumble String
- 5) Count Number of Words
- 6) Count Consonants
- 7) Enter a Different String
- 8) Print the String
- Q) Quit

If the user selects 1: Inverse the upper and lower case letters of the string. If the string contains numeric characters or special characters do not change them. NOTE: This option should actually change the string to its inverse. Note this option does not display the changed string. If a user wanted to inverse the string and then display the string's inverse they would select option 1 and then they would select option 8.

Example: If the string is: My name is John and I am 20 years old.

The inverse would be: mY NAME IS jOHN AND i AM 20 YEARS OLD.

If the user selects 2: – Reverse the order of the characters in the string. NOTE: This option should actually change the string to its reverse. Note this option does not display the changed string. If a user wanted to reverse the string and then display the string's reverse they would select option 2 and then they would select option 8.

Example: If the string is: 2015

The reverse would be: 5102

If the user selects 3: Convert all of the characters in the string to uppercase. If the String contains numeric characters or special characters do not change them. **NOTE:** This option should actually change the string to all uppercase letters. Note this option does not display the changed string. If a user wanted to change the string to uppercase and then display the new string (in all uppercase) they would select option 3 and then they would select option 8.

If the user selects 4:– Call a function named jumbleString. The jumbleString function takes a string as input and displays a jumbled version of that string. The jumbleString function should be called using current version of the string an argument (input) to the function.

Example: If the string passed to the jumbleString function is: hello

A jumbled version of the word would be: elhlo

NOTE #1: There are many different jumbled versions of each word that the `jumbleString` function could display. So `elhlo` is not the only correct output for the above example.

NOTE #2: Notice that this option does not actually change the string like the first two menu selections do, the `jumbleString` function just displays a jumbled version of the string rather than actually changing the string

If the user selects 5: Call a function named `countWords` that counts the number of words in the current string and displays a message stating how many words are in the string.

Examples: The string “2015” has one word

The string “Hello World” has two words

The string “ I am Woman ” has 3 words

If the user selects 6: Call a function named `countConsonants` that counts the number of consonants in the current string and displays a message indicating how many consonants the string contains. Consonants are letters that aren't vowels.

Example: If the string is: Hello

The number of consonants is: 3

So display: “The number of consonants in the string is 3.”

If the user selects 7: Let the user enter another string for processing. (This should just change the string stored in the original string variable you created at the beginning of the program.)

If the user selects 8: Print the String

So if the original string was “Hello” and the user processed the string with option 3 followed by option 2, followed by option 8, the string would print out as “OLLEH” (This is hello first converted to uppercase and then reversed).

If the user selects ‘Q’ or ‘q’: Quit the program

Allow the user to continue processing strings (using the menu) until they select ‘Q’ or ‘q’ to quit.

If the user makes an invalid menu selection, print an error message to the screen.

Your program must include the following three functions (you can include more if you want):

1. `countWords`: A function that takes a string as an argument and counts the number of words in it (this function should return an int which is the number of words in the string)
2. `countConsonants`: A function that takes a string as an argument and counts the number of consonants in it (this function should return an int which is the number of consonants in the string)
3. `jumbleString`: A function that take a string as an argument and displays a jumbled version of the original string. (HINT: This is a display function and display functions typically return nothing, as it does the print out or display within the function.)