LAB DIRECTIONS for LAB 4/09/2015

Based on Lesson Set 10

1. Create a folder named **Lesson Set 10**. Put all files you create for this lab in this folder.

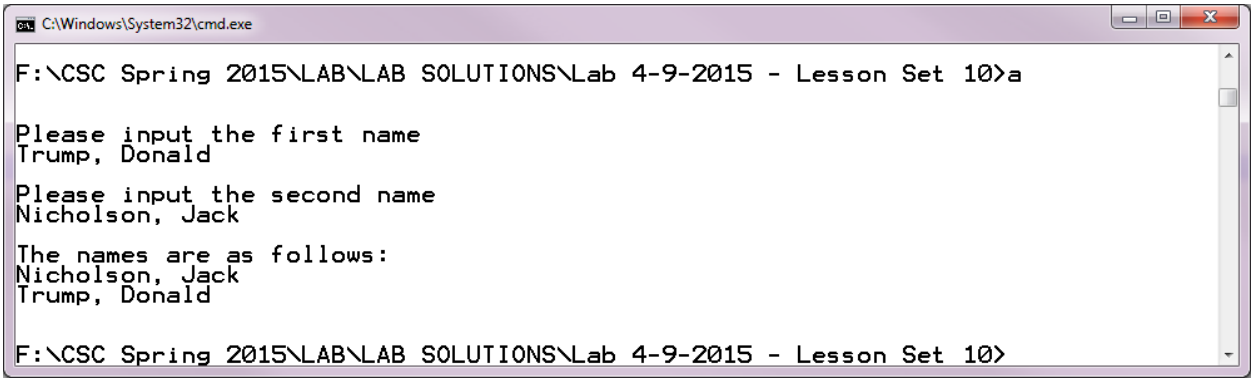
2. There will be no source files provided for you for this lab. You will be creating all files yourself.

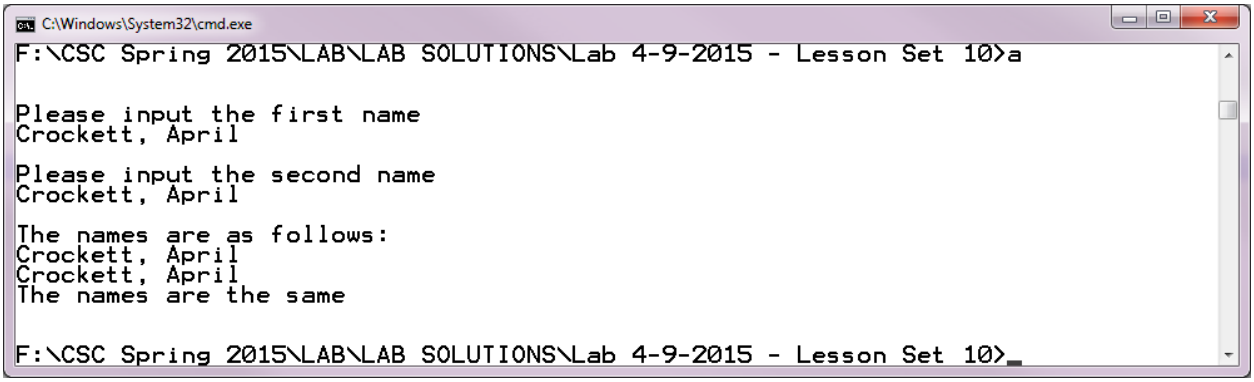
3. Write a program named `namecompare.cpp` that will read in two names and then print the names in alphabetical order.
   a. The `strcmp(string1, string2)` function compares `string1` to `string2`. It is a value returning function that returns a negative integer if `string1` < `string2`, 0 if `string1` == `string2`, and a positive integer if `string1` > `string2`. Write a program that reads two names (last name first followed by a comma followed by the first name) and then prints them in alphabetical order.

   b. The two names should be stored in separate character arrays holding a maximum of 25 characters each.

   c. Use the `strcmp()` function to make the comparison of the two names. Remember that `'a' < 'b'`, `'b' < 'c'`, etc. Be sure to include the proper header file to use `strcmp()`.

   d. Make two screen captures of this program running using the same user input as in my screen captures below. Put your screen capture in a document named Lesson Set 10 Screen Captures.



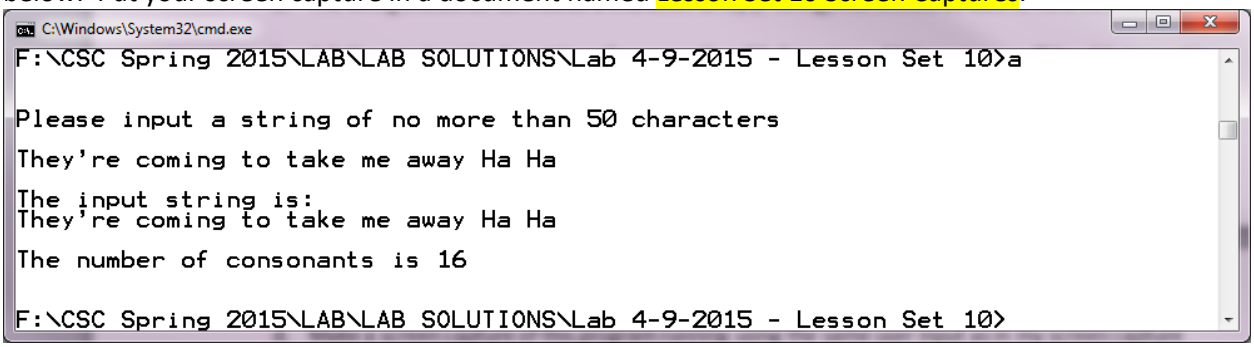


4. Write a program named `countCons.cpp` that will determine how many consonants are in an entered string of 50 characters or less. Output the entered string and the number of consonants in the string.
   a. The string should be defined as a character array (c-string) that should be able to hold 50 characters.

   b. Ask the user to enter in a string of no more than 50 characters. Make sure you will be able to read in spaces as well as other types of characters.

   c. Determine the number of consonants in the string. Include the letter 'y' as a consonant – not a vowel.

d. Make a screen capture of this program running using the same user input as in my screen capture below. Put your screen capture in a document named Lesson Set 10 Screen Captures.



What to Turn In: (by Wednesday, April 15, 2015)

- namecompare.cpp
- countCons.cpp
- Lesson Set 10 Screen Captures

How you will be graded

namecompare.cpp	40 points	FOLLOWS SPECIFICATIONS / DEFINES STRINGS AS C-STRINGS / USES CIN.GETLINE CORRECTLY / USES STRCMP / ALPHABETIZES STRINGS CORRECTLY
countCons.cpp	45 points	FOLLOWS SPECIFICATIONS / USES C-STRING TO STORE STRING / USES CIN.GETLINE CORRECTLY / CREATES A WORKING ALGORITHM TO FIND # OF CONSONANTS / PRINTS OUT STRING & # OF CONSONANTS
Lesson Set 10 Screen Captures	15 points	Three screen captures total – two for namecompare.cpp and one for countCons.cpp (5 points each)