Lab 6

Due Mar 12 by 11:59pm **Points** 25 **Submitting** a file upload **File Types** cpp

Strings and C-Strings: Chapter 8

The **General Programming Lab Instructions** apply to this assignment.

A palindrome is a phrase (i.e., a string of characters) that reads the same forward and backward (ignoring spaces, punctuation, and capitalization). For example: *A man, a plan, a canal: Panama!* is a palindrome. This lab consists of two programs that determine if a string represents a palindrome. Each program uses a different kind of string: The first program uses the C++ string class while the second program uses C-strings. Please note the following general rules:

- You may use any code that you wish from the example programs as a part of either of your programs.
- Do not include code from the examples that does not pertain to the labs

Test Cases

Valid Palindrome

a man a plan a canal panama

Invalid Palindrome

a man a plan a canax panama

Valid Palindrome

never odd or even

Invalid Palindrome

never odd or ven

Program 1: Palindromes With The string Class

Program 1 Requirements

1. Name your program `palindrome.cpp`
2. **All strings in the program are implemented as instances of the C++ string class**
3. Prompt for the user to enter a potential palindrome
4. Read the user input in to an instance of the string class (see string I/O (http://icarus.cs.weber.edu/~dab/cs1410/textbook/8.Strings/string_class_io.html) for an example)
    a. The test input will NOT contain punctuation characters (periods, commas, exclamation or question marks), etc. This means that your program is not required to remove punctuation characters.
    b. The test input will consist of only one case (i.e., it will not contain a mix of upper and lower case letters). This means that your program is not required to change the case of the input or test for upper or lower case letters.
    c. The test input will contain spaces, which your program must remove before testing the input to see if its a palindrome (see the code fragment below).
5. Test the string to see if it is a palindrome (follow either of the two examples in 8.3.4. palnumber.cpp (http://icarus.cs.weber.edu/~dab/cs1410/textbook/8.Strings/progexample/palnumber.html))
6. Write a message to the screen stating that the input was or was not a valid palindrome
7. The program terminates after printing the message - no prompts, loops, or pauses

Removing Spaces From A String

It is probably easiest to first remove all of the spaces from the string before testing it to see if it is a palindrome. The following code fragment (which you may use in your program) will remove the spaces from a string object:

```
        for (int i = 0; i < s.length(); i++)
                while (s[i] == ' ')
                        s.erase(i, 1);
```

Program 2: Palindromes With C-Strings

Program 2 Requirements

1. Name your program `cpalindrome.cpp`
2. **All strings in the program are implemented as C-strings**
3. The program accepts a potential palindrome on the command line
    a. The program does not prompt for nor read input from cin
    b. The test input will NOT contain punctuation characters (periods, commas, exclamation or question marks), etc.
    c. The test input will consist of only one case (i.e., it will not contain a mix of upper and lower case letters)
    d. The command line processor removes all spaces from the test input and places each word in a separate element of argv (see "Parsed Command Line" below)
    e. Concatenate all of the needed elements of argv together to form the candidate palindrome (see "Concatenating C-Strings" below)
    f. Review how **command line arguments** **(http://icarus.cs.weber.edu/~dab/cs1410/textbook/8.Strings/commandline.html)** work
    g. See the command line version of the **8.2.5. name_box** **(http://icarus.cs.weber.edu/~dab/cs1410/textbook/8.Strings/progexample/namebox2.html)** program for an example of accessing command line arguments and of entering command line arguments in Visual Studio
    h. Test the string to see if it is a palindrome (follow either of the two examples in **8.2.7. cpalnumber.cpp** **(http://icarus.cs.weber.edu/~dab/cs1410/textbook/8.Strings/progexample/cpalnumber.html)**)
4. The program writes a message stating that the input was or was not a valid palindrome
5. The program terminates after printing the message - no prompts or pauses

The Parsed Command Line

The system parses each word entered on the command line and stores it in a separate c-string that is one element of the argv array. For example, if the "panama" palindrome is entered on the command line (all lower case with all punctuation except for spaces removed), the resulting argv array would look like the following (note that the following illustrates how the input is stored in argv and argc, not something that you program):

```
argc = 8
argv[0] = cpalindrome (the program name - often not of interest and also not entered in VS)
argv[1] = a
argv[2] = man
argv[3] = a
argv[4] = plan
argv[5] = a
argv[6] = canal
argv[7] = panama
```

Concatenating C-Strings

Concatenate the command line arguments into a single c-string:

```
        char    palindrome[1000] = "";  // empty string
        for (int i = 1; i < argc; i++)
                strcat_s(palindrome, 1000, argv[i]);
```

About strcat_s

strcat_s is a Microsoft variation of strcat and will not be available on other platforms. If you are working on Linux or Mac, then you have two options:

1. Use the standard concatenation function `strcat` and add `#define _CRT_SECURE_NO_DEPRECATE` at the **top of your file** (this is also unique to Microsoft but will not cause any problems on other platforms)
2. Use `strcat` during program development and replace it with `strcat_s` prior to uploading your code
3. Choice 1 is also a viable option if you are working on Windows with Visual Studio

Instructions for setting "command line" arguments in Visual Studio:

1. Select "Project" from the main menu OR in the Solution Explorer, right-click the project name
2. Select "Properties" (at the bottom of the pop up menu
3. Expand "Configuration Properties" (on the left) and select "Debugging
4. Enter the arguments separated by spaces (do not include the program name) in the input box labeled "Command Arguments" and Press "OK"

Using the Linux/Mac Command Line

If you are developing your code on a Linux platform, write and compile your program as you normally would. Open a shell or terminal window and cd to your executable. If you named the output with the -o option (or if you are using an IDE that automatically names the the executable file, then replace "a.out" with the appropriate program name). From the command line, execute your program and specify the input:

```
./a.out a man a plan a canal panama
```

Lab 6 Scoring

Criteria	Ratings	Pts
palindrome.cpp Name The program file is named correctly.		1.0 pts
palindrome Detects Valid Palindrome The program correctly identifies a valid palindrome.		3.0 pts
palindrome Rejects Invalid Palindrome The program correctly identifies a string that is not a palindrome.		3.0 pts
palindrome string Class Used Throughout The program only uses the C++ string class (no C-strings are used).		1.0 pts
palindrome Input Allows Spaces The program accepts and processes potential palindrome strings that contain spaces.		1.0 pts
palindrome No Extraneous Code The program does not contain code that does not contribute to solving the assigned problem.		1.0 pts
cpalindrome.cpp Name The program file is named correctly.		1.0 pts
cpalindrome Detects Valid Palindrome The program correctly identifies a valid palindrome.		3.0 pts
cpalindrome Rejects Invalid Palindrome The program correctly identifies a string that is not a palindrome.		3.0 pts
cpalindrome C-string Used Throughout The program only uses C-strings (no instances of the string class are used).		1.0 pts
cpalindrome Command Line Input The program takes all input from the command line and does not prompt the user to enter data.		2.0 pts

Criteria	Ratings	Pts
cpalindrome No Extraneous Code The program does not contain code that does not contribute to solving the assigned problem.		1.0 pts
Miscellaneous (Both Programs) Projects created "Empty" and all "pause" or dummy read statements are commented out or removed. Programs are well-structured, indentation is consistent, blank lines enhance readability, etc.		4.0 pts
Total Points: 25.0		