Program #4

Program Purpose

Using Visual Studio 2017 create a console application. Create a program plan and then convert it into C++ statements. Practice debugging, declaring variables, type casting, formatting output, if statements and input /output to / from the console window and files, input validation, and loops.

Always bring to class

1. Gaddis' book, How-to handouts from the Blackboard and your class notes.
2. This assignment sheet & the grade sheet for this lab already printed out.
3. USB Flash drive(s) or other storage media.

LATE PROGRAMS will be not be accepted. Please be sure to upload compressed solution and Word document grade sheet to Canvas by due date/time.

Mandatory Instructions

Work with a partner on this program. Make sure to include the partner's name in the grade sheet. Each of you needs to upload the zip file and provide own grade sheet.

Write a program that will check if a word or a sentence is a palindrome. A palindrome is a word (or sentence) that is the same when read front to back and back to front. For example: race car, mom, poop, etc. Sentences can also be palindromes, for example "Ten animals I slam in a net" is a palindrome.

In order to verify that a word is a palindrome it needs to be reversed and then compared to the original. If both are the same, then we can conclude it is a palindrome. Sentences need a bit more processing. All spaces, punctuation marks have to be first removed from the sentence. Then in (both word and sentence case) we need to upper case all letters. Then sentence needs to be reversed and compared to the compressed and upper cased sentence.

Example:
Initial input: Ten Animals I slam in a net.
Compressed: TenAnimalsIslaminanet
Lower cased: TENANIMALSISLAMINANET
Reversed: TENANIMALSISLAMINANET
The last two are the same ➔ so this sentence is a palindrome.

You will need to set up the main processing of your program as a DO WHILE loop. After one word or sentence is entered and processed you will ask the user if he/she wants to enter another. If they answer "Yes" or "yes" then the loop should continue and capture more input from the console window. If the user answers anything else then program should end. Keepgoing should be declared as a bool variable.

```
do
{
     // Processing for a word/sentence

} while (keepgoing != false)
```

Inside this main loop you should do the following:

1. Get user input from the console window
2. Compress the input, i.e., remove all punctuation characters and blanks and upper case each letter (this MUST be a separate loop inside of the main processing loop)
3. Reverse compressed string (this MUST be another separate loop inside the main processing loop)
4. Compare compressed string to reversed string… show appropriate message to the user
5. Ask if the user wants to provide another input word/sentence

*** * * NOTES * * ***

1. The step 2 and 3 above **MUST be** implemented as while loops (no built in functions to do your work are allowed here!!! --- for example to reverse a string)
2. When capturing word, sentence, choice of whether to continue or not, use getline(cin, yourinputvar);
3. The input sentences can have the following punctuation characters: ! , . ‘ ? ; “ ( ) - and space

Program #4

BONUS Instructions

When the user decides not to keepgoing process the input file “prog4_input.txt” and translate all sentences found in the file producing an output file “prog4_report.txt”. There should be one message shown on screen “File processing done” when file is closed. Here are some sample palindromes that you will find in the file:

Milestones? Oh, 'twas I saw those, not Selim.
No, it's a bar of gold, a bad log for a bastion.
"Naomi, sex at noon taxes", I moan.
Snug & raw was I ere I saw war & guns.
Doc, note, I dissent. A fast never prevents a fatness. I diet on cod.
Live was I ere I saw evil.

Program Documentation & Style:

1. Declare all variables and constants that your program uses at the beginning of your program.
2. Your program should include two types of comments:
   a. Header Comments at the top including lines with:
      - Your name, course name, and class time
      - Program assignment number, program file name and due date
      - A sentence or two explaining the purpose of the program
      - A description of the input data needed by the program when you run it
      - A description of the processing (calculations) done by the program
      - A description of the results (output) produced by the program

   b. In-line comments: There should be an in-line comment for each main step in your program. In general, this means a comment with each group of C++ statements that handle the declarations, input, processing, and the output steps of your program
3. Use meaningful identifier names
4. Include clear prompts for the user about entering the data
5. Include clear descriptions of the results when you display them

What to turn in?

1. Log into Canvas, locate this assignment, and upload the compressed project folders.
2. Upload the grade sheet (Word document) after you have edited it to provide requested information.
3. You’re done.

Program #4