A file, called data, contains some unknown number of lines that contain 1 or more non-space characters. You are to read them all in, and output each along with whether the line is a palindrome or not, disregarding punctuation, spaces, and capitalization.

You will write the following functions (prototypes given) and no others:

- **string process(string);** returns a string such that it is in lower case and contains no punctuation or spacing
- **bool is_palindrome(string);** should use a recursive method to determine if the string is a palindrome
  - Remember that a recursive solution has both a base case (or cases) and a recursive case (or cases) and you must decide which to use using some sort of logic.
- **int palindrome_fix_location(string);** should be as described below:
  - This function should return a location where some number of characters could be added to a non-palindrome string to make it a palindrome.
  - There are many ways to do this, for example the string aabcaa can be made into a palindrome by simply reversing and duplicating the string at position 0 (before the string) or position 6 (after the string), but a more elegant solution would determine that you could just add the character c at location 2.
  - An optimal solution will determine the minimum number of characters to add to the string. You do not need to find the optimal solution but if you do then you will receive 1 extra credit PD point. Likewise, PD deductions will result if your program always reports that it should simply reverse the entire string (or the entire string minus one character) and put it at the beginning or end.
- **string palindrome_addition(string, int);** given a string and the location where a string should be added to make it a palindrome, this should return the text that needs to be added at that location

Using the above functions, write a program that will produce output similar to that on the next page -- importantly, your output should be identical with the potential exception of the location of the replacement and the replacement itself.

**You are not allowed to use multiple string variables in your main body. No arrays are allowed.**

Make sure to write comments as such:

- To describe the program
- To describe the function
- To describe any particularly complicated logic or loops

See an example compilation and execution on the next page with input in **bold.**
ls
data palindrome.cpp
cat data
lappal
lapal
A man, a plan, a canal, Panama!
lap
alapa
bath
aabcaa
abc...123...cba

g++ -Wall -Werror -Wpedantic -Wextra -std=c++11 palindrome.cpp
./a.out
Original line: lappal
Processed line: lappal
Line is a palindrome

Original line: lapal
Processed line: lapal
Line is a palindrome

Original line: A man, a plan, a canal, Panama!
Processed line: amanaplanacanalpanama
Line is a palindrome

Original line: lap
Processed line: lap
Line is NOT a palindrome
Characters to insert at location 0 are pa
Final line: palap

Original line: alapa
Processed line: alapa
Line is NOT a palindrome
Characters to insert at location 1 are pa
Final line: apalapa

Original line: bath
Processed line: bath
Line is NOT a palindrome
Characters to insert at location 0 are hta
Final line: htabath

Original line: aabcaa
Processed line: aabcaa
Line is NOT a palindrome
Characters to insert at location 2 are c
Final line: aacbcaca

Original line: abc...123...cba
Processed line: abc123cba
Line is NOT a palindrome
Characters to insert at location 3 are 32
Final line: abc32123cba