```
Script started on 2023-11-29 18:17:15-06:00 [TERM="xterm" TTY="/dev/pts/0" COLUMNS=
mf98604@ares:~$ pwd
/home/students/mf98604
mf98604@ares:~$ cat palindrome.info

Name:        Philip May'r
Class:       CSC121-001

Activity:    Palindrome Determining Program

Options:     Add (Level 1.5) to make your palindrome determining code
             into a function.
             Add (Level 1.5) to check for phrase palindromes
             as well as word palindromes.
             Add (Level 1.5) to place your palindrome checking function(s)
             in a library.

Level:       7 (2.5 + 1.5 + 1.5 + 1.5)

Description: Determines whether a word, phrase, or sequence is a palindrome.

mf98604@ares:~$ show-code palindrome.cpp


palindrome.cpp:


#include "ispalindrome.h"
#include <iostream>
#include <string>

using namespace std;

int main(void)
{
    string input_str;

    cout << "\nA Palindrome is a word, phrase, or sequence that reads "
         << "alike backward as forward."

         << "\n\nThis program looks for palindromes. "
         << "Enter a word, phrase, or sequence below:\n\n";

    getline(cin, input_str);

    cout << "\n\"" << input_str << "\""
         << " is " << (is_palindrome(input_str) ? "" : "Not ")
         << "a Palindrome.\n\n";
}
mf98604@ares:~$ show-code ispalindrome.cpp


ispalindrome.cpp:


#include "ispalindrome.h"
#include <cctype>
#include <string>

bool is_palindrome(std::string str)
{
    std::string::size_type len = str.length();
    bool is_palindrome = true;

    for (std::string::size_type i = 0, j = len - 1;
         i <= j && is_palindrome;
         i++, j--)
    {
        if ((ispunct(str[i]) || isspace(str[i])) &&
            (ispunct(str[j]) || isspace(str[j])))
        {
            continue;
        }
        else if (ispunct(str[i]) || isspace(str[i]))
        {
            j++;
        }
        else if (ispunct(str[j]) || isspace(str[j]))
        {
            i--;
        }
        else if (tolower(str[i]) != tolower(str[j]))
        {
            is_palindrome = false;
        }
    }

    return is_palindrome;
}
mf98604@ares:~$ show-code ispalindrome.h


ispalindrome.h:


#include <string>

#ifndef ISPALINDROME_H_INC
#define ISPALINDROME_H_INC

bool is_palindrome(std::string str);

#endif
mf98604@ares:~$ CPP palindrome ispalindrome
ispalindrome.cpp...
palindrome.cpp***
```

```
mf98604@ares:~$ ./palindrome.out

A Palindrome is a word, phrase, or sequence that reads alike backward as forward.

This program looks for palindromes. Enter a word, phrase, or sequence below:

In the beginning

"In the beginning" is Not a Palindrome.

mf98604@ares:~$ ./palindrome.out

A Palindrome is a word, phrase, or sequence that reads alike backward as forward.

This program looks for palindromes. Enter a word, phrase, or sequence below:

abba

"abba" is a Palindrome.

mf98604@ares:~$ ./palindrome.out

A Palindrome is a word, phrase, or sequence that reads alike backward as forward.

This program looks for palindromes. Enter a word, phrase, or sequence below:

radar

"radar" is a Palindrome.

mf98604@ares:~$ ./palindrome.out

A Palindrome is a word, phrase, or sequence that reads alike backward as forward.

This program looks for palindromes. Enter a word, phrase, or sequence below:

1234567654321

"1234567654321" is a Palindrome.

mf98604@ares:~$ ./palindrome.out

A Palindrome is a word, phrase, or sequence that reads alike backward as forward.

This program looks for palindromes. Enter a word, phrase, or sequence below:

Never Odd or Even

"Never Odd or Even" is a Palindrome.

mf98604@ares:~$ ./palindrome.out

A Palindrome is a word, phrase, or sequence that reads alike backward as forward.
```

```
This program looks for palindromes. Enter a word, phrase, or sequence below:

Live not on Evil

"Live not on Evil" is a Palindrome.

mf98604@ares:~$ ./palindrome.out

A Palindrome is a word, phrase, or sequence that reads alike backward as forward.

This program looks for palindromes. Enter a word, phrase, or sequence below:

...MMXXX..XXXMM...

"...MMXXX..XXXMM..." is a Palindrome.

mf98604@ares:~$ cat palindrome.tpq

Thought-Provoking Questions - Lab 7 - Palindrome

1.)
An individual character within a string may be accessed using the subscript
operator []. For instance, in a string named 'class' having a value of
"Computer Science 121", class[0] returns 'C', and class[2] returns 'm'.

2.)
The size or length of a string type may be gotten by calling
the string object's .size() or .length() member function.
Both functions return the same values.

3.)
We can tell whether the first and last characters of a string are the same
or not by comparing them with the equality operator ==.

4.)
For strings with an odd number of characters, the middle character need
not be looked at, as a palindrome must be symmetrical about
the middle character. Only the characters before and after the middle
character are compared one to another.

5.)
For a string with a length of an even number of characters, each character
must be considered. For a string with a length of an odd number
of characters, the central character may be left out of consideration.

6.)
The program does consider '34743' a palindrome. The algorithm implemented
works for any string of characters, regardless of whether they be digits,
symbols, or letters.

mf98604@ares:~$ exit
exit

Script done on 2023-11-29 18:20:01-06:00 [COMMAND_EXIT_CODE="0"]
```