```
/**************************************************************************
* Program Description
---------------------------------------------------------------------------
This program takes a work and uses a recursive function to see if the
word inputed is a Palindrom. It ignores whitespaces, capital letter,and
  puntuation.



OUTPUT
---------------------------------------------------------------------------
Outputs the word entered and whether it is a palindrome or not.
**************************************************************************/




Enter a word (exit to quit): no lemon, no melon
no lemon, no melon is a Palindrome

Enter a word (exit to quit): Was it a cat I saw?
was it a cat i saw? is a Palindrome
```

```cpp
/*****************************************************************************
* AUTHOR : Saul Moreno
* ASSIGNMENT#1 : Recursion
* CLASS : CS1D
* SECTION : MW 2:00pm
* DUE DATE : 1/29/23
*****************************************************************************/
#include "Header.h"
#include "Palindrome.h"

using namespace std;

int main()
{
    Palindrome answer;        //This is an instance of the Palidrome class
    string name = " ";        //IN - Stores the user input
    int wordLength = 0;       //CALC - Used to find the length of the word
    bool finalAnswer = false; // OUT - Determines if the word is a palindrome


    cout << "/*****************************************************************************\n"
         << "* Program Description\n"
         << "------------------------------------------------------------------------------\n"
         << "This program takes a work and uses a recursive function to see if the \n"
         << "word inputed is a Palindrom. It ignores whitespaces, capital letter,"
         << "and puntuation.\n";

    cout << "\n\n\nOUTPUT\n"
         << "------------------------------------------------------------------------------\n"
         << "Outputs the word entered and whether it is a palindrome or not.\n"
         << "*****************************************************************************/\n\n\n\n\n";
    cout << "Enter a word (exit to quit): ";
    getline(cin, name);

    wordLength = name.length();

    char* stringToArray = new char(wordLength);
    answer.convertStringToLower(name, stringToArray, wordLength);


    finalAnswer = answer.recursivePalindrome(stringToArray, 0, wordLength - 1);

    if (finalAnswer == 1)
    {
        for (int i = 0; i < wordLength; i++)
            cout << stringToArray[i];
        cout << " is a Palindrome";
    }
    else
    {
        for (int i = 0; i < wordLength; i++)
            cout << stringToArray[i];
        cout << " is not a Palindrome";
    }

    return 0;
}
```

```
/*************************************************************************
* AUTHOR : Saul Moreno
* ASSIGNMENT#1 : Recursion
* CLASS : CS1D
* SECTION : MW 2:00pm
* DUE DATE : 1/29/23
*************************************************************************/
#pragma once
#include <iostream>
#include <string>
#include <vector>
#include <cctype>
#include <algorithm>
```

```cpp
/*****************************************************************************
* AUTHOR : Saul Moreno
* ASSIGNMENT#1 : Recursion
* CLASS : CS1D
* SECTION : MW 2:00pm
* DUE DATE : 1/29/23
*****************************************************************************/
#include "Palindrome.h"

Palindrome::Palindrome() {
    secondCompare = ' ';
    secondSecondCompare = ' ';
}


Palindrome::~Palindrome() {


}

/*****************************************************************************
*FUNCTION - recursivePalindrome
*_____________________________________________________________________________
*This Function receives the lower case array, uses recursion to find out
* if the word is a palindrom while ignore anything that is not a lower
* case letter
*_____________________________________________________________________________
*PRE-CONDITIONS
*     arrayTolower[]:  Has to be previously defined
*     start:           Has to be previously defined
*     end:             Has to be previously defined
*
*POST-CONDITIONS
*      This function will return a true or false value to main.
*
*****************************************************************************/
bool Palindrome::recursivePalindrome(char arrayTolower[], int start, int end) {

    if (start == end) {
        return true;
    }

    if (start > end) {
        return false;
    }

    secondCompare = arrayTolower[start];
    secondSecondCompare = arrayTolower[end];

    while ((int)secondCompare == 32 || (122 >= (int)secondCompare && (int)
      secondCompare <= 97)) {
        start++;
```

```cpp
            secondCompare = arrayTolower[start];
        }

        while ((int)secondSecondCompare == 32 || 122 >= (int)secondSecondCompare &&
          (int)secondSecondCompare <= 97) {
            end--;
            arrayTolower[end];
            secondSecondCompare = arrayTolower[end];
        }
        if (secondCompare == secondSecondCompare)
        {
            return true;
        }

        if (arrayTolower[start] == arrayTolower[end])
            return recursivePalindrome(arrayTolower, start + 1, end - 1);
        else
        {
            return false;
        }
    }

    /*****************************************************************************
    *FUNCTION - convertStringToLower
    *_____________________________________________________________________________
    *This Function take the string and turn any capital letter to a lower case
    *_____________________________________________________________________________
    *PRE-CONDITIONS
    *     word:           Has to be previously defined
    *     arrayToLower[]: Has to be previously defined
    *     size:           Has to be previously defined
    *
    *POST-CONDITIONS
    *      This function will return value to main.
    *
    *****************************************************************************/
    void Palindrome::convertStringToLower(std::string word, char arrayTolower[], int
      size) {
        char lowerCase = ' '; //IN - variable to store the letter that is now lower
          case

        for (int wordSize = 0; wordSize < size; wordSize++) {
            lowerCase = tolower(word[wordSize]);
            arrayTolower[wordSize] = lowerCase;

        }
    }
```

```
/*****************************************************************************
* AUTHOR : Saul Moreno
* ASSIGNMENT#1 : Recursion
* CLASS : CS1D
* SECTION : MW 2:00pm
* DUE DATE : 1/29/23
*****************************************************************************/
#include "Header.h"
#include <string>
#include <vector>
#pragma once
class Palindrome
{
public:
    Palindrome();
    ~Palindrome();
    bool recursivePalindrome(char arrayTolower[], int start, int end);
    void convertStringToLower(std::string word, char arrayTolower[], int size);

private:
    char secondCompare;
    char secondSecondCompare;

};


```