**CSCI 301 Moustafa Elsayed**

**Computer Science 2**

**Project 2: Identify Palindromes**

**Introduction**

Palindromes is comparing the first character with the last character, if they are the same then we move closer to the middle and compare the second character with the before the last character and so on until we reach the middle, if the characters on the left are the same on the right then we can switch each character to its opposite position and we will have the same pattern of characters (which is a palindrome) . This program takes the input of the user as a string, counts the number of characters in that input, assigns each character to the array, then calls a function to check if this input is a palindrome or not. And depending on the output of the function the program will print two different messages.

**Data Structures**

This program uses an integer “low” in the main function and is initialized with the value of 0 to mark the start of the array. And an integer “high” in the main function to hold the value of the last character inside the array. A character array “a[50]” in the main function to hold the value of each character the user inputs. A Boolean variable “result” in the main function to hold the returned value of the function “pal()” and determine which message will the program print. A string variable “candi” in the main function to hold the input of the user. A Boolean variable “flag” in the “pal()” function to signal if this input is a palindrome or not.

**Functions**

This program uses 3 functions. The function “pal()” that takes the array “a[]” , “low” and “high” variables from the main function as input. This function checks for the conditions that would mean that the program has reached the middle of the character input of the user by checking that the “low” and the “high” variable are not the same or that the “low” variable is larger than the “high” variable, if that condition is true then the function returns the flag variable. If the condition is not true then the function checks that the value of the array at the “low” location (which is the first character) is not the same as the value of the array at the “high” location (which is the last character), if that condition is true then the “flag” variable becomes false and is returned to signal that the input is not a palindrome. If the condition is not true and the value of the first character of the array is the same as the value of the last character of the array, then the “low” variable is incremented and the “high” variable is decremented to decrease the number of characters we need to compare, and the “pal()” function is recurred to assign its value to the “flag” with a larger “low” value and a smaller “high” value as inputs. Then the function returns the function returns the value of the “flag” variable. The function “.length()” is used to get the number of characters in the user input using the “#include<string>” library. The function “.at()” is used to get each character of the user input assigned to the array “a[]” using the library “#include<string>”.

**The Main Program**

The main program asks the user for the input then assigns that input to the string variable “candi”. The program then calls the “.length()” function to get the number of characters and assign it to the “high” variable. The program uses a loop that has the value of “high” variable as its parameter where the array “a[]” is assigned each character of the user input using the “.at()” function. Then the program decrements the “high” variable to get rid of the extra empty value of the “high” variable. The program calls the “pal()” function and assigns its value to the “result” variable. Then the function checks the value of the “result” variable, if it is true the program will print “The string is a palindrome” , if it is false the program will print “The string is not a palindrome”.