|  |  |
| --- | --- |
| **Total Marks:** | **7.5** |
| **Obtained Marks:** |  |

**DATA STRUCTURE**

**AND**

**ALGORITHM**

**Lab Report # 07**

**Submitted To: Mam Tehreen**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Submitted By**: **Hammad Qureshi**  .

**Reg. Numbers: 2112114**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question no 1:**

**A palindrome is a string that reads the same forwards as backwards. Using only a fixed number of stacks and queues, the stack and queue ADT functions, and a fixed number of int and char variables, write a code to determine if a string is a palindrome. Assume that the string is read from  
standard input one character at a time. The code should output**

**true or false as appropriate.**

**Code:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  //Palindrome means a reserve of input string same as original  //if same then YES otherwise No  // Function that returns true  // if string is a palindrome  bool isPalindrome(string s)  {  int length = s.size();    // Creating a Stack  stack<char> st;    // Finding the mid  int i, mid = length / 2;    for (i = 0; i < mid; i++) {  st.push(s[i]);  }    // Checking if the length of the string  // is odd, if odd then neglect the  // middle character  if (length % 2 != 0) {  i++;  }    char ele;  // While not the end of the string  while (s[i] != '\0')  {  ele = st.top();  st.pop();    // If the characters differ then the  // given string is not a palindrome  if (ele != s[i])  return false;  i++;  }    return true;  }    // Driver code  int main()  {  string s = "madam";  cout<<" Input String " <<s<< " Palindrome Of String Is "<<endl;  if (isPalindrome(s)) {  cout << "Yes";  }  else {  cout << "No";  }    return 0;  } |

**CONSOLE SCREEN:**



