

|  |
| --- |
| **Title: DATA STRUCTURE AND ALGORITHM** |

**Name:** Faiza Iftikhar

**Roll no:** 200901043

**Section**: A

**Date of Experiment:** 12 October, 2021

**Lab:** 01

**Task # 01:**

Write a program to let the user enter a string of his own choice. Check whether the given string is a palindrome or not.

**CODE :**

class Stack: #defining stack

    def \_\_init\_\_(self): #returning all the elements of stack

        self.items = []

    def is\_empty(self): #Checking stack is empty or not

        return self.items == []

    def push(self, data): #pushing elements in the stack

        self.items.append(data)

    def pop(self): #popping elements from the stack

        return self.items.pop()

a = Stack()

String = input('Please enter the string: ') #Taking the string from the user

for ch in String: #Applying for loop

    a.push(ch) #Pushing every entered character in string

reversed\_String = ''

while not a.is\_empty(): #The loop will run if string is not empty

    reversed\_String = reversed\_String + a.pop() #Popping every charecter from "a" string and stored it on variable reversed\_string

if String == reversed\_String: #Now checking string is equal to reversed string or not

    print('The string is a palindrome.') #its palindrome if equal

else:

    print('The string is not a palindrome.') #Not palindrome if not equal

**CODE 2nd Method:**

String=input(("Please enter the string :")) #taking any string from user

reversed\_String=String[::-1] #Reversing the string and stored it on variable reversed\_String. Here

by colon and -1 it reversed the string

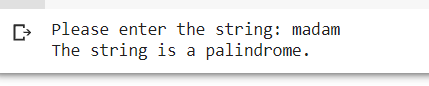
if(String==reversed\_String): #Camparing string and reversed string

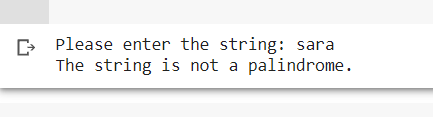
      print("The string is a palindrome") #Palindrome if it is equal

else:

      print("Not a palindrome") #Not a palindrome if its not equal

**Output:**





**Task # 02:**

Write a program to check the balanced parenthesis in the expression or not using stack

**CODE:**

from collections import deque

ex = input("Enter any expression: ") #Taking input from user

stack = deque()

output="Balanced paranthesis in expression"

count = 0 #defining counter

for character in ex: #Applying loop

    if (character == '(' or character == '{' or character == '['):

        stack.append(character) #adding characters in the stack

        count += 1 #counter will increment

    elif (character == ')' or character == '}' or character == ']'):

        if (len(stack) != 0): #checking if stack length is not zero

            if (character == ')'):

                if(stack[-1] == '('):

                    count -= 1 #then count will decrement

            if (character== ']'):

                if (stack[-1] == '['):

                    count -= 1 #count decrement

            if (character == '}'):

                if (stack[-1] == '{'):

                    count -= 1 #conter will decrement

            stack.pop()

        else:

            output= "unbalanced paranthesis in expression"

            break

if (count != 0): #if count is not equal to zero

    output= "unbalanced paranthesis in expression"

print (output)

**Output:**

