**Assignment- D09**

**Name of Student: Sumit Bhamare**

**Roll No.:08**

**Problem Statement:**

# A palindrome is a string of character that‘s the same forward and backward. Typically, punctuation, capitalization, and spaces are ignored. For example, ‖Poor Dan is in a droop‖ is a palindrome, as can be seen by examining the characters ―poor danisina droop‖ and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original- in a palindrome, the sequence will be identical. Write C++ program with functions-

# 1. To check whether given string is palindrome or not that uses a stack to determine whether a string is a palindrome.

# 2. To remove spaces and punctuation in string, convert all the Characters to lowercase, and then call above Palindrome checking function to check for a palindrome

# 3. To print string in reverse order using stack

**Program:**

#include<iostream>

#include<string.h>

#include<ctype.h>

using namespace std;

#define SIZE 50

class Stack

{

private:

char input\_str[SIZE];

char stack\_data[SIZE];

int top,length,char\_count;

public:

Stack() //constructor

{

top=-1;

length=0;

char\_count=0;

}

void push(char);

char pop();

void input\_data();

void check\_palindrome();

void get\_characters\_only();

void display\_original();

void display\_reverse();

};

void Stack::push(char temp)

{

if(top==SIZE-1)

{

cout<<"\n Stack Overflow!!!";

return;

}

top++;

stack\_data[top]=temp;

}

char Stack::pop()

{

if(top==-1)

{

cout<<"\n Stack Underflow!!!";

char ch='\n';

return ch;

}

char temp=stack\_data[top];

top--;

return temp;

}

void Stack::input\_data()

{

cout<<"\n Enter a String: ";

cin.getline(input\_str,SIZE);

length=strlen(input\_str);

}

void Stack::get\_characters\_only()

{

char temp[SIZE];

int i,j;

for(i=0; i<length; i++)

{

temp[i]=input\_str[i];

}

j=0;

for(i=0; i<length; i++ )

{

if(isalpha(temp[i]))

{

input\_str[j]=tolower(temp[i]);

j++;

}

}

length=j; //on removing spaces,punctuations

//new length is obtained

}

void Stack::check\_palindrome()

{

for(int i=0; i<length; i++)

push(input\_str[i]);

for(int i=0; i<length; i++)

{

if(input\_str[i]==pop())//finding a match

char\_count++;

}

if(char\_count==length)

cout<<"\n Given string is a Palindrome!!!\n";

else

cout<<"\n Given string is not a Palindrome!!!\n";

}

void Stack::display\_original()

{

for(int i=0; i<=length-1; i++)

cout<<input\_str[i];

}

void Stack::display\_reverse()

{

for(int i=length-1; i>=0; i--)

cout<<input\_str[i];

}

int main()

{

Stack obj;

obj.input\_data();

obj.get\_characters\_only();

cout<<"\n Displaying the original String: ";

obj.display\_original();

cout<<"\n Displaying the reversed string: ";

obj.display\_reverse();

cout<<"\n\t Checking for Palindrome....\n";

obj.check\_palindrome();

return 0;

}

**Output:**

