**#Assignment 10**

/\*

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-

a) To print original string followed by reversed string using stack

b) To check whether given string is palindrome or not

\*/

#include<iostream>

#include<string.h>

#define max 50

using namespace std;

class STACK

{

private:

char a[max];

int top;

public:

STACK()

{

top=-1;

}

void push(char);

void reverse();

void convert(char[]);

void palindrome();

};

void STACK::push(char c)

{

top++;

a[top] = c;

a[top+1]='\0';

cout<<endl<<c<<" is pushed on stack ...";

}

void STACK::reverse()

{

char str[max];

cout<<"\n\nReverse string is : ";

for(int i=top,j=0; i>=0; i--,j++)

{

cout<<a[i];

str[j]=a[i];

}

cout<<endl;

}

void STACK::convert(char str[])

{

int j,k,len = strlen(str);

for(j=0, k=0; j<len; j++)

{

if( ( (int)str[j] >= 97 && (int)str[j] <=122 ) || ( (int)str[j] >= 65 && (int)str[j] <=90 ))

{

if( (int)str[j] <=90 )

{

str[k] = (char)( (int)str[j] + 32 );

}else

{

str[k] = str[j];

}

k++;

}

}

str[k]='\0';

cout<<endl<<"Converted String : "<<str<<"\n";

}

void STACK::palindrome()

{

char str[max];

int i,j;

for(i=top,j=0; i>=0; i--,j++)

{

str[j]=a[i];

}

str[j]='\0';

if(strcmp(str,a) == 0)

cout<<"\n\nString is palindrome...";

else

cout<<"\n\nString is not palindrome...";

}

int main()

{

STACK stack;

char str[max];

int i=0;

cout<<"\nEnter string to be reversed and check is it palindrome or not : \n\n";

cin.getline(str , 50);

stack.convert(str);

while(str[i] != '\0')

{

stack.push(str[i]);

i++;

}

stack.palindrome();

stack.reverse();

}