Stack

#include"stack.h"

int main(){

char str[20];

cout<<"Enter the string:";

cin>>str;

int len=strlen(str);

Stack s(len);

//push ele in stack

for(int i=0;str[i]!='\0';i++){

s.push(str[i]);

}

//check palindrome

char ele;

int flag=0,i=0;

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

if(s.pop(ele)){

if(ele!=str[i]){

flag=1;

}

}

i++;

}

if(flag==1){

cout<<"string is not palindrome";

}

else{

cout<<"String is Palindrome";

}

return 0;

}

#include<iostream>

using namespace std;

class Stack{

int size;

int top;

char\* ptr;

public:

int getTop();

Stack(int);

bool isFull();

bool isEmpty();

bool push(char);

bool pop(char&);

};

#include "stack.h"

Stack::Stack(int s){

this->size=s;

this->top=-1;

this->ptr=new char[size];

}

bool Stack::isFull(){

if(top==size-1){

return true;

}

else{

return false;

}

}

bool Stack::isEmpty(){

if(top==-1){

return true;

}

else{

return false;

}

}

bool Stack::push(char ch) {

if(isFull()){

cout<<"OverFlow";

return false;

}

else{

ptr[++top]=ch;

return true;

}

}

bool Stack::pop(char& ch){

if(isEmpty()){

cout<<"Underflow!!";

return false;

}

else{

ch=ptr[top--];

}

}

int Stack::getTop(){

return this->top;

}