DSA Lab06

23K2001

M.Muzammil Siddiqui

BCS-3J

Q1:

//23K2001 - Muzammil

#include<iostream>

using namespace std;

class stacks{

    private:

        int top,size;

    public:

        char \*arr;

        stacks():top(-1),size(0),arr(nullptr){}

        stacks(int s):top(-1),size(s){

            arr = new char[size];

            for(int i=0;i<size;i++)

                arr[i]='!';

        }

        void push(char e){

            if(top>=(size-1)){

                cout<<"Stack overflow occured!"<<endl;

                return;

            }

            arr[++top] = e;

        }

        char pop(){

            if(top<0){

                cout<<"Stacks underflow occured!"<<endl;

                return '!';

            }

            char last = arr[top--];

            return last;

        }

        bool checkPalindrome(){

            stacks reversed(this->size);

            for(int i=0;i<size;i++)

                reversed.push(this->arr[i]);

            for(int i=0;i<size;i++){

                if(this->arr[i]!=reversed.pop())

                    return false;

            }

            return true;

        }

        ~stacks(){ delete[] arr; }

};

int main(){

    int s;

    cout<<"Enter size of stack: ";

    cin>>s;

    stacks flex(s);

    char e;

    cout<<"Enter "<<s<<" elements:"<<endl;

    for(int i=0;i<s;i++){

        cin>>e;

        flex.push(e);

    }

    cout<<endl<<"Checking for Palindrome:"<<endl;

    if(flex.checkPalindrome())

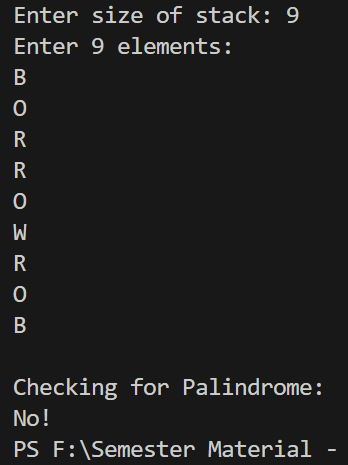
        cout<<"Yes!"<<endl;

    else

        cout<<"No!"<<endl;

    return 0;

}



A screen shot of a computer

Description automatically generated

Q2:

//23K2001 - Muzammil

#include<iostream>

using namespace std;

class stacks{

    private:

        int top,size;

    public:

        string \*arr;

        stacks():top(-1),size(0),arr(nullptr){}

        stacks(int s):top(-1),size(s){

            arr = new string[size];

            for(int i=0;i<size;i++)

                arr[i]="!";

        }

        void push(string e){

            if(top>=(size-1)){

                cout<<"Stack overflow occured!"<<endl;

                return;

            }

            arr[++top] = e;

        }

        string pop(){

            if(top<0){

                cout<<"Stacks underflow occured!"<<endl;

                return "!";

            }

            string last = arr[top--];

            return last;

        }

        bool checkPalindrome(){

            stacks reversed(this->size);

            for(int i=0;i<size;i++)

                reversed.push(this->arr[i]);

            for(int i=0;i<size;i++){

                if(this->arr[i]!=reversed.pop())

                    return false;

            }

            return true;

        }

        bool isEmpty(){

            if(top<0)

                return true;

            return false;

        }

        ~stacks(){ delete[] arr; }

};

int main(){

    int s;

    cout<<"Enter to-do list size: ";

    cin>>s;

    stacks flex(s);

    string e;

    cout<<"Enter "<<s-1<<" tasks:"<<endl;

    for(int i=0;i<s-1;i++){

        cin>>e;

        flex.push(e);

    }

    cout<<endl<<"Enter a task to add to top:"<<endl;

    cin>>e;

    flex.push(e);

    cout<<endl<<"Checking if stack is empty:"<<endl;

    if(flex.isEmpty())

        cout<<"Yes!"<<endl;

    else

        cout<<"No!"<<endl;

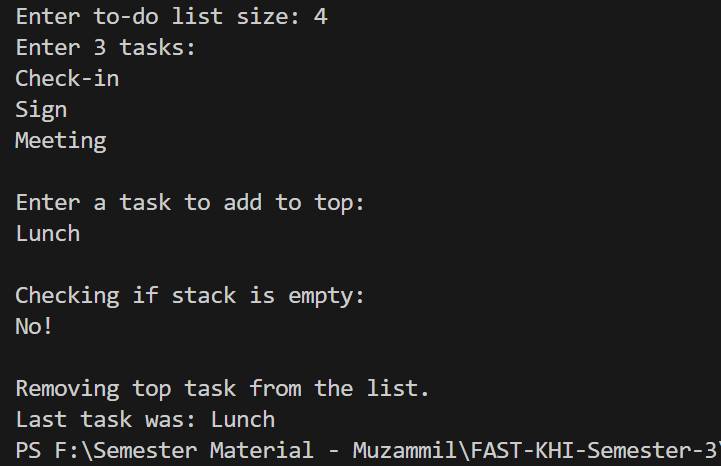
    cout<<endl<<"Removing top task from the list."<<endl;

    e = flex.pop();

    cout<<"Last task was: "<<e<<endl;

    return 0;

}



Q3: