**ASSIGNMENT 4**

**QUESTION 1**

#include<bits/stdc++.h>

using namespace std;

#define size 100

class Stack{

    public:

int arr[size];

int top;

Stack(){

 top=-1;

}

void push(int x){

    if(top==(size-1)) cout<<"stack overflow"<<endl;

    else{

        top++;

        arr[top]=x;

    }

}

void pop(){

    if(top==-1) cout<<"empty stack"<<endl;

    else{

        top--;

    }

}

void peek(){

    if(top==-1) cout<<"empty stack"<<endl;

    else{

        cout<<"the top element is: "<<arr[top]<<endl;

    }

}

void display(){

    if(top==-1) cout<<"empty stack"<<endl;

    else{

        for(int i=top;i>=0;i--){

            cout<<arr[i]<<endl;

        }

    }

}

bool isfull(){

    return (top==size-1);

}

bool isempty(){

    return (top==-1);

}

};

int main(){

int choice , value;

Stack s;

cout<<"1. push"<<endl<<"2. pop"<<endl<<"3. peek"<<endl<<"4. display"<<endl<<"5. isfull"<<endl<<"6. isempty"<<endl;

cout<<"enter a choice: ";

cin>>choice;

switch(choice){

    case 1:

    cout<<"enter a value: ";

    cin>>value;

    s.push(value);

    break;

    case 2:

cout<<"the element is popped";

s.pop();

break;

    case 3:

cout<<"the topmost element is: ";

s.peek();

break;

case 4:

s.display();

break;

case 5:

cout << (s.isempty() ? "Stack is Empty" : "Not Empty") << endl;

break;

     case 6:

        cout << (s.isfull() ? "Stack is Full" : "Not Full") << endl;

        break;

        deafult:

        cout<<"invalid choice!";

}

return 0;

}

**QUESTION 2**

#include<bits/stdc++.h>

using namespace std;

#define size 100

class reversestring{

    public:

    char arr[size];

    int top;

    reversestring(){

        top=-1;

    }

    void push(int x){

        if(top==size-1) cout<<"stack overflow";

        else{

            top++;

            arr[top]=x;

        }

    }

    char pop(){

       if(top==-1) cout<<"stack empty";

        else{

            return arr[top--];

        }

    }

bool isempty(){

    return(top==-1);

}

};

int main(){

string s;

cout<<"enter the string: ";

getline(cin,s);

reversestring r;

for(int i=0;i<s.length();i++){

    r.push(s[i]);

}

while(!r.isempty()){

cout<<r.pop();

}

return 0;

}

**QUESTION 3**

#include<bits/stdc++.h>

using namespace std;

#define size 100

class parenthesis{

   public:

    char arr[size];

    int top;

    parenthesis(){

        top=-1;

    }

    void push(char x){

        if(top==size-1) cout<<"stack overflow";

        else{

            top++;

            arr[top]=x;

        }

    }

    char pop(){

       if(top==-1) cout<<"stack empty";

        else{

            return arr[top--];

        }

    }

    char peek(){

        if(top==-1) cout<<"stack overflow";

        else return arr[top];

    }

bool isempty(){

    return(top==-1);

}

};

bool isbalanced(string expr){

    parenthesis p;

    for(int i=0;i<expr.length();i++){

        if (expr[i] == '(' || expr[i] == '[' || expr[i] == '{') {

        p.push(expr[i]);

    }

    else{

        if(p.isempty()) return false;

        char ch=p.peek();

        p.pop();

        if ((expr[i] == ')' && ch != '(') ||

                (expr[i] == ']' && ch != '[') ||

                (expr[i] == '}' && ch != '{')) {

                return false;

            }

    }

}

}

int main(){

string expr;

    cout << "Enter an expression: ";

    getline(cin, expr);

if(isbalanced(expr)) cout<<"the string is balanced";

else cout<<"not balanced";

return 0;

}

**QUESTION 5**

#include <bits/stdc++.h>

using namespace std;

#define SIZE 100

class IntStack {

public:

int arr[SIZE];

int top;

IntStack() { top = -1; }

void push(int x) {

if(top == SIZE-1) cout << "Stack Overflow!" << endl;

else arr[++top] = x;

}

int pop() {

if(top == -1) return 0;

else return arr[top--];

}

bool isEmpty() { return top == -1; }

};

int evaluatePostfix(string postfix) {

IntStack s;

for(int i = 0; i < postfix.length(); i++) {

char ch = postfix[i];

if(ch >= '0' && ch <= '9') {

s.push(ch - '0');

} else { // operator

int b = s.pop();

int a = s.pop();

switch(ch) {

case '+': s.push(a+b); break;

case '-': s.push(a-b); break;

case '\*': s.push(a\*b); break;

case '/': s.push(a/b); break;

}

}

}

return s.pop();

}

int main() {

string postfix;

cout << "Enter a postfix expression (single-digit operands): ";

getline(cin, postfix);

int result = evaluatePostfix(postfix);

cout << "Result: " << result << endl;

return 0;

}