NAME: MOAZZAM FAROOQUI

ROLL NO: CT-24068

COURSE CODE: CT-159

ASSIGNMENT: DSA PRACTICE SESSION TASK#3

Q1. Valid Parentheses

SOURCE CODE:

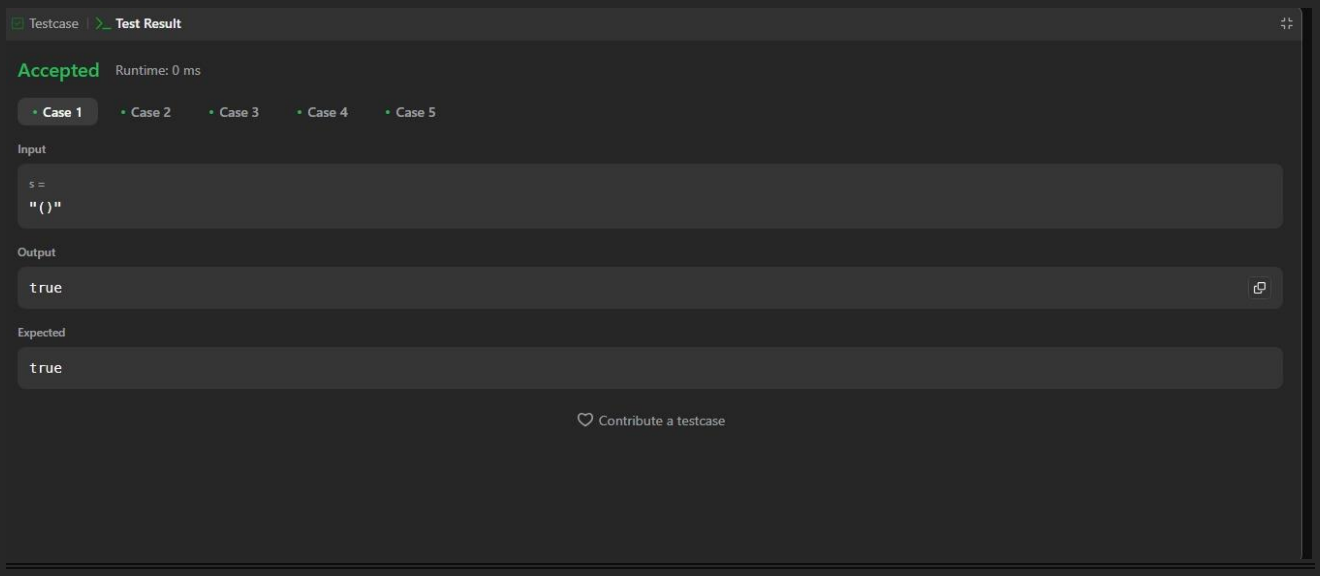
Code

C++ Auto

```cpp
class Solution {
public:
    bool isValid(string s){
        stack<char> st;
        for(char c:s){
            if(c=='(' || c=='{' || c=='['){
                st.push(c);
            }
            else{
                if(st.empty()) return false;
                char top=st.top();
                st.pop();
                if((c==')' && top!='(') ||
                   (c=='}' && top!='{') ||
                   (c==']' && top!='[')) {
                    return false;
                }
            }
        }
        return st.empty();
    }
};
```

Saved Ln 1, Col 1

ACCEPTANCE STATUS:



Q2. Evaluate Reverse Polish Notation

SOURCE CODE:

Code

C++ Auto

```cpp
class Solution {
public:
    int evalRPN(vector<string>& tokens) {
        stack<int>st;
        for(string t:tokens){
           if(t=="+" || t=="-" || t=="*" || t=="/"){
               int b=st.top(); st.pop();
               int a=st.top(); st.pop();

               if(t=="+") st.push(a+b);
               else if(t=="-") st.push(a-b);
               else if(t=="*") st.push(a*b);
               else if(t=="/") st.push(a/b);
           }
           else{
               st.push(stoi(t));
           }
        }
           return st.top();
        }
};
```

Saved Ln 1, Col 1

ACCEPTANCE STATUS:

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

tokens =

["2","1","+","3","*"]

Output

9

Expected

9

Contribute a testcase

Q3. Baseball Game

SOURCE CODE:

Code

C++ Auto

```cpp
class Solution {
public:
    int calPoints(vector<string>& operations) {
        stack<int>st;
        for(string op:operations){
            if(op=="C"){
                st.pop();

            }
            else if(op=="D"){
                st.push(st.top()*2);
            }
            else if(op=="+"){
                int a=st.top();st.pop();
                int b=st.top();
                st.push(a);
                st.push(a+b);
            }
            else{
                st.push(stoi(op));
            }
        int sum=0;
        while(!st.empty()){
            sum+=st.top();
            st.pop();
        }
        return sum;
        }
};
```

ACCEPTANCE STATUS:

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

operations =
["5","2","C","D","+"]

Output

30

Expected

30

Contribute a testcase

Q4. Remove All Adjacent Duplicates in String

SOURCE CODE:

Code

C++ Auto

```cpp
class Solution {
public:
    string removeDuplicates(string s) {
        string result="";
        int i=0;
        while(i<s.size()){
            char c=s[i];
            if(!result.empty() && result.back()==c){
                result.pop_back();
            }
            else{
                result.push_back(c);
            }
            i++;
        }
        return result;
    }
};
```

Saved Ln 1, Col 1

ACCEPTANCE STATUS:

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

s =
"abbaca"

Output

"ca"

Expected

"ca"

Contribute a testcase