# Worksheet - 5

**NAME: AKSHAT CHAUDHARY** **SUBJECT: IT SKILLS**

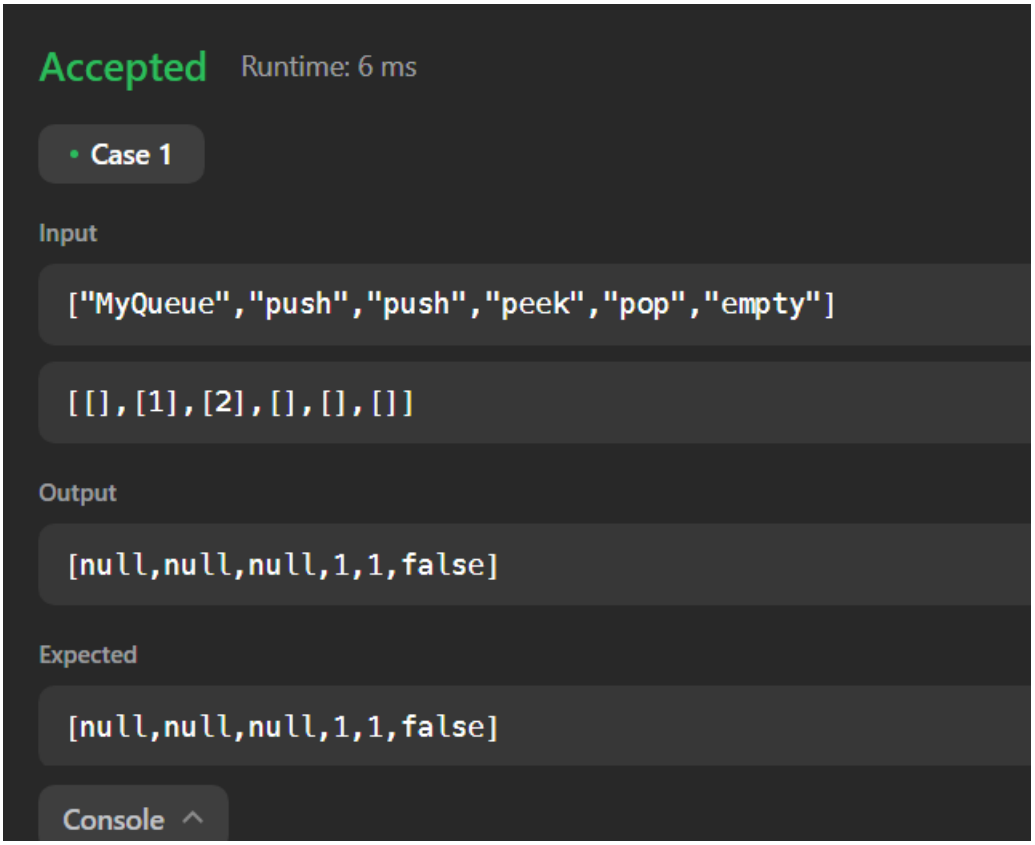
**UID: 20BCS5751** **Date: 09/01/2023**

**Que-1: Implement Queue using Stacks**

**Code:**

```
class MyQueue {
    stack<int> s;
public:
    MyQueue() {
    }

    void push(int x) {
        if(s.empty())
            s.push(x);
        else{
            stack<int>temp;
            while(!s.empty()){
                temp.push(s.top());
                s.pop();
            }
            s.push(x);
            while(!temp.empty()){
                s.push(temp.top());
                temp.pop();
            }
        }
    }

    int pop() {
        int ans = s.top();
        s.pop();
        return ans;
    }

    int peek() {
        return s.top();
    }

    bool empty() {
        return s.empty();
    }
};
```

**Output:**



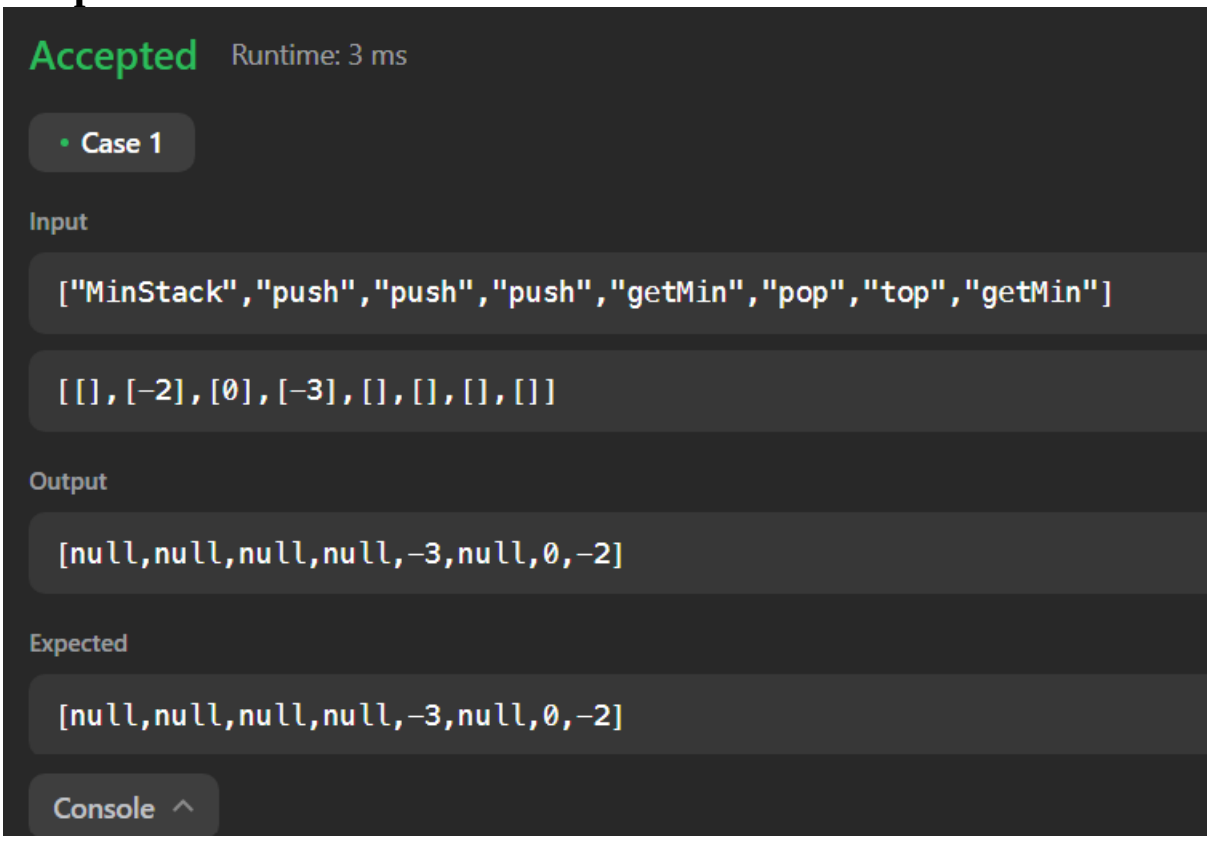

**Que-2: Min Stack**

**Code:**

```
class MinStack {
public:
    ListNode *head;
    ListNode *min;
    MinStack() {
        head=nullptr;
    }
    void push(int val) {
        if(!head){
            head= new ListNode(val);
            min=new ListNode(val);
        }
        else{
            ListNode *temp=new ListNode(val);
            temp->next=head;
            head=temp;
            if(val<=min->val){
```

```
                ListNode *tempmin=new ListNode(val);
                tempmin->next=min;
                min=tempmin;
            }
        }
    }

    void pop() {
        if(head->val==min->val)
            min=min->next;
        head=head->next;
    }

    int top() {
        return head->val;
    }

    int getMin() {
        return min->val;
    }
};
```

**Output:**

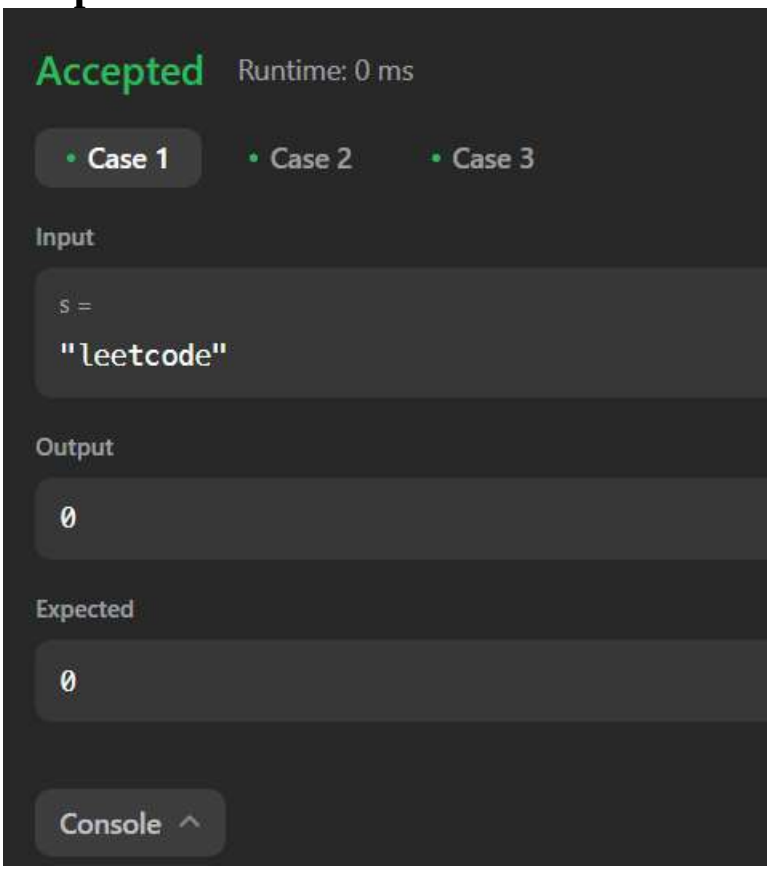


## Que-3: First Unique Character in a String

### Code:

```
class Solution {
public:
    int firstUniqChar(string s) {
        unordered_map<char,int> map;
        for(int i=0;i<s.size();i++){
            map[s[i]]++;
        }
        for(int i=0;i<s.size();i++){
            if(map[s[i]]==1){
                return i;
            }
        }
        return -1;
    }
};
```

### Output:

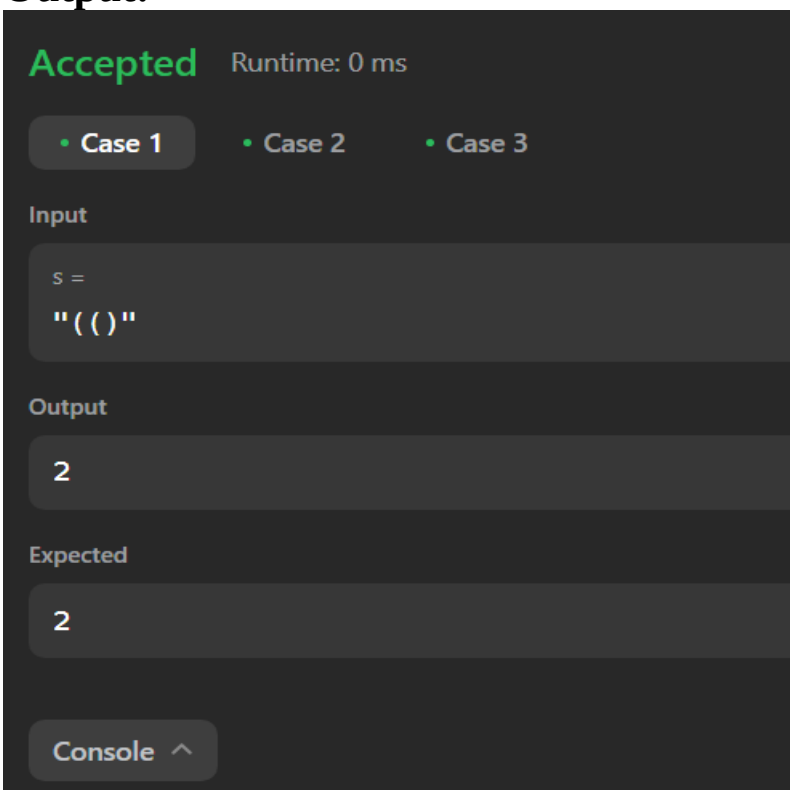


## Que-4: Longest Valid Parentheses

**Code:**

```
class Solution {
public:
    int longestValidParentheses(string s) {
        stack<int> st;
        st.push(-1);
        int ans=0;
        for(int i=0;i<s.size();i++){
            if(s[i]=='('){
                st.push(i);
            }
            else{
                st.pop();
                if(st.empty()){
                    st.push(i);
                }
                else{
                    if(ans<i - st.top())
                        ans = i - st.top();
                }
            }
        }
        return ans;
    }
};
```

**Output:**



## Que-5: Validate Stack Sequences

### Code:

```
class Solution {
public:
    bool validateStackSequences(vector<int>& pushed, vector<int>& popped) {
        stack<int> st;
        int j = 0;
        for(auto val : pushed){
            st.push(val);
            while(st.size() > 0 && st.top() == popped[j]){
                st.pop();
                j++;
            }
        }
        return st.size() == 0;
    }
};
```

### Output: