



Worksheet - 5

NAME: Gautam Kumar SEC-DWWC 43

UID: 20BCS9807 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:

```
Accepted Runtime: 6 ms

• Case 1

Input

["MyQueue", "push", "push", "peek", "pop", "empty"]

[[],[1],[2],[],[],[]]

Output

[null,null,null,1,1,false]

Expected

[null,null,null,1,1,false]
```

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){
```







```
Accepted Runtime: 3 ms

• Case 1

Input

["MinStack", "push", "push", "getMin", "pop", "top", "getMin"]

[[], [-2], [0], [-3], [], [], []]

Output

[null, null, null, -3, null, 0, -2]

Expected

[null, null, null, null, -3, null, 0, -2]
```



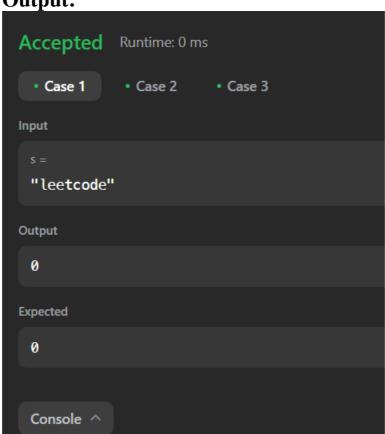




Que-3: First Unique Character in a String

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;
    }
};</pre>
```





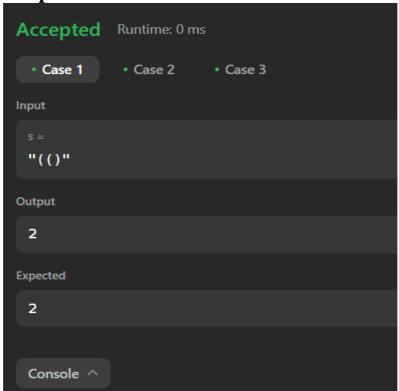




Que-4: Longest Valid Parentheses

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++){</pre>
             if(s[i]=='('){
                 st.push(i);
             else{
                 st.pop();
                 if(st.empty()){
                     st.push(i);
                 else{
                     if(ans<i - st.top())</pre>
                          ans = i - st.top();
                 }
             }
        return ans;
```









Que-5: Validate Stack Sequences

Code:

```
Accepted Runtime: 0 ms

• Case 1
• Case 2

Input

pushed =
[1,2,3,4,5]

popped =
[4,5,3,2,1]

Output

true

Expected

Console ^
```