

# WORKSHEET\_5

## 1. Product of the Last K Numbers

```
vector<int> products={1};

ProductOfNumbers() {

}

void add(int num) {
    if(num==0) products={1};
    else{
        if(products.size()==1) products.push_back(num);
        else{
            products.push_back(products.back()*num);
        }
    }
}

int getProduct(int k) {
    if (k >= products.size()) return 0;
    return products.back() / products[products.size() - k - 1];
}
```

The screenshot shows a web browser window with the LeetCode website. The page displays the submission status for the problem "Product of the Last K Numbers" (LeetCode 1352). The status is "Accepted". The code is written in C++ and is shown in the editor on the right side of the page. The code defines a class `ProductOfNumbers` with a `vector<int> products` and methods `add` and `getProduct`. The `add` method adds a number to the `products` vector, and the `getProduct` method returns the product of the last `k` numbers. The code is as follows:

```
1 class ProductOfNumbers {
2 public:
3     vector<int> products={1};
4     ProductOfNumbers() {
5     }
6
7
8     void add(int num) {
9         if(num==0) products={1};
10        else{
11            if(products.size()==1) products.push_back(num);
12            else{
13                products.push_back(products.back()*num);
14            }
15        }
16    }
17
18    int getProduct(int k) {
19        if (k >= products.size()) return 0;
20        return products.back() / products[products.size() - k - 1];
21    }
22 };
23
24 /**
25  * Your ProductOfNumbers object will be instantiated and called as such:
26  * ProductOfNumbers* obj = new ProductOfNumbers();
27  * obj->add(num);
28  * int param_2 = obj->getProduct(k);
29  */
```

At the bottom of the page, there is a feedback section with a question: "Quick question! How satisfied are you with our new design?". It includes five emoji buttons: a sad face, a neutral face, a happy face, a very happy face, and a fire emoji. The feedback section also shows "Very Dissatisfied" and "Very Satisfied" labels.

## 2. Book Exercises

```
#include <iostream>
#include<bits/stdc++.h>
using namespace std;

int main() {
ios_base::sync_with_stdio(false);
cin.tie(NULL);
    int n;
    cin>>n;
    stack<pair<pair<int,string>,int>> st1;
    for(int i=0;i<n;i++){
        int x;
        cin>>x;
        string str;
        if(x!=-1 ){
            cin>>str;
        }else if(x==-1){
            cout<<st1.top().second<<" "<<st1.top().first.second<<endl;
            st1.pop();
            continue;
        }
        if (x==0){
            continue;
        }
        if(!st1.empty() and st1.top().first.first<x ){
            st1.top().second++;
            continue;
        }

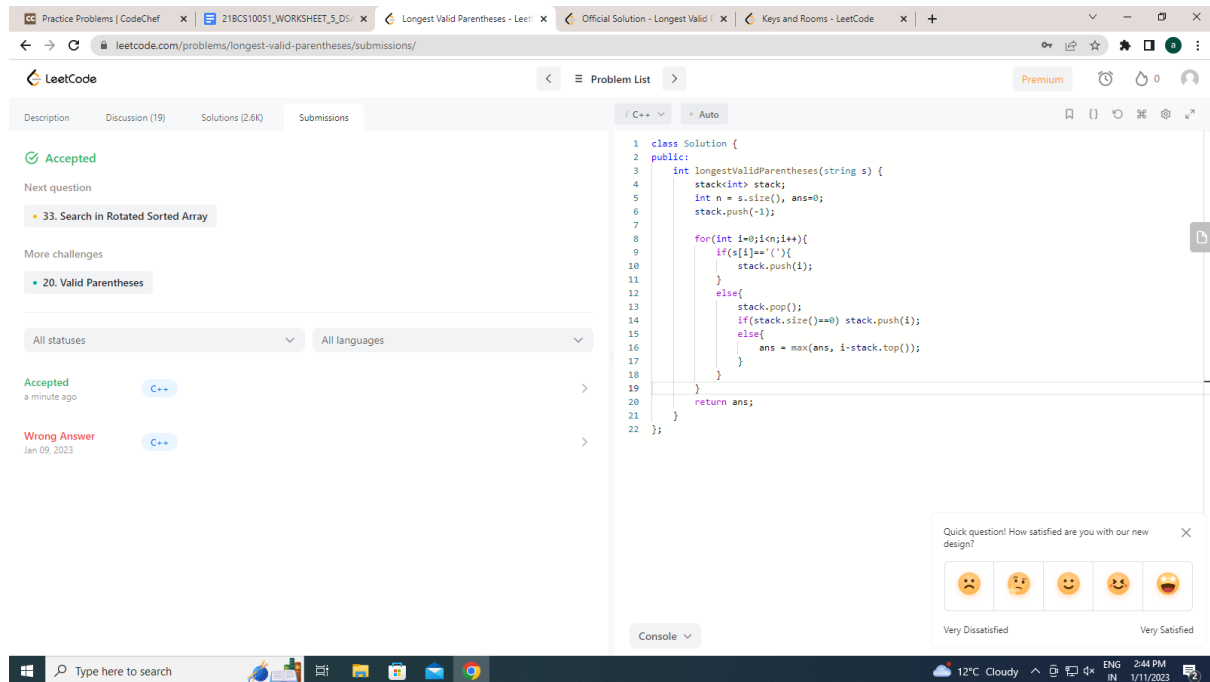
        pair<int,string> p = make_pair(x,str);
        pair<pair<int,string>,int> whole_pair= make_pair(p,0);
        st1.push(whole_pair);
    }
    return 0;
}
```

A screenshot of a web browser showing a CodeChef problem page titled "Book Exercises". The page has a dark theme. At the top, there are browser tabs for "Practice Problems | CodeChef", "Tennis Tournament - Problems |", "Untitled document - Google Doc...", "Book Exercises - Problems | Code...", and "Firewall Authentication Keepalive". The URL bar shows "codechef.com/problems/BDX?tab=statement". The problem page includes a "Statement" tab, a "Hints" tab, a "Submissions" tab, a "Solution" tab, and an "Ask a Doubt" tab. The "Statement" tab is active, showing the problem description. The problem description states: "Harry is a bright student. To prepare thoroughly for exams, he completes all the exercises in his book. Now that the exams are approaching fast, he is doing book exercises day and night. He writes down and keeps updating the remaining number of exercises on the back cover of each book. Harry has a lot of books messed on the floor. Therefore, he wants to pile up the books that still have some remaining exercises into a single pile. He will grab the books one-by-one and add the books that still have remaining exercises to the top of the pile. Whenever he wants to do a book exercise, he will pick the book with the minimum number of remaining exercises from the pile. In order to pick the book, he has to remove all the books above it. Therefore, if there are more than one books with the minimum number of remaining exercises, he will take the one which requires the least number of books to remove. The removed books are returned to the messy floor. After he picks the book, he will do all the remaining exercises and trash the book. Since number of books is rather large, he needs your help to tell him the number of books he must remove, for picking the book with the minimum number of exercises. Note that more than one book can have the same name." The "Input" section states: "The first line contains a single integer N denoting the number of actions. Then N lines follow. Each line starts with an integer. If the integer is -1, that means Harry wants to do a book exercise. Otherwise, the integer is number of the remaining exercises in the book he grabs next. This is followed by a string". The "Test against Custom Input" section shows a list of inputs: 6, 9 english, 6 mathematics, 8 geography. The "Problem Solver Badge" section shows a progress bar at 63 / 250, with a note "Solve 187 more problems to get Silver Badge". The "Status" section shows "Status: ✓ Correct Answer" and "Submission ID: 84848441". The "Time" section shows "Time: 0.78s". The "Congratulatory message" says "Congratulations on solving the problem. Visit our practice section to solve more interesting problems." and "View another problem". The "Compile &amp; Run" and "Submit Code" buttons are visible. The bottom of the page shows a Windows taskbar with a search bar, task icons, and system tray information: 13°C Fog, 3:45 PM, 1/9/2023.

## 32. Longest Valid Parentheses

```
class Solution {
public:
    int longestValidParentheses(string s) {
        stack<int> stack;
        int n = s.size(), ans=0;
        stack.push(-1);

        for(int i=0;i<n;i++){
            if(s[i]=='('){
                stack.push(i);
            }
            else{
                stack.pop();
                if(stack.size()==0) stack.push(i);
                else{
                    ans = max(ans, i-stack.top());
                }
            }
        }
        return ans;
    }
};
```



## Min Stack

```
class MinStack
```

```
{
```

```
    Stack<Long> stack = new Stack<>();  
    Long mini;
```

```
    public MinStack()
```

```
    {
```

```
        mini = Long.MAX_VALUE;
```

```
    }
```

```
    public void push(int val)
```

```
    {
```

```
        Long value = Long.valueOf(val);
```

```
        if(stack.isEmpty())
```

```
        {
```

```
            mini = value;
```

```
            stack.push(value);
```

```
        }
```

```
        else
```

```
        {
```

```
            if(value < mini)
```

```
            {
```

```
                stack.push(2 * value - mini);
```

```
                mini = value;
```

```

    }
    else
    {
        stack.push(value);
    }
}

public void pop()
{
    if(stack.isEmpty())
    {
        return;
    }

    Long value = stack.pop();
    if(value < mini)
    {
        mini = 2 * mini - value;
    }
}

public int top()
{
    Long value = stack.peek();
    if(value < mini)
    {
        return mini.intValue();
    }
    return value.intValue();
}

public int getMin()
{
    return mini.intValue();
}
}

```

## . Implement Queue using Stacks

WhatsApp x 21BCS10051\_ASHISH\_VID x WORKSHEET\_7\_DSA - Go x 21BCS10051\_WORKSHEET x Implement Queue using Stacks x

leetcode.com/problems/implement-queue-using-stacks/submissions/

WhatsApp Web Numerical\_OT\_46\_A... Chandigarh Univers... Folder - Google Dri... Lapindromes Codec... Lecture Materials |... DBMS MCQ (Multip... Chandigarh Univers...

LeetCode Problem List Premium 0

Description Discussion (26) Solutions (3.4K) Submissions

Accepted

Next question

- 233. Number of Digit One

More challenges

- 225. Implement Stack using Queues

All statuses All languages

Accepted a few seconds ago C++

```
1 class MyQueue {
2     private:
3         stack<int> stack1;
4         stack<int> stack2;
5
6     public:
7         MyQueue()
8         {
9
10        }
11
12        void push(int x)
13        {
14            while(!stack1.empty())
15            {
16                stack2.push(stack1.top());
17                stack1.pop();
18            }
19            stack2.push(x);
20
21            while(!stack2.empty())
22            {
23                stack1.push(stack2.top());
24                stack2.pop();
25            }
26        }
27    }
```

Console Run Submit

Search 12°C Cloudy ENG 07:11 PM

## First Unique Character in a String

WhatsApp x 21BCS10051\_ASHISH\_VID x WORKSHEET\_7\_DSA - Go x 21BCS10051\_WORKSHEET x First Unique Character in a String x

leetcode.com/problems/first-unique-character-in-a-string/submissions/

WhatsApp Web Numerical\_OT\_46\_A... Chandigarh Univers... Folder - Google Dri... Lapindromes Codec... Lecture Materials |... DBMS MCQ (Multip... Chandigarh Univers...

LeetCode Problem List Premium 0

Description Discussion (15) Solutions (6.2K) Submissions

Accepted

Next question

- 388. Longest Absolute File Path

More challenges

- 2351. First Letter to Appear Twice

All statuses All languages

Accepted a few seconds ago C++

```
1 class Solution {
2     public:
3         int firstUniqChar(string s) {
4             unordered_map<char, int> mp;
5             for(int i=0; i<s.size(); i++){
6                 mp[s[i]]++;
7             }
8             for(int i=0; i<s.size(); i++){
9                 if(mp[s[i]]==1) return i;
10            }
11            return -1;
12        }
13    };
```

Console Run Submit

Search 12°C Cloudy ENG 07:17 PM