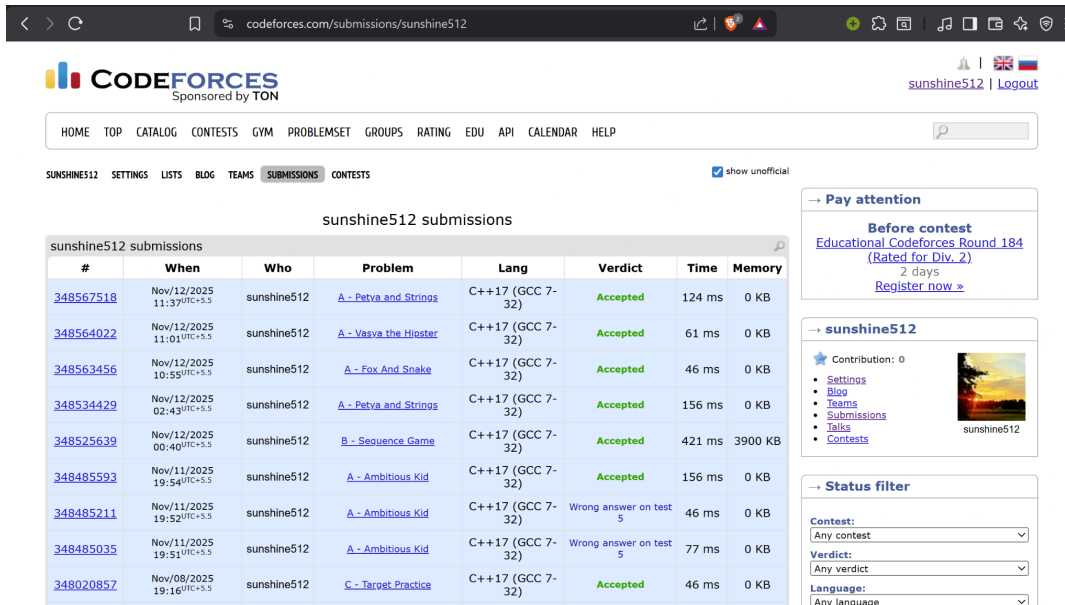


Assignment 1 - CP Basics

Submitted by: Vandana Kumari

Enroll No: 21501172024

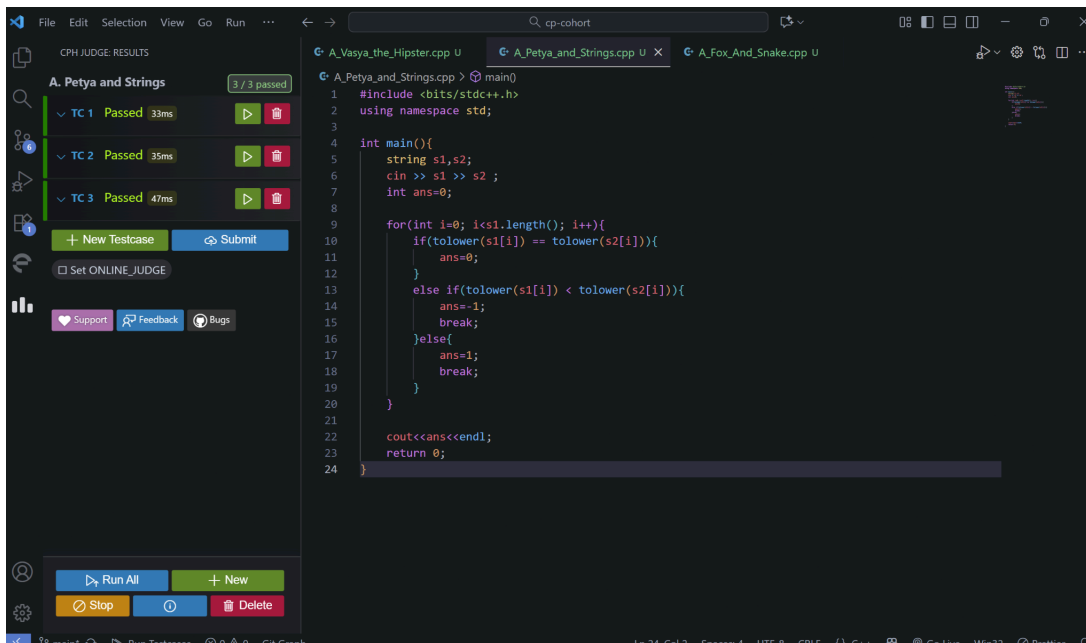
Q1, Q2, Q3 codes accepted in codeforces



The screenshot shows the Codeforces website with the user's submission history for the contest 'sunshine512'. The table lists 10 submissions, all of which were accepted. The problems solved include 'A - Petya and Strings', 'A - Vasya the Hipster', 'A - Fox And Snake', 'A - Petya and Strings', 'B - Sequence Game', 'A - Ambitious Kid', and 'C - Target Practice'. The right sidebar contains links to the contest page and a status filter.

#	When	Who	Problem	Lang	Verdict	Time	Memory
348567518	Nov/12/2025 11:37 UTC+5.5	sunshine512	A - Petya and Strings	C++17 (GCC 7-32)	Accepted	124 ms	0 KB
348564022	Nov/12/2025 11:01 UTC+5.5	sunshine512	A - Vasya the Hipster	C++17 (GCC 7-32)	Accepted	61 ms	0 KB
348563456	Nov/12/2025 10:55 UTC+5.5	sunshine512	A - Fox And Snake	C++17 (GCC 7-32)	Accepted	46 ms	0 KB
348534429	Nov/12/2025 02:43 UTC+5.5	sunshine512	A - Petya and Strings	C++17 (GCC 7-32)	Accepted	156 ms	0 KB
348525639	Nov/12/2025 00:40 UTC+5.5	sunshine512	B - Sequence Game	C++17 (GCC 7-32)	Accepted	421 ms	3900 KB
348485593	Nov/11/2025 19:54 UTC+5.5	sunshine512	A - Ambitious Kid	C++17 (GCC 7-32)	Accepted	156 ms	0 KB
348485211	Nov/11/2025 19:52 UTC+5.5	sunshine512	A - Ambitious Kid	C++17 (GCC 7-32)	Wrong answer on test 5	46 ms	0 KB
348485035	Nov/11/2025 19:51 UTC+5.5	sunshine512	A - Ambitious Kid	C++17 (GCC 7-32)	Wrong answer on test 5	77 ms	0 KB
348020857	Nov/08/2025 19:16 UTC+5.5	sunshine512	C - Target Practice	C++17 (GCC 7-32)	Accepted	46 ms	0 KB

Q1)



The screenshot shows a C++ code editor with the solution for problem 'A - Petya and Strings'. The code implements a function to compare two strings, s1 and s2, based on their ASCII values. It uses a loop to iterate through the characters of s1 and s2, comparing them and updating the answer accordingly. The code is compiled and run successfully, as indicated by the 'Run All' button and the '3 / 3 passed' status.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     string s1,s2;
6     cin >> s1 >> s2 ;
7     int ans=0;
8
9     for(int i=0; i<s1.length(); i++){
10         if(tolower(s1[i]) == tolower(s2[i])){
11             ans=0;
12         }
13         else if(tolower(s1[i]) < tolower(s2[i])){
14             ans=-1;
15             break;
16         }else{
17             ans=1;
18             break;
19         }
20     }
21
22     cout<<ans<<endl;
23     return 0;
24 }
```

Q2)

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     int n,m;
6     cin>>n>>m;
7
8     for(int i=0; i<n; i++){
9         string s(m, '#');
10        for(int j=0; j<m; j++){
11            if(i%2!=0){
12                if(i==3 || i==7 || i==11 || i==15 || i==19 || i==23 || i==27 || i==31 ||
13                   i==35 || i==39 || i==43 || i==47){
14                    if(j!=0){
15                        s[j]='.';
16                    }
17                }else{
18                    if(j!=m-1){
19                        s[j]='.';
20                    }
21                }
22            }
23        }
24        cout<<s<<endl;
25    }
26    return 0;
27 }
```

Q3)

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     int a,b;
6     cin>>a>>b;
7
8     cout<<min(a,b)<<" ";
9     cout<<abs(a-b)/2<<endl;
10    return 0;
11 }
12
```

Q4)

Stack < > ↕

Description | Accepted × | Editorial | Solutions | Submit

← All Submissions

Accepted 99 / 99 testcases passed

Vandana submitted at Nov 12, 2025 11:47

Solution

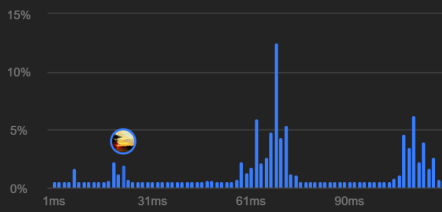
Runtime

22 ms | Beats 93.63%

Analyze Complexity

Memory

81.56 MB | Beats 5.76%



Code

Java Auto

```
1 class Solution {
2     public int largestRectangleArea(int[] heights) {
3         Deque<Integer> st = new ArrayDeque<>();
4         int maxArea=0;
5         int ele=0, nse=0, pse= 0;
6         for(int i=0; i<heights.length; i++){
7             while(!st.isEmpty() && heights[st.peek()]>heights[i]){
8                 ele=st.pop();
9                 nse=i;
10                pse=st.isEmpty() ? -1: st.peek();
11                maxArea= Math.max(maxArea, heights[ele] * (nse-pse-1));
12            }
13            st.push(i); //ham index ko push krenge
14        }
15
16        while(!st.isEmpty()){
17            nse=heights.length;
18            ele=st.pop();
19            pse= st.isEmpty() ? -1: st.peek();
20
21            maxArea= Math.max(maxArea, heights[ele] * (nse-pse-1));
22        }
23
24        return maxArea;
25    }
```

Saved

Testcase

Test Result