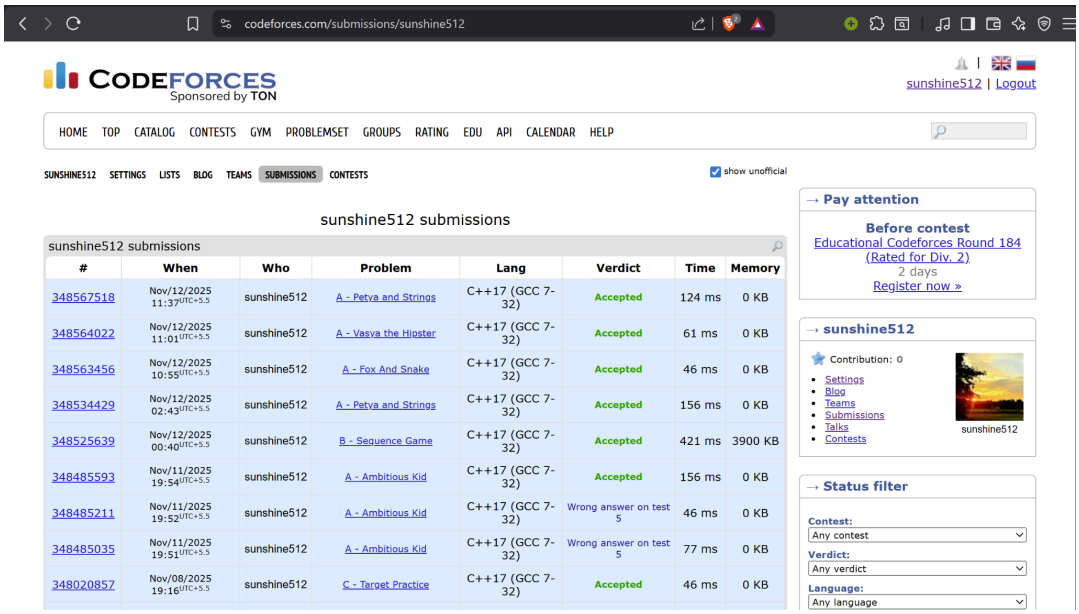
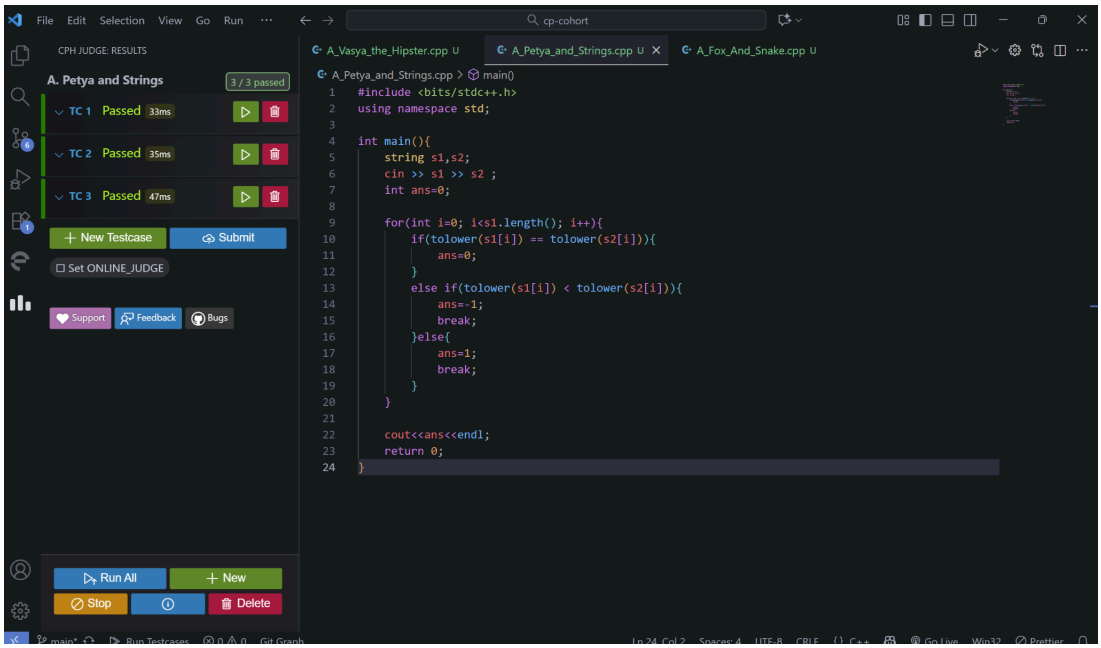
Assignment 1 - CP Basics

Submitted by: Vandana Kumari
Enroll No: 21501172024

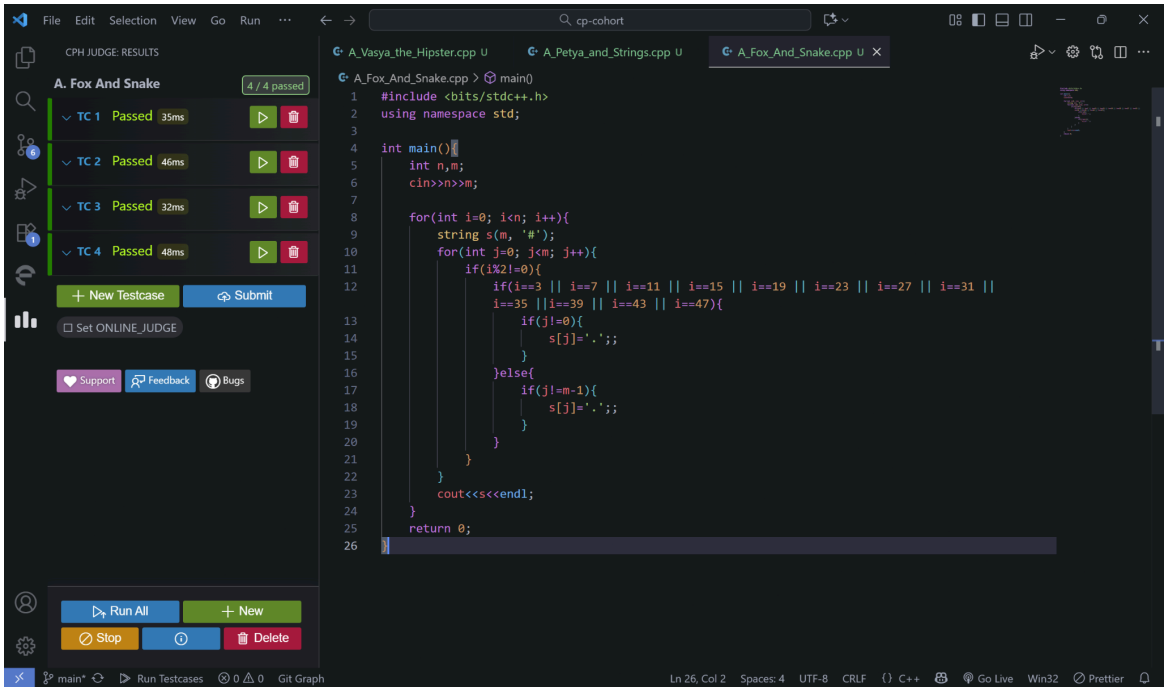
Q1, Q2, Q3 codes accepted in codeforces



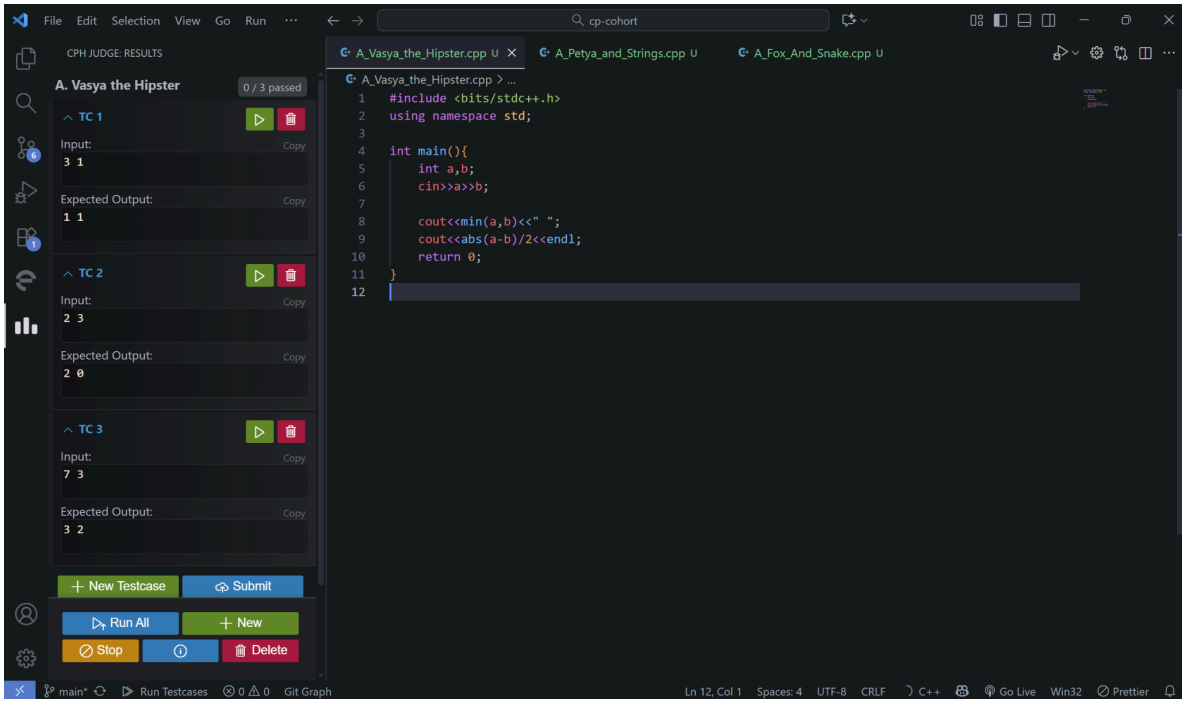
Q1)



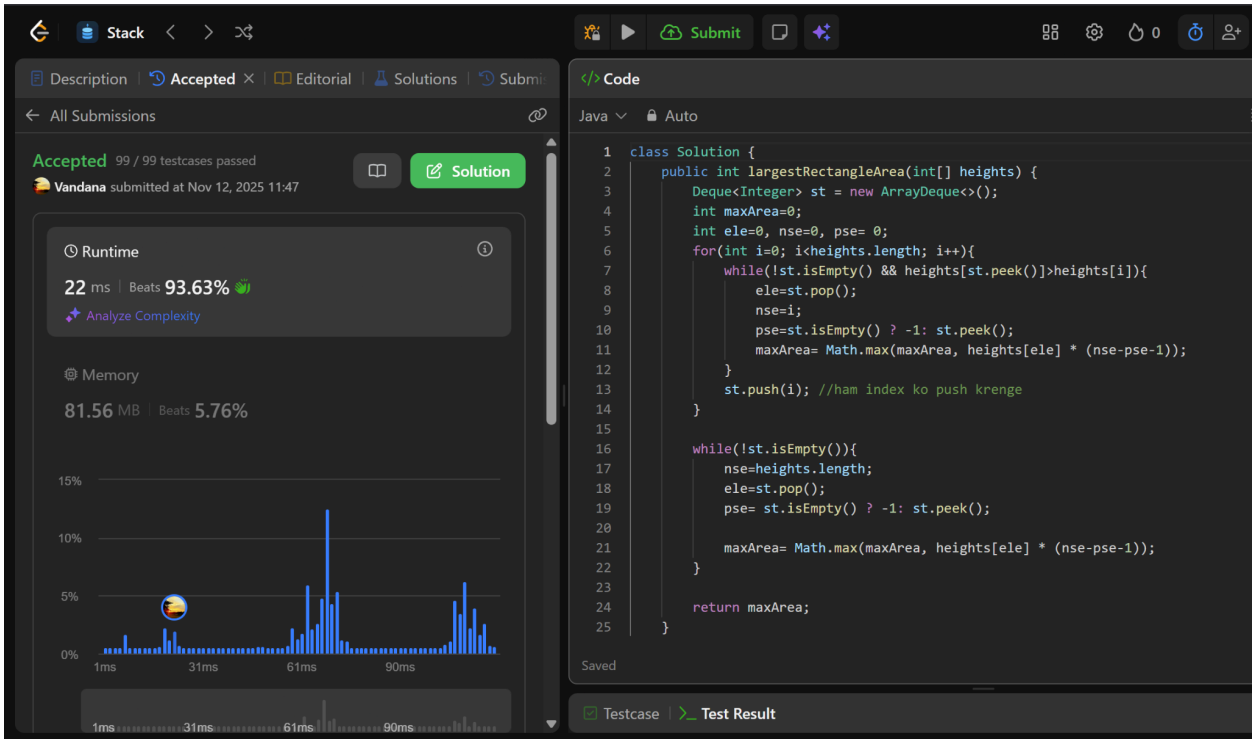
Q2)


File Edit Selection View Go Run
cp-cohort
CPH JUDGE: RESULTS
A. Fox And Snake
4 / 4 passed
TC 1 Passed 35ms
TC 2 Passed 46ms
TC 3 Passed 32ms
TC 4 Passed 48ms
New Testcase
Submit
Set ONLINE_JUDGE
Support
Feedback
Bugs
Run All
New
Stop
Delete
A_Vasya_the_Hipster.cpp U
A_Petya_and_Strings.cpp U
A_Fox_And_Snake.cpp U
A_Fox_And_Snake.cpp > main()
#include <bits/stdc++.h>
using namespace std;
int main(){
int n,m;
cin>>n>>m;
for(int i=0; i<n; i++){
string s(m, '#');
for(int j=0; j<m; j++){
if(i%2!=0){
if(i==3 || i==7 || i==11 || i==15 || i==19 || i==23 || i==27 || i==31 ||
i==35 ||i==39 || i==43 || i==47){
if(j!=0){
s[j]='.';;
}
}else{
if(j!=m-1){
s[j]='.';;
}
}
}
}
cout<<s<<endl;
}
return 0;
}
main* Run Testcases 0 0 Git Graph
Ln 26, Col 2 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 Prettier

Q3)


File Edit Selection View Go Run
cp-cohort
CPH JUDGE: RESULTS
A. Vasya the Hipster
0 / 3 passed
TC 1
Input:
3 1
Expected Output:
1 1
TC 2
Input:
2 3
Expected Output:
2 0
TC 3
Input:
7 3
Expected Output:
3 2
New Testcase
Submit
Run All
New
Stop
Delete
A_Vasya_the_Hipster.cpp U
A_Petya_and_Strings.cpp U
A_Fox_And_Snake.cpp U
A_Vasya_the_Hipster.cpp > ...
#include <bits/stdc++.h>
using namespace std;
int main(){
int a,b;
cin>>a>>b;
cout<<min(a,b)<<" ";
cout<<abs(a-b)/2<<endl;
return 0;
}
main* Run Testcases 0 0 Git Graph
Ln 12, Col 1 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 Prettier

Q4)


Stack
Submit
Description
Accepted
Editorial
Solutions
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%
15%
10%
5%
0%
1ms
31ms
61ms
90ms
Code
Java
Auto
class Solution {
public int largestRectangleArea(int[] heights) {
Deque<Integer> st = new ArrayDeque<>();
int maxArea=0;
int ele=0, nse=0, pse= 0;
for(int i=0; i<heights.length; i++){
while(!st.isEmpty() && heights[st.peek()]>heights[i]){
ele=st.pop();
nse=i;
pse=st.isEmpty() ? -1: st.peek();
maxArea= Math.max(maxArea, heights[ele] * (nse-pse-1));
}
st.push(i); //ham index ko push krenge
}
while(!st.isEmpty()){
nse=heights.length;
ele=st.pop();
pse= st.isEmpty() ? -1: st.peek();
maxArea= Math.max(maxArea, heights[ele] * (nse-pse-1));
}
return maxArea;
}
Saved
Testcase
Test Result