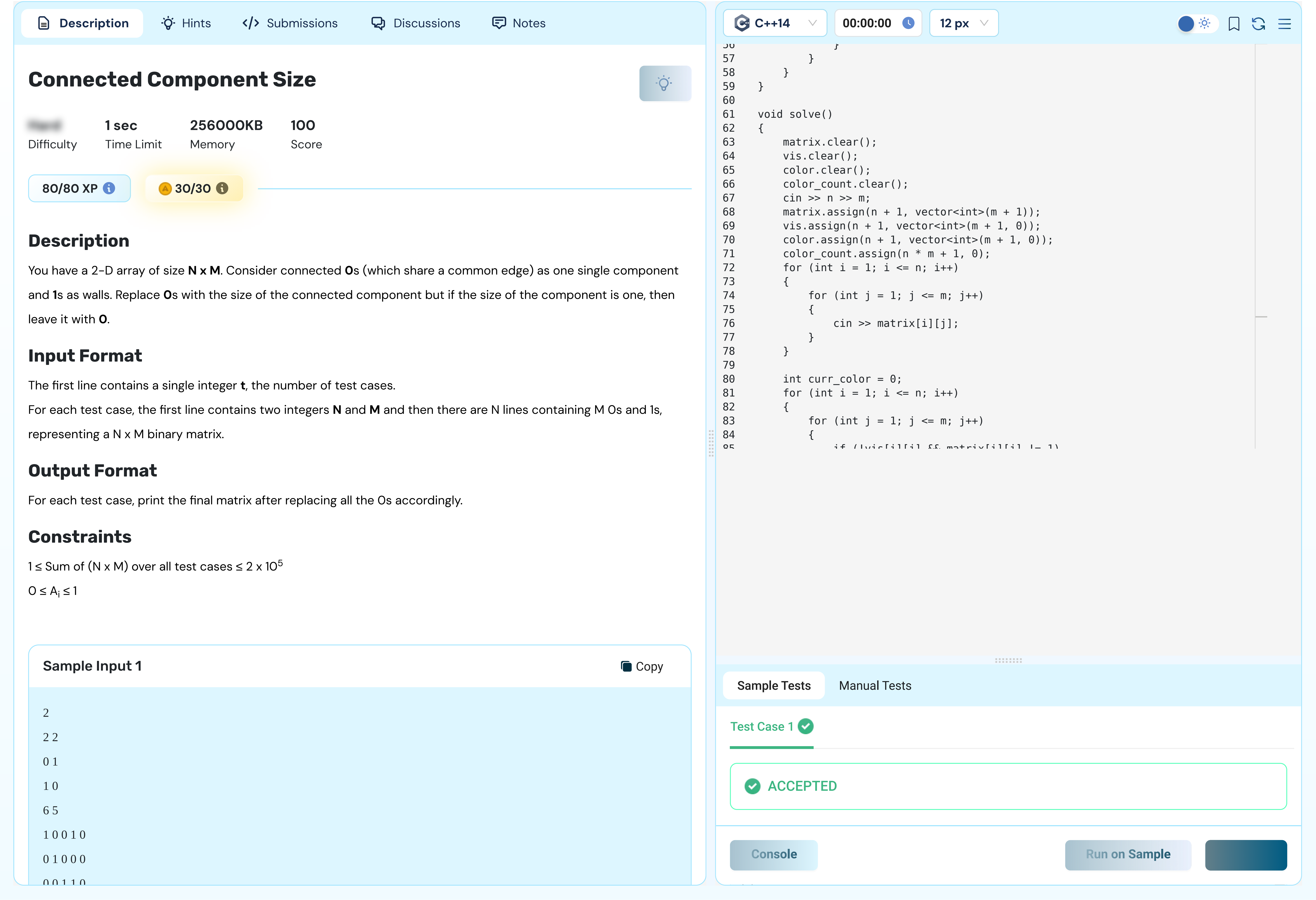

Connected Component Size

[redacted]	1 sec	256000KB	100
Difficulty	Time Limit	Memory	Score

80/80 XP ⓘ   30/30 ⓘ

Description

You have a 2-D array of size **N x M**. Consider connected **0**s (which share a common edge) as one single component and **1**s as walls. Replace **0**s with the size of the connected component but if the size of the component is one, then leave it with **0**.

Input Format

The first line contains a single integer **t**, the number of test cases.

For each test case, the first line contains two integers **N** and **M** and then there are N lines containing M 0s and 1s, representing a N x M binary matrix.

Output Format

For each test case, print the final matrix after replacing all the 0s accordingly.

Constraints

1 ≤ Sum of (N x M) over all test cases ≤ 2 x 10^5

0 ≤ A_i ≤ 1

Sample Input 1

```
2
2 2
0 1
1 0
6 5
1 0 0 1 0
0 1 0 0 0
0 0 1 1 0
```

```
            }
        }
    }
}

void solve()
{
    matrix.clear();
    vis.clear();
    color.clear();
    color_count.clear();
    cin >> n >> m;
    matrix.assign(n + 1, vector<int>(m + 1));
    vis.assign(n + 1, vector<int>(m + 1, 0));
    color.assign(n + 1, vector<int>(m + 1, 0));
    color_count.assign(n * m + 1, 0);
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= m; j++)
        {
            cin >> matrix[i][j];
        }
    }

    int curr_color = 0;
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= m; j++)
        {
            if (!vis[i][j] && matrix[i][j] != 1)
```

✓ ACCEPTED