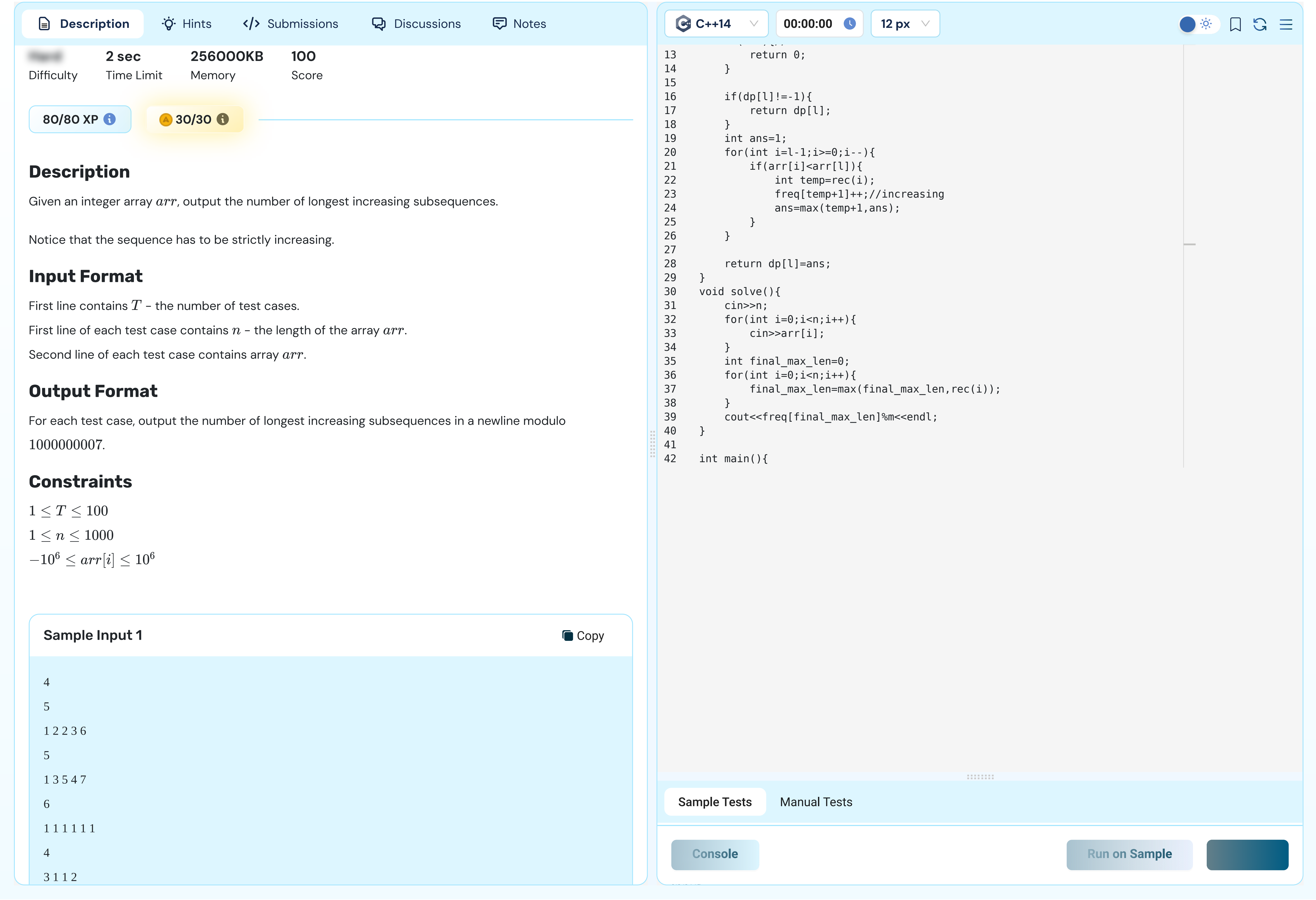

[redacted] Difficulty

2 sec Time Limit

256000KB Memory

100 Score

80/80 XP

30/30

Description

Given an integer array arr, output the number of longest increasing subsequences.

Notice that the sequence has to be strictly increasing.

Input Format

First line contains T - the number of test cases.

First line of each test case contains n - the length of the array arr.

Second line of each test case contains array arr.

Output Format

For each test case, output the number of longest increasing subsequences in a newline modulo 1000000007.

Constraints

$1 \leq T \leq 100$

$1 \leq n \leq 1000$

$-10^6 \leq arr[i] \leq 10^6$

Sample Input 1

Copy

```
4
5
1 2 2 3 6
5
1 3 5 4 7
6
1 1 1 1 1 1
4
3 1 1 2
```

```
        return 0;
    }

    if(dp[l]!=-1){
        return dp[l];
    }
    int ans=1;
    for(int i=l-1;i>=0;i--){
        if(arr[i]<arr[l]){
            int temp=rec(i);
            freq[temp+1]++;//increasing
            ans=max(temp+1,ans);
        }
    }

    return dp[l]=ans;
}
void solve(){
    cin>>n;
    for(int i=0;i<n;i++){
        cin>>arr[i];
    }
    int final_max_len=0;
    for(int i=0;i<n;i++){
        final_max_len=max(final_max_len,rec(i));
    }
    cout<<freq[final_max_len]%m<<endl;
}

int main(){
```