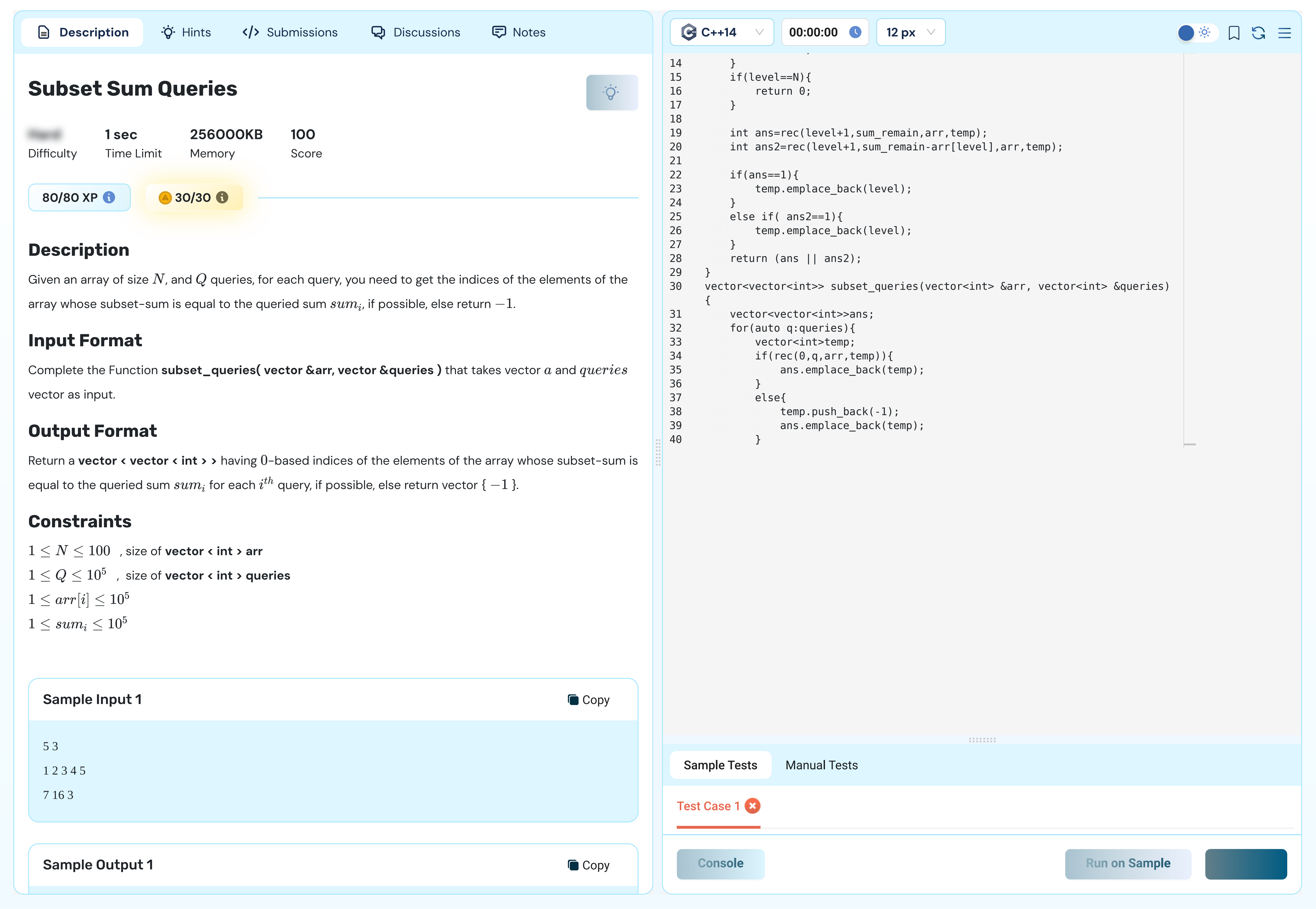

Subset Sum Queries

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

Description

Given an array of size N, and Q queries, for each query, you need to get the indices of the elements of the array whose subset-sum is equal to the queried sum sum_i, if possible, else return -1.

Input Format

Complete the Function **subset_queries(vector &arr, vector &queries)** that takes vector a and $queries$ vector as input.

Output Format

Return a **vector < vector < int > >** having 0-based indices of the elements of the array whose subset-sum is equal to the queried sum sum_i for each i^{th} query, if possible, else return vector { -1 }.

Constraints

$1 \leq N \leq 100$, size of **vector < int > arr**

$1 \leq Q \leq 10^5$, size of **vector < int > queries**

$1 \leq arr[i] \leq 10^5$

$1 \leq sum_i \leq 10^5$

Sample Input 1

```
5 3
1 2 3 4 5
7 16 3
```

Sample Output 1

```
    }
    if(level==N){
        return 0;
    }

    int ans=rec(level+1,sum_remain,arr,temp);
    int ans2=rec(level+1,sum_remain-arr[level],arr,temp);

    if(ans==1){
        temp.emplace_back(level);
    }
    else if( ans2==1){
        temp.emplace_back(level);
    }
    return (ans || ans2);
}
vector<vector<int>> subset_queries(vector<int> &arr, vector<int> &queries)
{
    vector<vector<int>>ans;
    for(auto q:queries){
        vector<int>temp;
        if(rec(0,q,arr,temp)){
            ans.emplace_back(temp);
        }
        else{
            temp.push_back(-1);
            ans.emplace_back(temp);
        }
```