Experiment 4

Searching and sorting

Student Name: Bhanu Pundir
UID: 20BCS1438
Branch: CSE
Section/Group:- 620-B
Semester: 5th
Date of Performance:28 Nov 22
Subject Name: CC Lab
Subject Code: 20CSP-314

Aim/Overview of the practical:

Searching and sorting

Q1. HackerLand National Bank has a simple policy for warning clients about possible fraudulent account activity. If the amount spent by a client on a particular day is greater than or equal to 2X the client's median spending for a trailing number of days, they send the client a notification about potential fraud. The bank doesn't send the client any notifications until they have at least that trailing number of prior days' transaction data.

Given the number of trailing days d and a client's total daily expenditures for a period of n days, determine the number of times the client will receive a notification over all n days.

Input Format

The first line contains two space-separated integers n and d, the number of days of transaction data, and the number of trailing days' data used to calculate median spending respectively.
The second line contains n space-separated non-negative integers where each integer i denotes $expenditure[i]$.

Constraints

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq d \leq n$
- $0 \leq expenditure[i] \leq 200$

Output Format

Sample Input 0

```
STDIN               Function
-----               --------
9 5                 expenditure[] size n =9, d = 5
2 3 4 2 3 6 8 4 5   expenditure = [2, 3, 4, 2, 3, 6, 8, 4, 5]
```

Sample Output 0

```
2
```

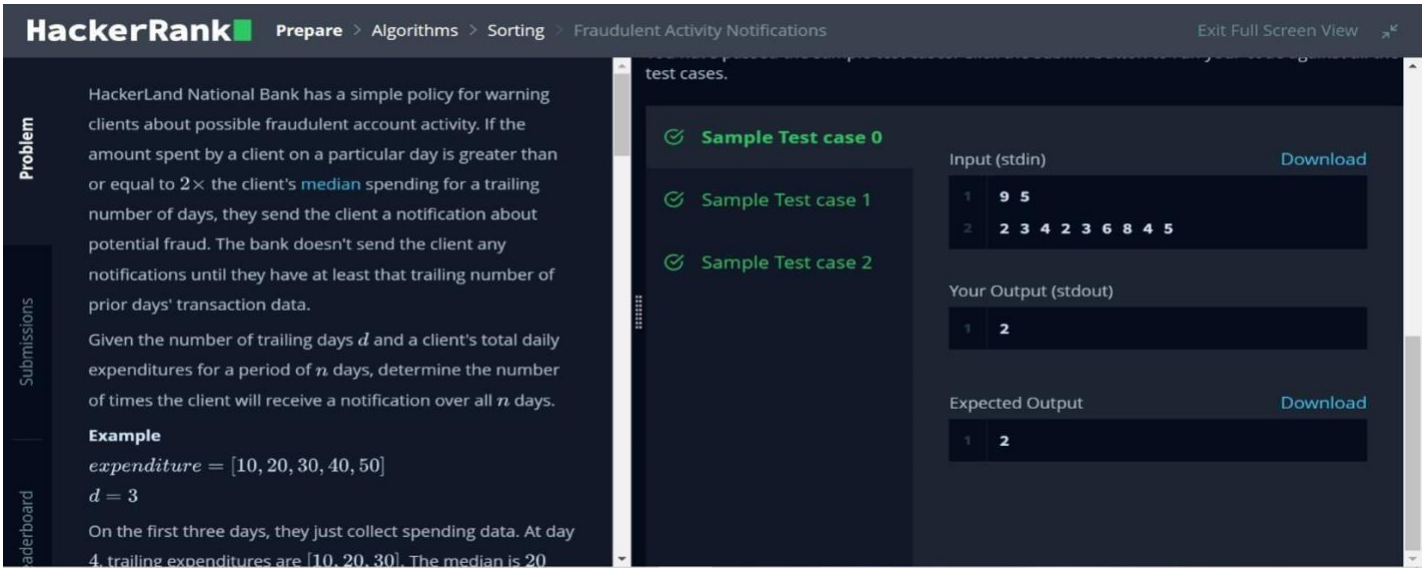
CODE:-

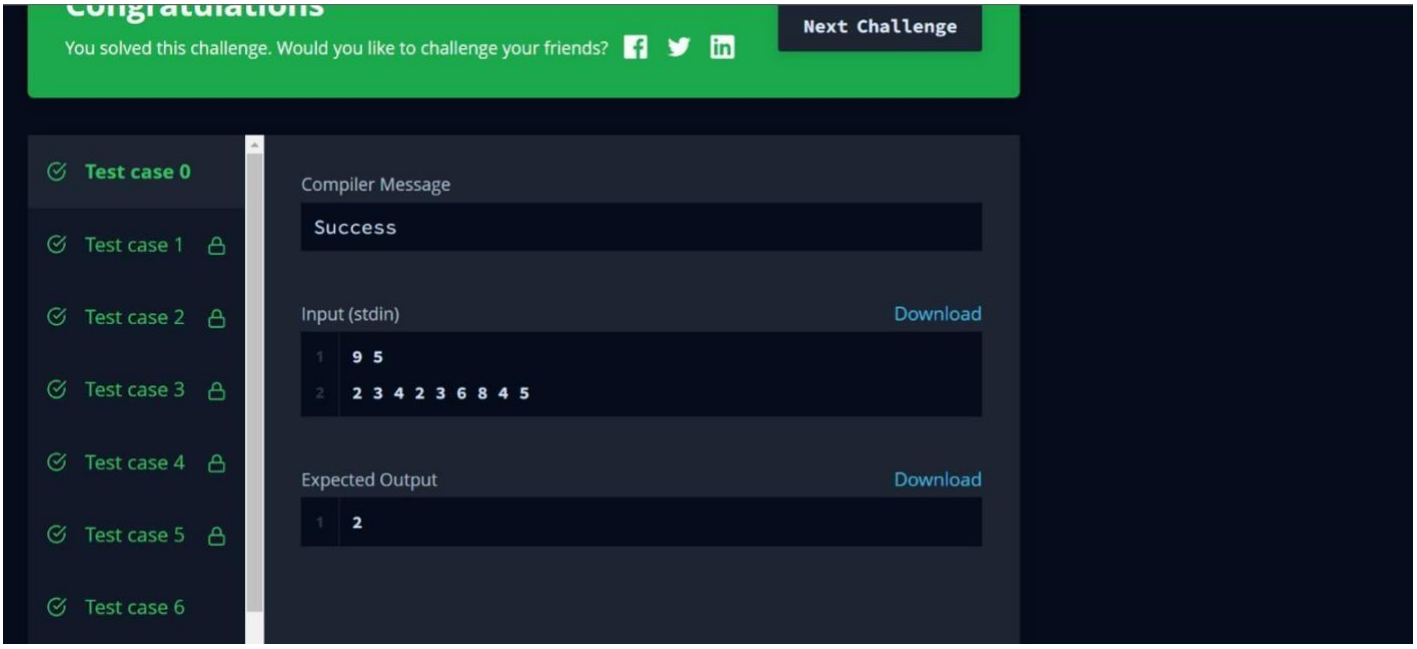
```
#include
<iostream>
#include <vector>
#include <map>
#include <set>
#include
<algorithm> using
namespace std;
#define MAXE 210
int A[200010];
int F[MAXE];
```

```
int median2(int D) { int p = 0;
   for (int i = 0; i < MAXE;
     i++) { p += F[i];
      if (p * 2 > D) { return 2 * i;
    } else if (p * 2 == D) {
       for (int j = i + 1;
          ; j++) { if (F[j])
          {
          return i + j;
         }
       }
    }
  }
 return -1;
}


int main()
 { int N,
 D;
cin >> N >> D;
for (int i = 0; i < N;
  i++) { cin >> A[i];
}
int result = 0;
for (int i = 0; i < N;
  i++) { if (i >= D) {
```

```
    if (A[i] >= median2(D)) {
     ++result;
      }
      F[A[i - D]]--;
  }
  F[A[i]]++;
}
cout << result <<
endl; return 0;
}
```

OUTPUT:-




Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6
Compiler Message
Success
Input (stdin)
Download
9 5
2 3 4 2 3 6 8 4 5
Expected Output
Download
2

Q2. PROBLEM STATEMENT

MISSING NUMBERS

Prepare > Algorithms > Search > Missing Numbers

Missing Numbers ★

1012.57

Rank: 112

Problem | Submissions | Leaderboard | Discussions | Editorial | Topics

Given two arrays of integers, find which elements in the second array are missing from the first array.

Example

$arr = [7, 2, 5, 3, 5, 3]$

$brr = [7, 2, 5, 4, 6, 3, 5, 3]$

The brr array is the orginal list. The numbers missing are $[4, 6]$.

Notes

- If a number occurs multiple times in the lists, you must ensure that the frequency of that number in both lists is the same. If that is not the case, then it is also a missing number.
- Return the missing numbers sorted ascending.
- Only include a missing number once, even if it is missing multiple times.
- The difference between the maximum and minimum numbers in the original list is less than or equal to 100.

Function Description

Complete the missingNumbers function in the editor below. It should return a sorted array of missing numbers.

missingNumbers has the following parameter(s):

- int arr[n]: the array with missing numbers
- int brr[m]: the original array of numbers

Input Format

There will be four lines of input:

n - the size of the first list, arr

The next line contains n space-separated integers $arr[i]$

m - the size of the second list, brr

The next line contains m space-separated integers $brr[i]$

Constraints

- $1 \leq n, m \leq 2 \times 10^5$
- $n \leq m$
- $1 \leq brr[i] \leq 10^4$
- $max(brr) - min(brr) \leq 100$

Sample Input

```
10
203 204 205 206 207 208 203 204 205 206
13
203 204 204 205 206 207 205 208 203 206 205 206 204
```

Sample Output

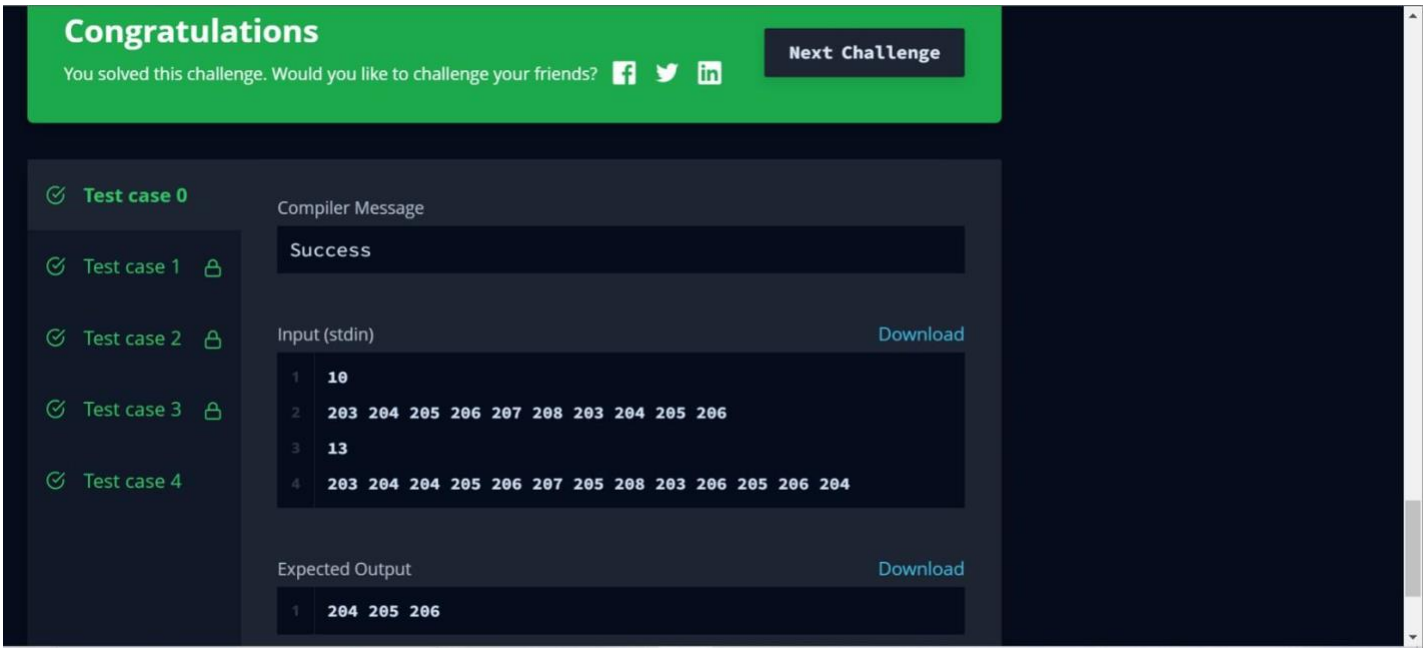
```
204 205 206
```

Code:

```
#include <bits/stdc++.h>
using namespace std;

const int maxn = 10000;

int A[maxn * 2 + 5];

int main ()
{

    int n, m;

    int xmin = maxn, xmax = -maxn;

    cin >> n;
```

```
    for (int i = 0; i < n; i++){

        int tmp;

        cin >> tmp;

        A[tmp]--;

    }
    cin >> m;

    for (int i = 0; i < m; i++){

        int tmp;

        cin >> tmp;

        A[tmp]++;

        if (xmax < tmp){
            xmax = tmp;
        }
        if (xmin > tmp){
           xmin = tmp;
        }

    }

    for (int i = xmin; i <= xmax; i++){

        if (A[i] > 0){

            cout << i << " ";

        }

    }

    return 0;

}
```

OUTPUT:-


Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Compiler Message
Success
Input (stdin)
Download
10
203 204 205 206 207 208 203 204 205 206
13
203 204 204 205 206 207 205 208 203 206 205 206 204
Expected Output
Download
204 205 206