# **Experiment -1.4**

Student Name: SANSKAR AGRAWAL	UID: 20BCS5914
Branch: BE-CSE	Section/Group: 806-B
Semester: 5 <sup>th</sup>	<b>Subject: Competitive Coding Lab</b>

## **Question -1**

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

#### **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

#### Returns

• *int[]:* an array of integers

## **Input Format**

- There will be four lines of input:
- n the size of the first list, arr

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

m - the size of the second list, brr

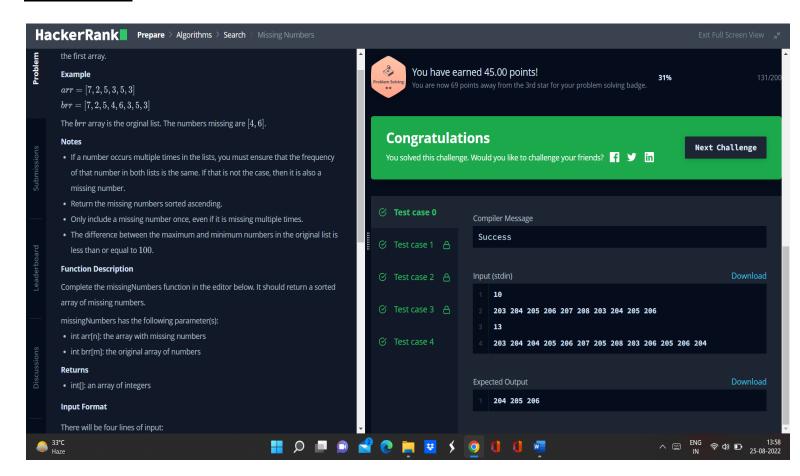
The next line contains m space-separated integers brr[i].

#### **SOLUTION:**

```
#include<iostream>
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++ ) {</pre>
  if(A[i] > 0)
   cout << i << " ";
 }
```

```
return 0;
```

#### **TESTCASES:**



## **Question -2**

Sorting is useful as the first step in many different tasks. The most common task is to make finding things easier, but there are other uses as well. In this case, it will make it easier to determine which pair or pairs of elements have the smallest absolute difference between them.

## **Function Description**

Complete the *closestNumbers* function in the editor below.

closestNumbers has the following parameter(s):

• *int arr[n]:* an array of integers

#### Returns

- int[]: an array of integers as described

## **Input Format**

The first line contains a single integer, the length of arr.

The second line contains n space-separated integers, arr[i].

## **SOLUTION**:

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
int cmp(int x,int y)
  return x<y;
int main() {
 vector<int> arr,temp;
 int i,n,diff,x;
  cin>>n;
  for(i=0;i<n;i++)
     cin>>x;
     arr.push_back(x);
  }
  sort(arr.begin(),arr.end(),cmp);
  diff=arr[1]-arr[0];
  for(i=2;i<n;i++)
     if(diff>(arr[i]-arr[i-1]))
       diff=arr[i]-arr[i-1];
       temp.clear();
       temp.push_back(arr[i-1]);
       temp.push_back(arr[i]);
     else if(diff==(arr[i]-arr[i-1]))
```

```
temp.push_back(arr[i-1]);
  temp.push_back(arr[i]);
}

for(i=0;i<temp.size();i++)
  cout<<temp[i]<<" ";

return 0;</pre>
```

#### **TESTCASES**:

