

Experiment -1.4

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

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++ ) {
        if( A[i] > 0 ) {
            cout << i << " ";
        }
    }
}
```

```
return 0;
}
```

TESTCASES:

HackerRank Prepare > Algorithms > Search > Missing Numbers

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 original 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

Returns

- `int[]`: an array of integers

Input Format

There will be four lines of input:

Test case 0 ✓

Test case 1 ✓

Test case 2 ✓

Test case 3 ✓

Test case 4 ✓

Compiler Message

Success

Input (stdin) Download

```
1 10
2 203 204 205 206 207 208 203 204 205 206
3 13
4 203 204 204 205 206 207 205 208 203 206 205 206 204
```

Expected Output Download

```
1 204 205 206
```

Problem Solving You have earned 45.00 points! You are now 69 points away from the 3rd star for your problem solving badge. 31% 131/200

Congratulations

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)

Next Challenge

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;
}

```

TESTCASES:

The screenshot displays the HackerRank interface for the 'Closest Numbers' problem. The left sidebar contains navigation links: Leaderboard, Discussions, Editorial, and Topics. The main content area is divided into two columns. The left column provides problem details, including 'Explanation 0' with the calculation $(30) - (-20) = 50$, 'Sample Input 1' (12, -20, -3916237, -357920, -3620601, 7374819, -7330761, 30, 6246457, -6461594), 'Sample Output 1' (-520, -470, -20, 30), 'Explanation 1' with the calculation $(-470) - (-520) = 30$, 'Sample Input 2' (4, 5, 4, 3, 2), 'Sample Output 2' (2, 3, 3, 4, 4, 5), and 'Explanation 2' stating the minimum difference is 1 with valid pairs (2, 3), (3, 4), and (4, 5). The right column shows a 'Congratulations' message: 'You have earned 35.00 points! You are now 12.51 points away from the 3rd star for your problem solving badge.' It also lists six test cases, all marked as successful. The 'Compiler Message' section shows 'Success'. The 'Input (stdin)' section shows two test cases: 1) 10 and 2) the same sample input as in the left column. The 'Expected Output' section shows the output for the first test case: -20 30. The bottom of the screen shows a Windows taskbar with the date 25-08-2022 and time 13:59.