



Dashboard &gt; Algorithms &gt; Search &gt; Missing Numbers

Badge Progress [\(Details\)](#)

Points: 976 Rank: 37885

# Missing Numbers

by HackerRank

Problem

Submissions

Leaderboard

Discussions

Editorial

Topics

Numeros, the Artist, had two lists **A** and **B**, such that **B** was a permutation of **A**. Numeros was very proud of these lists. Unfortunately, while transporting them from one exhibition to another, some numbers were left out of **A**. Can you find the missing numbers?

## 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.
- You have to print all the missing numbers in ascending order.
- Print each missing number once, even if it is missing multiple times.
- The difference between maximum and minimum number in **B** is less than or equal to **100**.

## Input Format

There will be four lines of input:

**n** - the size of the first list

This is followed by **n** space-separated integers that make up the first list.

**m** - the size of the second list

This is followed by **m** space-separated integers that make up the second list.

## Constraints

- $1 \leq n, m \leq 2 \times 10^5$
- $n \leq m$
- $1 \leq x \leq 10^4, x \in B$
- $X_{max} - X_{min} < 101$

## Output Format

Output the missing numbers in ascending order.

## 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
```

## Explanation

**204** is present in both arrays. Its frequency in **A** is **2**, while its frequency in **B** is **3**. Similarly, **205** and **206** occur twice in **A**, but thrice in **B**. So, these three numbers are our output. The rest of the numbers have the same frequency in both lists.

Submissions: [36320](#)

Max Score: 45

Difficulty: Easy

Rate This Challenge:



Need Help?

[Caching](#)[More](#)

Current Buffer (saved locally, editable)

C++14

```
1 #include <iostream>
2 #include <algorithm>
3 #include <map>
4 using namespace std;
5 #define ul unsigned long int
6
7 int main()
8 {
9     ul N1=0, N2=0, minArray=0;
10    cin>>N1;
11
12    map<ul,ul> map1, map2;
13
14    for(auto i=0; i<N1;++i)
15    {
16        ul temp;
17        cin>>temp;
18        map1[temp]++;
19    }
20
21    cin>>N2;
22    for(auto i=0; i<N2;++i)
23    {
24        ul temp;
25        cin>>temp;
26        map2[temp]++;
27    }
28
29    minArray=min(map1.size(), map2.size());
30
31    auto itrMap1 = map1.cbegin();
32    auto itrMap2 = map2.cbegin();
33    for(auto i=0; i<minArray;++i)
34    {
35        if(itrMap1->second != itrMap2->second)
36            cout<<itrMap1->first<<" ";
37
38        ++itrMap1;
39        ++itrMap2;
40    }
41    return 0;
42 }
```

Line: 1 Col: 1

[Upload Code as File](#) ☐ Test against custom input

Run Code

Submit Code