

# Get Common Elements - 2

---

## Easy

1. You are given a number  $n_1$ , representing the size of array  $a_1$ .
2. You are given  $n_1$  numbers, representing elements of array  $a_1$ .
3. You are given a number  $n_2$ , representing the size of array  $a_2$ .
4. You are given  $n_2$  numbers, representing elements of array  $a_2$ .
5. You are required to find the intersection of  $a_1$  and  $a_2$ . To get an idea check the example below:

- if  $a_1 \rightarrow 1\ 1\ 2\ 2\ 2\ 3\ 5$
- and  $a_2 \rightarrow 1\ 1\ 1\ 2\ 2\ 4\ 5$
- intersection is  $\rightarrow 1\ 1\ 2\ 2\ 5$

Note  $\rightarrow$  Don't assume the arrays to be sorted. Check out the question video.

## Constraints

- $1 \leq n_1, n_2 \leq 100$
- $0 \leq a_1[i], a_2[i] < 10$  Time complexity should be  $O(n)$

## Format

### Input

- A number  $n_1$
- $n_1$  number of elements line separated
- A number  $n_2$
- $n_2$  number of elements line separated

### Output

- All relevant elements of intersection in separate lines
- The elements of intersection should be printed in order of their occurrence in  $a_2$ .

## Example

### Sample Input

```
7
1
1
2
2
2
3
5
7
```

```
1
1
1
2
2
4
5
```

## Sample Output

```
1
1
2
2
5
```

## Solution

```
import java.io.*;
import java.util.*;

public class GetCommonElement2 {

    public static void main(String[] args) throws Exception {
        Scanner scn = new Scanner(System.in);
        int n1 = scn.nextInt();
        int[] a = new int[n1];
        for (int i = 0; i < n1; i++)
            a[i] = scn.nextInt();

        int n2 = scn.nextInt();
        int[] b = new int[n2];
        for (int i = 0; i < n2; i++)
            b[i] = scn.nextInt();

        // FreqMap of A array
        HashMap<Integer, Integer> hm = new HashMap<>();
        for (int ele : a)
            hm.put(ele, hm.getOrDefault(ele, 0) + 1);

        for (int ele : b) {
            if (hm.containsKey(ele) && hm.get(ele) > 0) {
                System.out.println(ele);

                int oldFreq = hm.get(ele);
                int newFreq = oldFreq - 1;
                hm.put(ele, newFreq);
            }
        }
    }
}
```

```
}
```

```
}
```