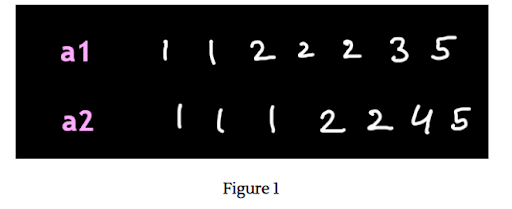
**. PROBLEM DISCUSSION**

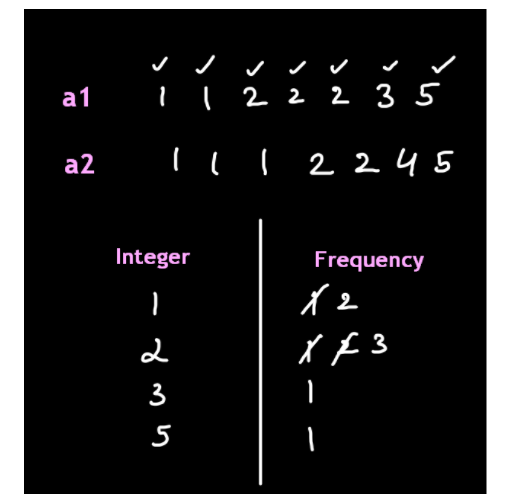
● You are given 2 arrays a1 and a2 of size n1 and n2 respectively. ● You are required to print all the elements of a2 which are also present in a1 (in order of their occurence in a2). ● Make sure to not print duplicates (a2 may have the same value present multiple times). ● Say, the given arrays are as shown in figure 1.

****

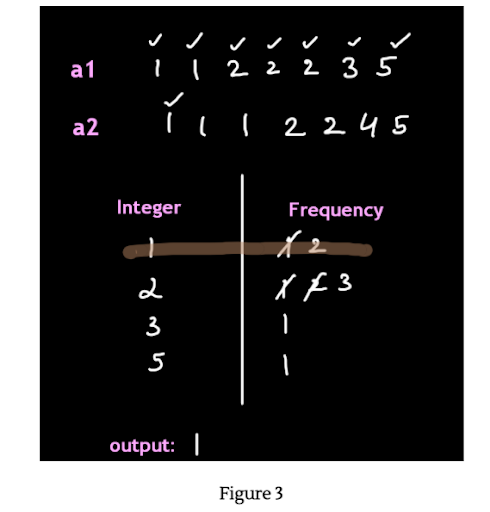
According to this question, the output should be: "1 2 5".

**2. Approach :**

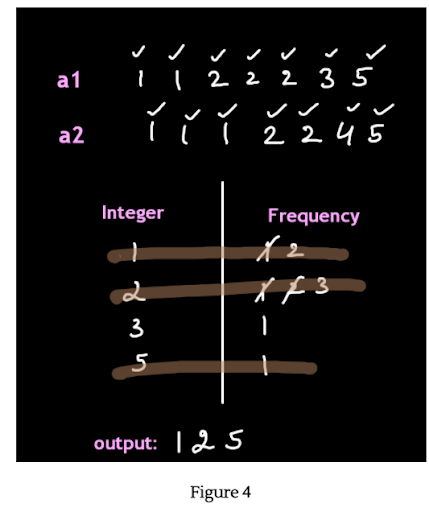
We make a Hashmap table for every unique character of a1 vs. its frequency.

****

In figure 2, we iterate through every element of the array a1 and subsequently increase the frequency against it. Now, we iterate through each element of the array a2. If that element is present in our Hashmap, then it is printed and the record of that element is removed from the hashmap as shown in figure 3. If the element is not present then we do nothing.

****

On iterating through all the elements of a2, the output obtained is as follows.

****

**3. CODE**

ConsoleCpp

#include <iostream>

#include <unordered\_map>

using namespace std;

int main() {

int n1,n2;

cin >> n1;

int arr1[n1];

unordered\_map<int,int> mp;

for (int i = 0; i < n1; i++) {

cin>>arr1[i];

mp[arr1[i]]++;

}

cin >> n2;

int arr2[n2];

for (int i = 0; i < n2; i++) {

cin >> arr2[i];

if(mp.find(arr2[i])!=mp.end()){

cout << arr2[i] << endl;

mp.erase(arr2[i]);

}

}

}

**4. TIME & SPACE COMPLEXITY**

TIME COMPLEXITY

O(n)

SPACE COMPLEXITY

O(n) where n=number of entities in the hashmap.