**Even occurring elements**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)[bit](http://www.practice.geeksforgeeks.org/tag-page.php?tag=bit&isCmp=0)

Given an array that contains odd number of occurrences for all numbers except for a few elements which are present even number of times. Find the elements which have even occurrences in the array in O(n) time complexity and O(1) extra space.

**Note:**In some array, array contains only odd number then you have to print only a blank new line.

**Input:**  
The first line of input contains a single integer T denoting the number of test cases. ThenT test cases follow. Each test case consist of two lines. The first line of each test case consists of an integer N, where N is the size of array.  
The second line of each test case contains N space separated integers denoting array elements.

**Output:**

Corresponding to each test case, in a new line, print the elements which have even occurrences in the array.

**Constraints:**

1 ≤ T ≤ 100  
1 ≤ N ≤ 200  
1 ≤ A[i] ≤ 63  
  
**Example:**

**Input**  
3  
11  
9 12 23 10 12 12 15 23 14 12 15  
5  
23 12 56 34 32  
4  
10 34 10 56

**Output**  
12 23 15  
  
10

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=1003>

#include <iostream>

#include <stdio.h>

#include <algorithm>

#include <map>

#include <vector>

#define ll long long int

using namespace std;

int main() {

   int t;

   scanf("%d", &t);

   while(t--) {

      int n;

      scanf("%d", &n);

      int arr[n];

      for(int i =0; i<n; i++)  {

        scanf("%d", &arr[i]);

      }

      //guardo en un diccionario la frecuencia de cada elemento

      std::map<int,int> map;

      for(int i =0; i< n; i++) {

         map[arr[i]]++;

      }

      //al no haber en c++ un linked hash map, que te guarda el elemento-frecuencia

      //en el orden en QUE APARECE EN EL ARRAY arr.

      //declaro un array de distintos y recorro el array arr

      std::vector<int> distintos;

      std::vector<int>::iterator it;

      for(int i =0; i<n; i++) {

         //me fijo si el elem del arr ya esta en distintos distintos

         it = find (distintos.begin(), distintos.end() , arr[i]);

         //si no esta en distintos me fijo si la frecuencia del elemento es par

         if(map[arr[i]] % 2 ==0 && it == distintos.end() ){

             distintos.push\_back(arr[i]); //y lo guardo en distintos como elemento ya mostrado

             cout << arr[i] << " "; //lo muestro

         }

      }

      printf("**\n**");

   }

  return 0;

}