**Intersection of two arrays**

[math](http://www.practice.geeksforgeeks.org/tag-page.php?tag=math&isCmp=0)[Rockstand](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Rockstand&isCmp=1)

Given two array A and B, print intersection (or common elements) of the two array.  If no element is common in two array, then print Zero.

**Input:**

The first line of input contains an integer T denoting the number of test cases.  
The first line of each test case is N and M,N is the size of array A and M is size of array B.  
The second line of each test case contains N input A[i].  
The third line of each test case contains M input B[i].  
  
**Output:**

Print the intersecting elements.If no element is common in two array, then print "Zero" without quotes.  
  
**Constraints:**

1 ≤ T ≤ 50  
1 ≤ N, M ≤ 100  
1 ≤ A[i], B[i] ≤ 1000  
  
**Example:**

Input  
1  
5 3  
89 24 75 11 23  
89 2 4

Output  
89

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=536>

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <algorithm>

#include <vector>

using namespace std;

int main() {

int t;

scanf("%d", &t);

while(t--) {

int n,m;

scanf("%d %d", &n, &m);

std::vector<int> a;

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

int elem;

scanf("%d", &elem);

a.push\_back(elem);

}

std::vector<int> b;

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

int elem;

scanf("%d", &elem);

b.push\_back(elem);

}

std::sort(a.begin(), a.end());

bool contiene=false;

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

if(std::find(b.begin(), b.end(), a[i]) != b.end()) {

/\* v contains x \*/

printf("%d ", a[i]);

contiene = true;

}

}

if(!contiene) {

printf("Zero");

}

printf("\n");

}

return 0;

}