**Count the elements**

Show Topic Tags   

[Amazon](http://practice.geeksforgeeks.org/company/Amazon/)

Given two unsorted arrays A, B. They can contain duplicates. For each element in A , count elements less than or equal to it in array B .

**Time Complexity:** O(n)

**Input:**  
The first line contains a single integer Ti.e. the number of test cases. The first line of each test case consists of a integer N. The second and third line of each test case consists of N spaced integers representing array A and array B respectively.

**Output:**  
In one line for each element in array A print the elements less than or equal to it in array B with a comma ',' in between .

**Constraints:**  
1<=T<=100  
1<=N<=100

**Example:  
Input:**  
2  
6  
1 2 3 4 7 9  
0 1 2 1 1 4  
7  
95 39 49 20 67 26 63   
77 96 81 65 60 36 55   
**Output:**  
4,5,5,6,6,6  
6,1,1,0,4,0,3  
     

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

<http://practice.geeksforgeeks.org/problems/count-the-elements/0>

------con hashing en C++------

#include <iostream>

#include <stdio.h>

#include <vector>

using namespace std;

int main() {

//code

int t;

scanf("%d", &t);

while(t-- > 0) {

int n;

scanf("%d", &n);

int a[n];

int max =0;

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

scanf("%d", &a[i]);

if(a[i] > max) {

max = a[i];

}

}

int b[n];

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

scanf("%d", &b[i]);

}

int c[max+1];

for(int i =0; i < max+1; i++) {

c[i] = 0;

}

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

c[b[i]]++;

}

/\*

for(int i =0; i<max+1; i++) {

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

}

\*/

std::vector<int> res;

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

int j =0;

int cont =0;

while(j <= a[i]) {

cont += c[j];

j++;

}

res.push\_back(cont);

}

for(int i =0; i<n-1; i++) {

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

}

printf("%d", res[n-1]);

printf("\n");

}

return 0;

}

-------------------en java ordenando el array-------------------------

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication226;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Arrays;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication226 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

int n = Integer.parseInt(br.readLine());

String[] s\_a = br.readLine().trim().split(" ");

int[] a = new int[n];

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

a[i] = Integer.parseInt(s\_a[i]);

}

String[] s\_b = br.readLine().trim().split(" ");

int[] b = new int[n];

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

b[i] = Integer.parseInt(s\_b[i]);

}

Arrays.sort(b);

ArrayList<Integer> c = new ArrayList<Integer>();

for(int i =0; i<a.length; i++) {

int j =0;

int cont =0;

while(j<b.length && b[j] <= a[i]) {

cont++;

j++;

}

c.add(cont);

}

for(int i =0; i<c.size()-1; i++){

System.out.print(c.get(i)+",");

}

System.out.print(c.get(c.size()-1));

System.out.println();

}

}

}

-----------EDITORIAL-------------

**Count the elements [Editorial]**

Use Hashing, and Count the number of elements in array B in the hash array. Run a loop from i=0 to Max(maximum value in the array A) and keep on adding the Count(from hash array) and store the final sum in a new array D each time.

For each value in array A(a[i]) print D[a[i]].