**Relative Sorting**

Submissions: [15405](https://practice.geeksforgeeks.org/problem_submissions.php?pid=434)  Accuracy:

42.53%

   Difficulty: [Medium](https://practice.geeksforgeeks.org/Medium/0/0/)   Marks: 4

Associated Course(s): [Interview Preparation](https://practice.geeksforgeeks.org/courses/interview-preparation/)[Geeks Classes in Noida](https://practice.geeksforgeeks.org/courses/geeks-classes-in-noida/)

Show Topic Tags   

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

Given two arrays **A1[]** and **A2[]** of size **N** and **M**respectively. The task is to sort A1 in such a way that the relative order among the elements will be same as those in A2. For the elements not present in A2, append them at last in sorted order. It is also given that the number of elements in A2[] are smaller than or equal to number of elements in A1[] and A2[] has all distinct elements.

**Note:** Expected time complexity is **O(N log(N))**.

**Input:**  
First line of input contains number of testcases. For each testcase, first line of input contains length of arrays N and M and next two line contains N and M elements respectively.

**Output:**  
Print the relatively sorted array.

**Constraints:**  
1 ≤ T ≤ 100  
1 ≤ N,  ≤ 106  
1 ≤ A1[], A2[] <= 106

**Example:**  
**Input:**  
1  
11 4  
2 1 2 5 7 1 9 3 6 8 8  
2 1 8 3

**Output:**  
2 2 1 1 8 8 3 5 6 7 9

**Explanation:  
Testcase 1:** Array elements of A1[] are sorted according to A1[].

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/relative-sorting/0#ExpectOP) option \*\*

[Author: rajatjha](https://auth.geeksforgeeks.org/user/rajatjha/practice/)

<https://practice.geeksforgeeks.org/problems/relative-sorting/0>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Program

{

static void Ordenar(int[] arr, int[] sort)

{

int max = arr.Max();

int[] count = new int[max + 1];

for (int i = 0; i < arr.Length; i++)

{

count[arr[i]]++;

}

//List<int> ord = new List<int>();

for (int i = 0; i < sort.Length; i++)

{

for (int j = 0; j < count[sort[i]]; j++)

{

Console.Write(sort[i] + " ");

//ord.Add(sort[i]);

}

count[sort[i]] = 0;

}

for(int i =0; i<count.Length; i++)

{

if(count[i] > 0)

{

//ord.Add(i);

for (int j = 0; j < count[i]; j++)

{

Console.Write(i + " ");

}

}

}

}

static void Main(string[] args)

{

//int[] arr = { 2, 1, 2, 5, 7, 1, 9, 3, 6, 8, 8 };

//int[] sort = { 2, 1, 8, 3 };

//List<int> res = Ordenar(arr, sort);

//for(int i =0; i<res.Count; i++)

//{

// Console.Write(res[i] + " ");

//}

int t = int.Parse(Console.ReadLine());

while (t-- > 0)

{

string[] input = Console.ReadLine().Trim().Split(' ');

int n = int.Parse(input[0]);

int m = int.Parse(input[1]);

int[] a = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => int.Parse(e));

int[] b = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => int.Parse(e));

Ordenar(a, b);

Console.WriteLine();

}

Console.ReadLine();

}

}

}