**Merge Two Sorted Array**

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

You have to merge the two sorted arrays into one sorted array (in non-increasing order).

**Input:**

First line contains an integer **T**, denoting the number of test cases.  
First line of each test case contains two space separated integers **X and Y**, denoting the size of the two sorted arrays.  
Second line of each test case contains**X** space separated integers, denoting the first sorted array P.  
Third line of each test case contains**Y** space separated integers, denoting the second array Q.

**Output:**

For each test case, print (**X + Y**) space separated integer representing the merged array.

**Constraints:**

1 <= **T** <= 100  
1 <= **X, Y** <= 5\*104  
0 <= **Pi**, **Qi** <= 109

**Example:**

INPUT:

1

4 5

7 5 3 1

9 8 6 2 0

OUTPUT:

9 8 7 6 5 3 2 1 0

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

#include <iostream>

#include <stdio.h>

#include <vector>

#include <conio.h>

using namespace std;

int main() {

 int t;

 scanf("%d", &t);

    while(t--) {

        int X, Y;

        scanf("%d %d", &X, &Y);

        int P[X];

        for(int i = 0; i < X; i++) {

          scanf("%d", &P[i]);

        }

        int Q[Y];

        for(int i =0; i < Y; i++) {

         scanf("%d", &Q[i]);

        }

         int i=0, j=0;

        std::vector<int> merge;

        while (i < X && j < Y)

        {

            if (P[i] > Q[j])

            {

                merge.push\_back(P[i]);

                i++;

            }

            else

            {

                merge.push\_back(Q[j]);

                j++;

            }

        }

        while (i < X)

        {

            merge.push\_back(P[i++]);

        }

        while (j < Y)

        {

            merge.push\_back(Q[j++]);

        }

        for(int i =0; i < merge.size(); i++) {

            printf("%d ", merge[i]);

        }

        printf("**\n**");

    }

    getch();

 return 0;

}