**Sort an array according to other array**

Given two integer arrays A1[ ] and A2[ ] of size N and M respectively. Sort the first array A1[ ] such that all the relative positions of the elements in the first array are the same as the elements in the second array A2[ ]. For the elements not present in A2[ ] but in A1[ ], it appends them at the last in increasing order.

Note: If elements are repeated in the second array, consider their first occurance only.

Input Format

The first line contains the integer n.

The second line contains the integer m.

The third line contains n space-separated integers.

The fourth line contains m space-separated integers.

Constraints

1 ≤ N, M ≤ 106

1 ≤ A1[i], A2[i] ≤ 106

Output Format

A single line containing n space-separated integers.

Sample Input 0

11

4

2 1 2 5 7 1 9 3 6 8 8

2 1 8 3

Sample Output 0

2 2 1 1 8 8 3 5 6 7 9

Explanation 0

Array elements of A1[] are sorted according to A2[]. So 2 comes first then 1 comes, then comes 8, then finally 3 comes, now we append remaining elements in sorted order.

Sample Input 1

11

4

2 1 2 5 7 1 9 3 6 8 8

99 22 444 56

Sample Output 1

1 1 2 2 3 5 6 7 8 8 9

Explanation 1

No A1[] elements are in A2[] so we cannot sort A1[] according to A2[]. Hence we sort the elements in non-decreasing order.

**Solution :**

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int first(int arr[], int low, int high, int x, int n)

{

if (high >= low) {

int mid = low + (high - low) / 2;

if ((mid == 0 || x > arr[mid - 1]) && arr[mid] == x)

return mid;

if (x > arr[mid])

return first(arr, (mid + 1), high, x, n);

return first(arr, low, (mid - 1), x, n);

}

return -1;

}

void sortAccording(int A1[], int A2[], int m, int n)

{

int temp[m], visited[m];

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

temp[i] = A1[i];

visited[i] = 0;

}

sort(temp, temp + m);

int ind = 0;

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

int f = first(temp, 0, m - 1, A2[i], m);

if (f == -1)

continue;

for (int j = f; (j < m && temp[j] == A2[i]); j++) {

A1[ind++] = temp[j];

visited[j] = 1;

}

}

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

if (visited[i] == 0)

A1[ind++] = temp[i];

}

void printArray(int arr[], int n)

{

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

cout << arr[i] << " ";

cout << endl;

}

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

int m,n;

cin>>m;

cin>>n;

int A1[m], A2[n];

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

cin>>A1[i];

}

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

cin>>A2[i];

}

sortAccording(A1, A2, m, n);

printArray(A1, m);

return 0;

}