**Stevie !**

Attempted by: **943**

/

Accuracy: **49%**

/

Maximum Score: **20**

/

14 Votes

Tag(s):

Easy, Map

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

[Stevie G](https://en.wikipedia.org/wiki/Steven_Gerrard) is one of the greatest ever players to have played football, and he's going to be the one who recites all problems to you today . Let's have a look at him :



Now, he is a true admirer of mathematical geniuses like you. So he has a series of programming problems, among which the first one is :

You have been given 22 integer arrays A[]A[] and B[]B[] each of size NN. Now we call a pair of indices (i,j)(i,j) connected, if i=ji=j or A[i]=A[j]A[i]=A[j] .

Now, for each index ii in the array AA where 1≤i≤N1≤i≤N, you need to find the maximum B[j]B[j] such that indices ii and jj are connected. Can you do it ?

**Input Format** :

The first line contains a single integer NN.

The next line contains NN space separated integers, where the ithith integer denotes A[i]A[i]. The next line too contains NN space separated integers, where the ithith integer on this line denotes B[i]B[i].

**Output Format** :

Print NN space separated integers according to those mentioned in the problem statement.

**Constraints** :

1≤N≤200,0001≤N≤200,000

1≤A[i],B[i]≤1091≤A[i],B[i]≤109

**SAMPLE INPUT**

5

1 2 3 2 1

6 7 8 7 6

**SAMPLE OUTPUT**

6 7 8 7 6

**Explanation**

Indices 1 and 5 are connected. Therefore the answer for indices 1 and 5 will be whichever has a larger value of B[].

Similarly, indices 2 and 4 are connected.

Index 3 is not connected to any other index.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala 2.11.8, Swift, Visual Basic

<https://www.hackerearth.com/practice/basic-programming/implementation/basics-of-implementation/practice-problems/algorithm/stevie/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

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

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

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

Dictionary<int, int> diccio = new Dictionary<int, int>();

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

{

if (diccio.ContainsKey(a[i]))

{

diccio[a[i]] = Math.Max(diccio[a[i]], b[i]);

}

else

{

diccio[a[i]] = b[i];

}

}

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

{

Console.Write(diccio[a[i]] + " ");

}

Console.WriteLine();

Console.ReadLine();

}

}

}