

题目描述：

某学校举行运动会，学生们按编号(1、2、3...n)进行标识，现需要按照身高由低到高排列，对身高相同的人，按体重由轻到重排列；对于身高体重都相同的人，维持原有的编号顺序关系。请输出排列后的学生编号。

输入描述：

两个序列，每个序列由 n 个正整数组成 ($0 < n \leq 100$)。第一个序列中的数值代表身高，第二个序列中的数值代表体重。

输出描述：

排列结果，每个数值都是原始序列中的学生编号，编号从 1 开始

补充说明：

示例 1

输入：

```
4
100 100 120 130
40 30 60 50
```

输出：

```
2 1 3 4
```

说明：

输出的第一个数字 2 表示此人原始编号为 2，即身高为 100，体重为 30 的这个人。由于他和编号为 1 的人身高一样，但体重更轻，因此要排在 1 前面。

示例 2

输入：

```
3
90 110 90
```

45 60 45

输出：

1 3 2

说明：

1 和 3 的身高体重都相同，需要按照原有位置关系让 1 排在 3 前面，而不是 3 1 2

```
import java.util.Scanner;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;
import java.util.stream.Collectors;
// 注意类名必须为 Main，不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int[][] arr = new int[n][3];
        for (int i = 0; i < n; i++) {
            arr[i][0] = i + 1;
            arr[i][1] = in.nextInt();
        }
        for (int i = 0; i < n; i++) {
            arr[i][2] = in.nextInt();
        }
        List<Integer> ans = Arrays.stream(arr).sorted((e1, e2) -> e1[2] - e2[2]).sorted((e1, e2) ->
e1[1] - e2[1]).map(e->e[0]).collect(Collectors.toList());
        for (int i : ans) {
            System.out.print(i+" ");
        }
    }
}
```