


Synchronizing Lists

Problem ID: synchronizinglis**CPU Time limit:** 1 second**Memory limit:** 1024 MB**Difficulty:** 1.6**Author:** Greg Hamerly**Source:** Baylor Competitive Learning course**License:** 

It is so difficult to keep everything in order! For this problem, you have two lists of numbers, each list having length n . Each number in one list has a corresponding number in the other list. The correspondence is that if both lists were in sorted and placed side by side, each pair of numbers would correspond. For example, in the lists (48, 10, 97) and (7, 46, 20), the corresponding numbers are 10 with 7, 48 with 20, and 97 with 46.

You want to keep the lists synchronized, but you know that the second list has gotten out of order. So you want to put it in the same order as the first list, according to the correspondences.

For example, given the two lists above, the second list should be reordered as (20, 7, 46) to be in order with the first list.

Input

Input consists at most 100 test cases. Each test case begins with an integer $1 \leq n \leq 5\,000$, followed by $2n$ lines. The first n lines are the first list (which is in a fixed order), and the second n lines are from the second list (which may be out of sync with the first list). All integers given in the lists are in the range $[-10\,000, 10\,000]$. Input ends when $n = 0$. No list contains duplicates.

Output

For each test case, print out the second list so that it has the same ordering as the first list. Print a blank line between lists.

Sample Input 1

```
4
10
67
68
28
55
73
10
6
7
98
23
61
49
1
79
9
1
15
32
47
68
39
24
0
```

Sample Output 1

```
6
55
73
10

68
24
39
32
1
47
15
```