Sigma Arrays

You are given two arrays ***A*** and ***B***, each containing ***N*** number of distinct positive integers. In other words, ***A*** and ***B*** are sets of positive integers such that $n(A) = n(B) = N$. But a number in ***A*** can appear in ***B*** and vice versa.

For each integer (A_i) in ***A***, you need to pair it with an integer (B_j) in ***B***. You are not allowed to pick the same integer (B_j) from ***B*** more than once.

If it's possible to create ***N*** pairs such that the sum of each pair is the same, then the two arrays are considered as **Sigma Arrays**.

Given two arrays, your task is to determine whether they are **Sigma Arrays**, AND if they are **Sigma Arrays**, create ***N*** pairs of integers that display this quality.

Input Format

First line contains a single integer ***N***, the number of elements in each array.
Next line contains ***N*** integers, the elements of the array ***A***, with i^{th} of them being A_i.
Last line contains ***N*** integers, the elements of the array ***B***, with i^{th} of them being B_i.

Constraints

- $1 \leq N \leq 10^5$
- $1 \leq A_i, B_i \leq 10^9$

Limits

- **Time Limit**: 1s
- **Memory Limit**: 256MB

Output Format

If they **are Sigma Arrays**, print ***N*** lines, each containing a pair (A_i, B_j), sorted in the increasing order of A_i.
If the two arrays are **not Sigma Arrays**, print ***-1***

Refer the samples for a clearer picture.

Sample Input 0

```
8
20 10 11 4 8 3 1 5
29 27 25 22 26 20 19 10
```

Sample Output 0

```
1 29
3 27
4 26
5 25
8 22
10 20
```

```
11 19
20 10
```

Sample Input 1

```
10
2 12 4 5 6 8 15 3 45 99
1 5 3 24 15 13 48 56 32 10
```

Sample Output 1

```
-1
```