Problem F
Follower Forensics

Your social media company ConnectHub prides itself on how easily information spreads through its network.

Accounts can share posts directly with their followers using the share button, or with the accounts they follow by posting in the comments. ConnectHub is famous for the virality of its users: a post from any account was able to reach every other account, possibly through a chain of shares and comments.

But disaster struck: a database mishap wiped out all records of who follows whom. One piece of data may have survived, though — for each account, you have the number of its followers and the number of accounts it follows. Maybe even those numbers have been corrupted, but it's all you have.

Your task is to rebuild the missing database so that these counts are correct, and ConnectHub once again has full virality.

Input

The input consists of:

- One line with an integer n, the number of accounts, where $1 \leq n \leq 100\,000$.
- One line with n integers $a_1, \ldots, a_n$ satisfying $0 \leq a_i \leq n$ for each i, where a_i is the number of accounts followed by the ith account.
- One line with n integers $b_1, \ldots, b_n$ satisfying $0 \leq b_i \leq n$ for each i, where b_i is the number of accounts following the ith account.

Additionally, it is guaranteed that

$$\sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i \leq 2\,000\,000.$$

Note that the input is not guaranteed to allow for any reconstruction, fully viral or not. For those familiar with that terminology, full virality is equivalent to weak connectivity.

Output

If a reconstruction is possible, output a network

- One line with two integers: the number n of accounts, and the number m of connections. This is the same n as in the input.
- Then m lines, each consisting of two integers a and b, specifying that account a follows account b.

Otherwise output "`impossible`".

Note that it is not allowed for an account to follow the same account twice, nor to follow itself.

Sample Input 1

```
5
2 1 2 1 3
0 2 2 3 2
```

Sample Output 1

```
5 9
2 4
4 5
3 5
3 2
5 2
1 3
1 4
5 3
5 4
```

Sample Input 2

```
3
2 2 1
1 1 2
```

Sample Output 2

```
impossible
```