BFS-DFS

Time limit: *1000 ms*
Memory limit: *128 MB*

In this problem you are given the two orders of visiting the nodes in a BFS and a DFS, starting in node 1. Generate the edge list of a **simple, undirected, connected** graph corresponding to these orders.

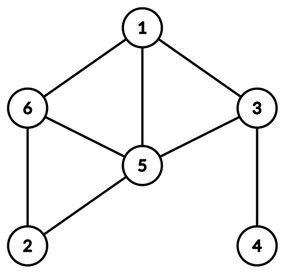
Statement clarification

One of the standard ways of storing a graph is using adjacency lists. Usually, the input is given as a list of edges. The program reads the numbers of nodes and edges, creates an empty list for each node, and then proceeds to read the edges. When an edge (a, b) is read, a is appended to the list of b and b is appended to the list of a. Consider the following input:

```
6 8
1 3
1 5
1 6
2 5
2 6
3 4
3 5
5 6

```

We have a graph with 6 nodes and 8 edges:



And we create the following lists:

- Node 1: $[3, 5, 6]$
- Node 2: $[5, 6]$
- Node 3: $[1, 4, 5]$
- Node 4: $[3]$
- Node 5: $[1, 2, 3, 6]$
- Node 6: $[1, 2, 5]$

In this problem we are concerned with the two most popular algorithms for traversing a graph: the Breadth First Search (BFS) and the Depth First Search (DFS).

The BFS pushes the starting node in a queue. While the queue is not empty, we pop the first node from the queue, go through its adjacency list, and enqueue the unvisited neighbours.

The DFS is usually implemented as a recursive function. First we call the function for the starting node. For each call, we go through the adjacency list of the current node, and recursively call the function for the unvisited neighbours.

Notice that the order of visiting the nodes is uniquely determined for both algorithms.

Standard input

The first line contains a single integer N, representing the number of nodes.

The second line contains a permutation of size N, representing the order of visiting the nodes for the BFS.

The third line contains a permutation of size N, representing the order of visiting the nodes for the DFS.

Standard output

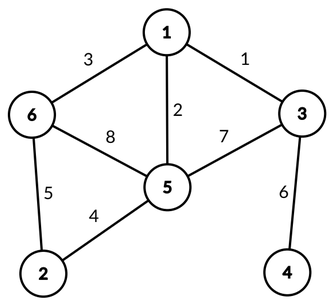
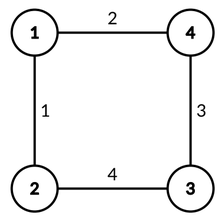
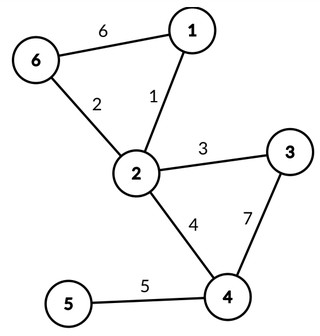
If there is no solution, output -1.

Otherwise, print a single integer M, representing the number of edges, on the first line.

Each of the next M lines should contain two integers a and b representing two nodes that share an edge.

Constraints and notes

- $1 \leq N \leq 4096$
- The number of edges M should be at most 10^5
- The two permutations will always start with 1
- The graph is considered to be undirected. Multiple edges and self loops are not allowed, and the graph should be connected.

Input	Output	Explanation
6 1 3 5 6 4 2 1 3 4 5 2 6	8 1 3 1 5 1 6 2 5 2 6 3 4 3 5 5 6	The labels on the edges represent the indices in the output edges. 
4 1 2 4 3 1 2 3 4	4 1 2 1 4 3 4 2 3	
6 1 2 6 3 4 5 1 2 6 3 4 5	7 1 2 2 6 2 3 2 4 4 5 1 6 3 4	
8 1 3 5 4 7 8 2 6 1 3 7 2 8 6 5 4	10 2 8 1 3 1 5 3 7 2 7 1 4 4 2 2 6 8 6 5 8