Problem E
Eulerian Flight Tour
Time Limit: 3 seconds

You have an airline route map of a certain region. All the airports in the region and all the *non-stop routes* between them are on the map. Here, a non-stop route is a flight route that provides non-stop flights in both ways.

Named after the great mathematician Leonhard Euler, an *Eulerian tour* is an itinerary visiting all the airports in the region taking a single flight of every non-stop route available in the region. To be precise, it is a list of airports, satisfying all of the following.

- The list begins and ends with the same airport.
- There are non-stop routes between pairs of airports adjacent in the list.
- All the airports in the region appear *at least once* in the list. Note that it is allowed to have some airports appearing multiple times.
- For all the airport pairs with non-stop routes in between, there should be *one and only one adjacent appearance* of two airports of the pair in the list in either order.

It may not always be possible to find an Eulerian tour only with the non-stop routes listed in the map. Adding more routes, however, may enable Eulerian tours. Your task is to find a set of additional routes that enables Eulerian tours.

Input

The input consists of a single test case.

n m
a_1 b_1
$\vdots$
a_m b_m

n ($3 \leq n \leq 100$) is the number of airports. The airports are numbered from 1 to n. m ($0 \leq m \leq \frac{n(n-1)}{2}$) is the number of pairs of airports that have non-stop routes. Among the m lines following it, integers a_i and b_i on the i-th line of them ($1 \leq i \leq m$) are airport numbers between which a non-stop route is operated. You can assume $1 \leq a_i < b_i \leq n$, and for any $i \neq j$, either $a_i \neq a_j$ or $b_i \neq b_j$ holds.

Output

Output a set of additional non-stop routes that enables Eulerian tours. If two or more different sets will do, any one of them is acceptable. The output should be in the following format.

k
c_1 d_1
$\vdots$
c_k d_k

k is the number of non-stop routes to add, possibly zero. Each of the following k lines should have a pair of integers, separated by a space. Integers c_i and d_i in the i-th line ($c_i < d_i$) are airport numbers specifying that a non-stop route is to be added between them. These pairs, (c_i, d_i) for $1 \leq i \leq k$, should be distinct and should not appear in the input.

If adding new non-stop routes can never enable Eulerian tours, output `-1` in a line.

Sample Input 1	Sample Output 1
4 2 1 2 3 4	2 1 4 2 3

Sample Input 2	Sample Output 2
6 9 1 4 1 5 1 6 2 4 2 5 2 6 3 4 3 5 3 6	-1

Sample Input 3	Sample Output 3
6 7 1 2 1 3 1 4 2 3 4 5 4 6 5 6	3 1 5 2 4 2 5

Sample Input 4	Sample Output 4
4 3 2 3 2 4 3 4	-1

Sample Input 5	Sample Output 5
5 5 1 3 1 4 2 4 2 5 3 5	0