Task 3. Domino

The set for playing Domino is composed of rectangular pieces of size 2 x 1, each of them divided in two equal halves by a straight line parallel to the shorter side of the rectangular. On each of the two halves dots are drown. The number of dots in one half varies from 0 to M, inclusive. The pieces of a domino set have all possible different unordered pairs of numbers. For example, if $M = 3$, then the corresponding domino full set contains 10 pieces: {0, 0}, {0, 1}, {0, 2}, {0, 3}, {1, 1}, {1, 2}, {1, 3}, {2, 2}, {2, 3}, {3, 3}. The pieces of the domino set can be arranged in *chains*. Two pieces can be joined by their shorter sides when the corresponding halves have the same number of dots drawn on them.

Suppose that N pieces are removed from the full set (but not all of them). It is interesting to know the minimal number of chains that it is possible to build so that each piece is included in exactly one chain. Write a program **domino** to solve the task: by given M and the list of the removed pieces you have to find the minimum number of chains with the described property.

Input

The first line of the **standard input** will contain the maximum number M of dots that are drawn on one half of a piece and the number N of removed pieces. It follows N lines and the i-th of these lines contains the numbers A_i and B_i – the number of dots on both halves of the i-th removed piece.

Output

The program has to print on the first line of the **standard output** the found minimum number of chains V. Each of the next V lines has to contain one of the found chains, presented as a sequence of numbers from the interval from 0 to M in which each two consecutive numbers are the dots of the two parts of the current piece. The numbers in each line are separated by one space. Each sequence has to end with -1.

Constraints:

$0 \le M \le 1024$

Example

Input	Output
3 5	1
0 2	2 2 3 0 0 1 -1
1 1	
1 2	
1 3	
3 3	

Explanation:

The chain of the pieces is: {2,2}, {2,3}, {3,0}, {0,0}, {0,1}.