

Task: DOM

Domino

You are given a chessboard of size $n \times m$. There are also some lines drawn on it. Each line separates two adjacent fields. You are going to put dominoes on the chessboard. Each domino covers two adjacent fields. You can put a domino on two adjacent fields only if they are not separated by a line. Your task is to find such an arrangement of dominoes on the chessboard, that each field is covered by exactly one domino. You can assume that for each input data there is a solution.

Task

This is an output-only task. This means, that you are given input files `dom1.in`, `dom2.in`, ..., `dom20.in`. Your task is to produce output files `dom1.out`, `dom2.out`, ..., `dom20.out`. Your submission should comprise a single `zip` or `tar-gzip` archive containing all the output files without any sub-directories. The format of input files, as well as the format of output files, which you are to produce, is described below.

Input

The first line of an input file contains two integers, n and m , separated by a single space — n is the number of rows and m is the number of columns of the chessboard, $1 \leq n, m \leq 100$, $n \cdot m$ is even. The fields of the chessboard are numbered from 1 to $n \cdot m$. The i -th field (from the left) in the j -th row (from the top) has number $(j - 1) \cdot m + i$ (for $1 \leq i \leq m$, $1 \leq j \leq n$).

The second line contains one positive integer l , $0 \leq l \leq 5000$. Each of the following l lines contains two integers separated by a single space. The $i + 2$ -nd line contains integers p_i and q_i (for $i = 1, \dots, l$), where $1 \leq p_i, q_i \leq n \cdot m$, p_i and q_i are numbers of two adjacent fields. It represents a line between fields number p_i and q_i .

Output

A single output file should consist of $\frac{n \cdot m}{2}$ lines describing an arrangement of the dominoes, one domino per line. Each of these lines should contain two integers separated by a single space: numbers of two adjacent fields covered by a domino. The dominoes may be described in any order. If there are several solutions, you should find any one of them.

Example

For the input data:

4 5
9
8 7
13 14
14 19
6 7
12 7
4 9
12 13
14 9
9 10

the correct result is:

3 4
1 6
2 7
8 9
5 10
14 15
11 16
12 17
13 18
19 20