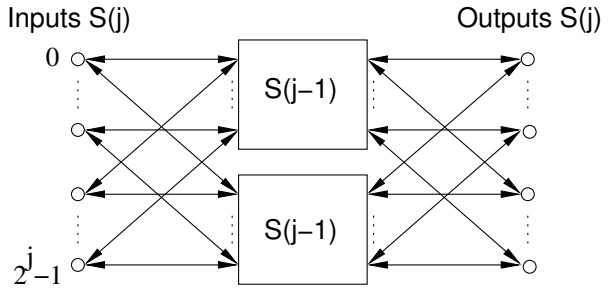
I Telephone Network

A telephone company wants to build a new telephone network in a city. The company has the goal that each person in the city should be able to call each other person. Of course, it is not possible to build direct connections between every pair of persons. Instead, the company uses a network made up of several layers.

We denote a network switch in layer j by $S(j)$. A switch $S(0)$ consists of one input, one output and a cable connecting the input to the output. A switch $S(j)$ with $j > 0$ consists of 2^j inputs, 2^j outputs and two switches $S(j-1)$. Input i of $S(j)$ ($0 \leq i < 2^j$) is connected via a cable to the inputs i mod 2^{j-1} of each of the two switches $S(j-1)$. Similarly, output i of $S(j)$ is connected to the outputs i mod 2^{j-1} of each of the two switches $S(j-1)$.



The connections between a switch $S(j)$ and the two switches $S(j-1)$ it consists of.

We are considering a network with one switch $S(n)$ in the outermost layer. It can be shown that any input and output of switch $S(n)$ has a unique connection path to any of the $S(0)$ switches. Therefore, any input of $S(n)$ can be connected to any of its outputs, and the connection path is uniquely determined by specifying through which switch $S(0)$ the connection is established.

We number the switches $S(0)$ belonging to the switch $S(n)$ from 0 to $2^n - 1$. The i-th switch $S(0)$ is defined as follows. Write the number i in binary as $b_{n-1}b_{n-2}\ldots b_0$. This defines a path from an input of $S(n)$ to the i-th switch $S(0)$ via the following procedure: for each j, $b_j = 0$ indicates that the path extends from $S(j+1)$ to the first $S(j)$ switch to which it is directly connected, and $b_j = 1$ indicates that the path extends to the second $S(j)$ switch. Note that regardless of which input of $S(n)$ is selected, this path arrives at the same $S(0)$ switch, which is given the number i. See also the figure below the sample data for details of how the numbering works.

Sometimes multiple connections are needed at the same time. In order to avoid interference, each of the inputs and outputs of all switches $S(j)$ ($0 \leq j \leq n$) can be used by at most one connection. Given a set of connection requests, can you find connection paths for each request such that the connection paths are disjoint?

Input

On the first line a positive integer: the number of test cases, at most 100. After that per test case:

- One line with two integers n ($1 \leq n \leq 16$) and m ($1 \leq m \leq 2^n$): the layer of the outermost switch and the number of connection requests.

- m lines, each with two integers a_i and b_i ($0 \leq a_i, b_i < 2^n$). Each such line represents a connection request from input number a_i of $S(n)$ to output number b_i. You may assume that the integers a_i are pairwise distinct, and the integers b_i are pairwise distinct as well.

Output

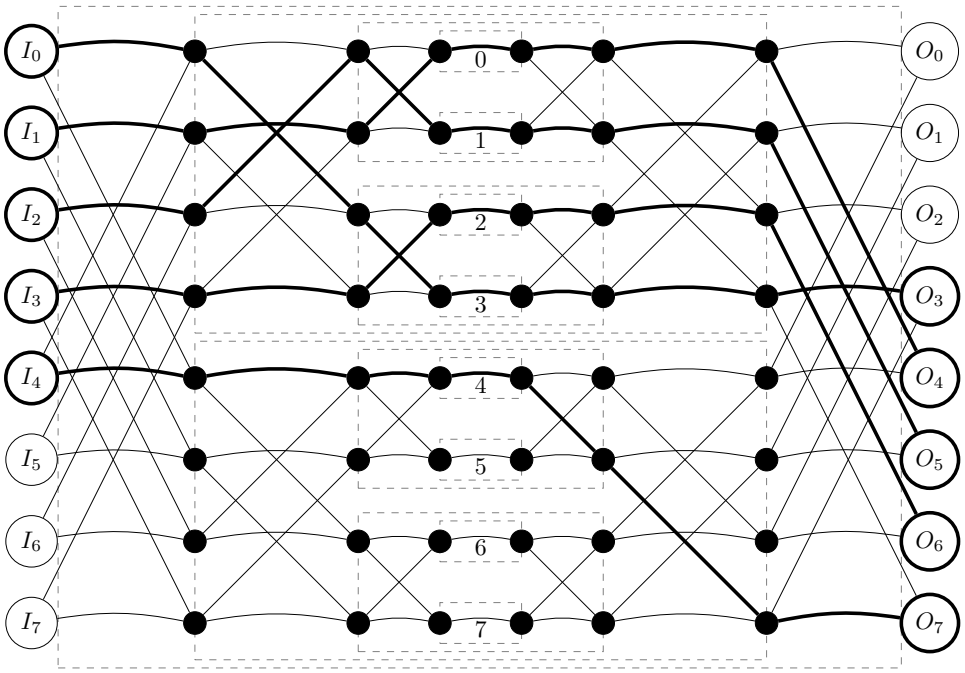
Per test case:

- One line with m integers $s_1, \ldots, s_m$, where s_i is the number of the $S(0)$ switch through which the connection of input a_i to output b_i is established.

The connection paths should be disjoint. You may print any valid solution, and you may assume that there is at least one valid solution.

Sample in- and output

Input	Output
2 1 1 0 1 3 5 0 3 1 4 2 5 3 6 4 7	0 3 0 1 2 4



A possible connection scheme for the second sample case, with used cables in bold.