Problem C. Topological Ordering

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

The topological ordering of a directed acyclic graph is a permutation of its vertices $p_1, \ldots, p_n$ such that for each arc, its source comes before its target in the permutation.

You are given a directed acyclic graph. For each pair of vertices (u, v) count the number of topological orderings such that vertex u comes before vertex v.

Input

The first line contains a single integer t, the number of test cases. Descriptions of t test cases follow.

In the first line of each test case there are two integers n and m: the number of vertices and arcs ($1 \le n \le 20$, $0 \le m \le n \cdot (n-1)/2$).

Each of the next m lines contains two integers u_i and v_i, denoting the arc from vertex u_i to vertex v_i ($1 \le u_i < v_i \le n$).

There are at most 100 test cases in the input. In at most 5 test cases $n > 10$.

Output

For each test case, print n lines of n numbers each. The j-th number in the i-th line should equal the number of topological orderings where vertex j is before vertex i. In particular, it should equal 0 if $i = j$.

Example

standard input	standard output
2 3 2 1 2 1 3 4 2 1 2 3 4	0 0 0 2 0 1 2 1 0 0 0 3 1 6 0 5 3 3 1 0 0 5 3 6 0