Write a program to compute the transitive closure of a directed graph with \( n \) vertices. That is, you must compute an \( n \times n \) matrix where at the \( j \)-th column of the \( i \)-th row there is a 1 if it is possible to go from \( i \) to \( j \), and there is a 0 otherwise.

**Input**

Input consists of several cases. Every case begins with \( n \) followed by the number of arcs \( m \). Follow \( m \) pairs \( x \ y \) to indicate an arc from \( x \) to \( y \), with \( x \neq y \). Assume \( 1 \leq n \leq 200 \), that the vertices are numbered between 0 and \( n - 1 \), and that there are no repeated arcs.

**Output**

For every graph, print its transitive closure, followed by a line with 20 dashes.

**Observation**

In the “large” private test cases, we have \( m = \Theta(n^2) \).

**Sample input**

\[
\begin{array}{c}
2 1 \\
0 1 \\
1 0 \\
4 5 \\
1 0 2 3 3 1 2 1 3 0
\end{array}
\]

**Sample output**

\[
\begin{array}{c}
1 1 \\
0 1 \\
1 \\
1 0 0 \\
1 1 0 0 \\
1 1 1 1 \\
1 1 0 1
\end{array}
\]