Problem I. DAG Query

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

This is an interactive problem.

Given a directed acyclic graph (DAG) with n vertices and m edges, each edge has an associated weight.

There may be multiple paths between any pair of vertices s and t. We define the weight of a path as the product of the weights of all edges along that path. Let $f(s, t, c)$ denote the sum of the path weights of all distinct paths from s to t after multiplying the weight of every edge in the graph by c. Since the result can be extremely large, $f(s, t, c)$ is taken modulo 998244353.

Xiao M will inform you of the structure of the DAG in advance (excluding the edge weights).

You may provide parameters s, t, and c; the interactor will return the value of $f(s, t, c)$. You are allowed to make at most 999 such queries to obtain certain information about the graph. After several query cycles, the interactor will provide a parameter k, and you need to determine the value of $f(1, n, k)$.

Input

The first line contains two positive integers n and m ($1 \le n \le 1000, 1 \le m \le 5000$), where n represents the number of vertices in the graph and m represents the number of edges.

In the following m lines, each line contains two positive integers x and y ($1 \le x, y \le 1000$), indicating that there is a directed edge from vertex x to vertex y in the graph. It is guaranteed that the graph is a Directed Acyclic Graph (DAG).

Interaction Protocol

For each query, you need to output in the format `? s t c` ($1 \le s, t \le n, 0 \le c < 998244353$) where s, t, c are all integers, and the interactor will return the answer of $f(s, t, c)$.

When you are confident that the information you have is sufficient to answer the query, output `!`, and the interactor will return a integer parameter k ($0 \le k < 998244353$), which means you are asked for the weight of $f(1, n, k)$.

You need to output the result of $f(1, n, k)$.

After printing a query or the answer, do not forget to output the end of the line and flush the output. Otherwise, you will get the verdict Idleness Limit Exceeded. To do this, use:

`fflush(stdout)` or `cout.flush()` in C++; `System.out.flush()` in Java; `flush(output)` in Pascal; `stdout.flush()` in Python; see the documentation for other languages.

Example

standard input	standard output
3 3 1 2 2 3 1 3 1 2 12 123	? 1 2 1 ? 2 3 1 ? 1 3 1 ! 31488