Problem C. Too Many Edges

Input file:	*standard input*
Output file:	*standard output*
Time limit:	15 seconds
Memory limit:	512 mebibytes

This is an interactive problem. You have to use the `flush` *operation right after printing each line. For example, you can use the function* `fflush(stdout)` *for C or C++,* `System.out.flush()` *for Java,* `flush(output)` *for Pascal, and* `sys.stdout.flush()` *for Python.*

You are an ordinary employee of a data analysis department. However, it seems that your colleagues are geniuses at making trouble. Just now they made some trouble again.

They downloaded a large graph from the remote server and wrote a program to analyze the graph. They nearly finished the analysis and thought the graph should be no longer useful. So they let their program add some new edges to the graph without any backups. But then they changed their mind and want to compute the longest path in the original graph. Downloading the graph again costs too much time. Now it is your time to save the day.

The original graph on the remote server is a directed acyclic graph $G = (V, E)$. All edges are unit length. The input of your program is the modified version $G' = (V, E')$. It is guaranteed that $E \subseteq E'$, **and G' is a directed acyclic graph**.

Your program should output $\ell(G)$, the length of the longest path in G. The length of a path equals the number of edges in the path.

To test whether an edge belongs to the original graph, your program can make queries to the remote server. To query the existence of edge $u \to v$, output "`? u v`". Flush the output stream after printing each query. The remote server will respond `1` if edge $u \to v$ belongs to the original graph, and `0` otherwise.

After your program gets the answer, print "`! ans`", where $ans = \ell(G)$, and **terminate your program normally** immediately after flushing the output stream.

Your program is allowed to make no more than $(|E'| - |E| + 1) \cdot (\ell(G) + 1)$ queries (not including printing the answer) to the remote server, although $|E|$ and $\ell(G)$ are unknown to you.

Input

Use standard input to read the responses to the queries.

The first line contains two integers n and m ($1 \le n \le 5 \cdot 10^4$; $1 \le m \le 10^5$): the number of vertices and edges of graph G'.

Each of the next m lines contains two integers u and v ($1 \le u, v \le n$) specifying a directed edge $u \to v$ in graph G'. No edge appears more than once.

It is guaranteed that $0 \le |E'| - |E| \le 200$.

The following lines will contain responses to your queries. Each response is either "`0`" or "`1`". The i-th of these lines is a response to your i-th query.

After answering $(|E'| - |E| + 1) \cdot (\ell(G) + 1)$ queries, the remote server no longer responds.

The testing system will allow you to read the response to a query only after your program prints the query and performs the `flush` operation.

Output

To make the queries, your program must use standard output.

Your program must print the queries in the form of "`? u v`", one query per line. Do not forget to end the line after each query. Your program must guarantee that $1 \le u, v \le n$. After printing each line your program must perform the `flush` operation.

The response to the query will be given in the standard input after you flush the output. In case your program finds the answer, print a line "! *ans*", where $ans = \ell(G)$, and terminate your program.

Example

standard input	*standard output*
5 5 1 2 1 3 2 5 3 4 4 5 1 1 1 0 1	? 1 2 ? 1 3 ? 2 5 ? 3 4 ? 4 5 ! 2