Problem H. Hidden Graph

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

There is a simple undirected graph with n vertices. This graph has one additional property:

- Any induced subgraph contains a vertex with a degree at most k.

You need to find this hidden graph. You can check for any subset whether it is an independent set. If it is not, we will show you an edge with both ends inside of this independent set.

We won't change the graph during the interaction, so you may assume that it is fixed.

However, we may choose any edge inside the induced subgraph to show.

In other words, in all tests, the graph is fixed, but the interactor is adaptive.

You need to guess the graph in at most $2nk + n$ queries.

Interaction Protocol

The interaction starts with a line containing one integer n: the number of vertices in the hidden graph ($2 \le n \le 2000$).

The integer k is not given, but it satisfies the constraint $1 \le k < n, nk \le 2000$.

After that, you can make queries.

To make a query, print one line containing "`?` k" ($1 \le k \le n$) and then k distinct integers $v_1, v_2 \ldots, v_k$ ($1 \le v_i \le n$). Separate consecutive integers by single spaces. Then `flush` the output.

After each query, read one line with two integers i, j. If there are no edges in the induced subgraph $v_1, v_2, \ldots, v_k$, $i = j = -1$, otherwise, $i \ne j, i \in v, j \in v$, and there is an edge with ends i and j in the graph.

When you find the graph, in the first line print one line containing "`!` m". And the next m lines should contain the description of edges of the graph, each of them should contain two integers u, v ($1 \le u, v \le n$), denoting the indices of vertices connected by an edge.

Your solution will get `Wrong Answer` or `Time Limit Exceeded` if you make more than $2nk + n$ queries.

Your solution will get `Idleness Limit Exceeded` if you don't print anything or forget to flush the output.

To `flush`, you need to do the following right after printing a query and a line break:

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

Example

standard input	standard output

3
1 2
2 3
1 3
	? 2 1 2
	? 2 2 3
	? 2 1 3
	! 3
	1 2
	2 3
	1 3