One Item Away

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	1024 megabytes

This is an interactive problem. Remember to flush the output buffer after every print. To flush your output, you can use:

- `fflush(stdout)` or `cout.flush()` in C/C++;
- `System.out.flush()` in Java and Kotlin;
- `sys.stdout.flush()` in Python.

There are n people and m items. Each item must be assigned to exactly one person.

Different people may value items differently. For any collection of items, a person can evaluate their satisfaction with it. You do not know exactly how these evaluations are computed, but you are allowed to query their values. You may ask at most $n \times m$ queries.

Formally, for each person i, there exists a function $u_i(S)$ that assigns a non-negative integer value to any subset S of items. These functions satisfy:

- $u_i(\varnothing) = 0$;
- If $S \subseteq T$, then $u_i(S) \leq u_i(T)$.

In other words, giving someone more items never makes them less satisfied.

However, people can be sensitive. If person i observes that another person j has a strictly better collection of items, they may start to complain.

Fortunately, people are also somewhat forgiving: if removing just one item from j's collection is enough to eliminate this advantage, then no complaint is made.

Formally, let s_i denote the set of items assigned to person i. Your goal is to assign every item to exactly one person so that, for every pair of people i and j, if person i considers j's collection strictly better than their own, i.e., $u_i(s_j) > u_i(s_i)$, then there exists an item $x \in s_j$ such that $u_i(s_j \setminus \{x\}) \leq u_i(s_i)$.

Input

There is only one test case in each test file.

The first line contains two integers n and m ($1 \leq n \leq 100$, $1 \leq m \leq 500$), indicating the number of people and the number of items.

Interaction Protocol

You can evaluate values using queries.

To ask for the value that person i assigns to a subset of items $\{x_1, x_2, \ldots, x_k\}$, output one line:

$$\texttt{?}\ i\ k\ x_1\ x_2\ \ldots\ x_k$$

where $1 \leq i \leq n$, $0 \leq k \leq m$, and $x_1, x_2, \ldots, x_k$ are pairwise distinct integers satisfying $1 \leq x_t \leq m$.

After flushing your output, your program should read a single integer v ($0 \leq v \leq 10^9$), which is equal to $u_i(\{x_1, x_2, \ldots, x_k\})$.

You may ask at most $n \times m$ queries. Also note that the interactor is adaptive. In particular, the answers to your queries may depend on your previous queries, but they are always consistent with some valid set of functions $u_i(\cdot)$ satisfying all constraints in the statement.

When you are ready to assign items, output one line:

$$\texttt{!}\ a_1\ a_2\ \ldots\ a_m$$

where a_j is the person to whom item j is assigned, and $1 \le a_j \le n$. Assigning items does not count as a query.

After flushing your output, your program should terminate immediately.

Example

standard input	standard output
2 3 10 20 10 5	 ? 1 1 1 ? 1 2 2 3 ? 2 1 1 ? 2 2 2 3 ! 2 1 1