

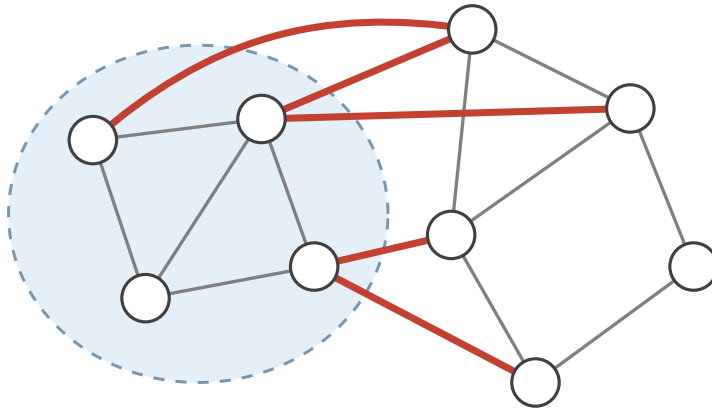
# Problem

## Guess the Graph

Time limit: 1 seconds

*This is an interactive problem.*

Little Cyan Fish has a hidden simple undirected graph with  $n$  vertices, labeled from 1 to  $n$ . You know the value of  $n$ , but you do not know any of the edges. Little Cyan Fish loves the concept of a *cut*. For a set of vertices  $S$ , let  $\text{cut}(S)$  denote the number of edges with exactly one endpoint in  $S$ .



Little Cyan Fish allows you to compare cuts by making queries. In one query, you choose two sets of vertices  $S_1$  and  $S_2$ . Little Cyan Fish tells you whether  $\text{cut}(S_1)$  is smaller than, equal to, or greater than  $\text{cut}(S_2)$ .

You can probably predict that the next token will be something about recovering the graph, but this time the task is easier. Little Cyan Fish gives you a vertex  $u$ , and he wants you to find all vertices adjacent to  $u$  in at most  $n + 10$  queries.

### Input

The first line of the input contains two integers  $n$  and  $u$  ( $1 \leq n \leq 1000$ ,  $1 \leq u \leq n$ ), indicating the number of vertices and the given vertex, respectively. Then, the interaction begins.

### Interaction Protocol

To make a query, print a line in the following format:

$$? \ s_1 \ s_2$$

where  $s_1$  and  $s_2$  are strings of length  $n$ , consisting only of characters 0 and 1. For each  $j \in \{1, 2\}$ , vertex  $i$  belongs to  $S_j$  if and only if the  $i$ -th character of  $s_j$  is 1. The sets may be empty, may overlap, and may be equal.

As a response to the query, you will get a single character  $<$ ,  $=$ , or  $>$ , meaning that  $\text{cut}(S_1)$  is less than, equal to, or greater than  $\text{cut}(S_2)$ , respectively. You must ensure that the number of queries you make does not exceed  $n + 10$ .

To report the answer, output a line in the following format:

$$! \ k \ v_1 \ v_2 \ \dots \ v_k$$

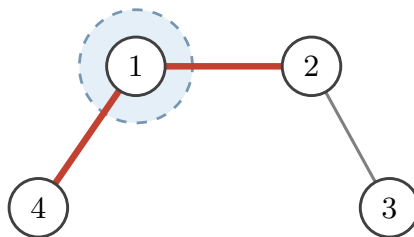
where  $k$  ( $0 \leq k \leq n - 1$ ) is the number of vertices adjacent to  $u$ , and  $v_1, v_2, \dots, v_k$  are their distinct labels in any order. If  $k = 0$ , output  $! \ 0$ . Reporting the answer does not count as a query. Your program must terminate immediately after reporting the answer.

After outputting a query or reporting the answer, do not forget to output a newline character and flush the output stream. If you make an invalid query or exceed the query limit, the interactor will close the input stream and your program must terminate.

In this problem, the interactor is adaptive. This means that the graph may change depending on your queries, as long as it does not contradict any previous response. Your answer must be correct for every graph consistent with all responses you have received.

| Read | Sample Interaction 1 | Write |
|------|----------------------|-------|
| 4 1  |                      |       |
|      | ? 0000 1000          |       |
| <    |                      |       |
|      | ? 0100 1100          |       |
| =    |                      |       |
|      | ? 0010 1010          |       |
| <    |                      |       |
|      | ? 0001 1001          |       |
| =    |                      |       |
|      | ! 2 2 4              |       |

One graph consistent with the sample interaction has edges (1, 2), (1, 4), and (2, 3).



**Testing Tool:** A testing tool is provided to help contestants develop and test their solutions. You can download this tool from the attachments. Run the tool with the `-h` option to display usage instructions. The tool checks the interaction and the reported answer, but does not enforce the time or memory limit.

The testing tool uses a fixed graph specified in its input file throughout the interaction, whereas the official interactor is adaptive. Passing tests with the testing tool does not guarantee that your solution is correct against the official interactor.