

## B Hamilton

Consider a directed graph with  $n$  nodes numbered  $1, 2, \dots, n$ . The graph is called a *tournament* if there is exactly one edge between all pairs of nodes in either direction. That is, for any pair of distinct nodes  $u$  and  $v$ , there is either an edge from  $u$  to  $v$  or from  $v$  to  $u$ .

A Hamiltonian cycle is a sequence  $c_1, c_2, \dots, c_n$  that visits all nodes and returns back to the start, following edges in the graph. For all  $1 \leq i \leq n - 1$ , there must be an edge from  $c_i$  to  $c_{i+1}$ . Additionally, there must be an edge from  $c_n$  to  $c_1$ .

You can freely construct a tournament graph of  $n$  nodes. The numbering of the nodes is then shuffled. By making queries on the edge directions in the shuffled graph, can you find a Hamiltonian cycle?

### Interaction

This is an interactive problem. Start by reading two integers  $n$  and  $t$ : the number of nodes and the number of test cases.

Next, print  $n$  lines to describe the tournament graph. On the  $u$ th of these lines, print  $n$  characters "0" or "1". A character "1" at position  $v$  indicates that there is an edge from  $u$  to  $v$ . There should be no edge from  $u$  to itself.

Then,  $t$  test cases follow. Each test case uses the same graph you provided, but the numbering is shuffled and kept secret by the grader. You may make some number of queries, after which you should report a Hamiltonian cycle.

To make a query, print "?  $u$   $v$ ", where  $1 \leq u, v \leq n$  are distinct nodes in the shuffled graph. The grader responds with either ">" if the edge is from  $u$  to  $v$ , or "<" if the edge is from  $v$  to  $u$ .

Once you have found a Hamiltonian cycle, print "!" followed by  $n$  integers  $c_1, c_2, \dots, c_n$ . Note that the numbers  $c_i$  should follow the shuffled numbering. The next test case begins immediately after you have printed the answer.

A testing script can be downloaded [here](#). The beginning of the script contains instructions on how to use it.

### Constraints

- $4 \leq n \leq 500$
- $1 \leq t \leq 200$

## Example interaction

5 2

```
01110
00101
00010
01001
10100
```

? 1 2

>

? 2 3

>

? 3 4

>

? 4 5

>

? 5 1

>

! 1 2 3 4 5

? 1 2

<

? 1 5

>

? 4 3

>

? 4 5

<

? 3 2

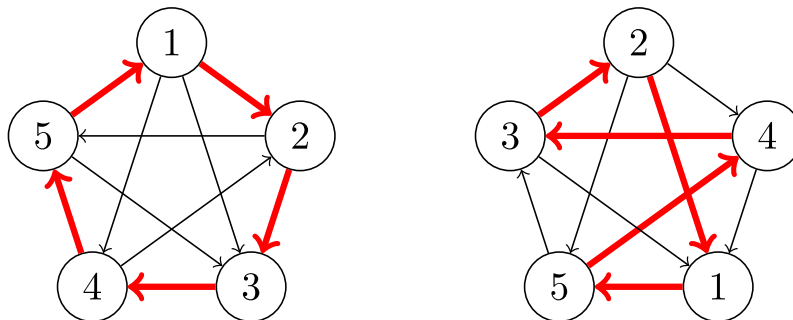
>

! 1 5 4 3 2

*Explanation:* The nodes happen to be shuffled to the original order in the first test case and therefore 1, 2, 3, 4, 5 is a Hamiltonian cycle.

In the second case, the node numbers 1, 2, 3, 4, 5 are shuffled to 2, 4, 1, 5, 3, in this order. The sequence 1, 5, 4, 3, 2 is indeed a Hamiltonian cycle because 3, 4, 2, 5, 1 was one in the original graph.

In the figure below, the original graph is shown on the left and the shuffled graph from the second test case is shown on the right. The two Hamiltonian cycles are highlighted in red.



## Scoring

There is only one test input in each subtask, with  $t = 200$  test cases. In each test case, the numbering of the graph is shuffled uniformly at random. Making more than  $10^4$  queries in a single test case results in the verdict WRONG ANSWER.

Let  $Q$  be the average number of queries made by your program among all test cases belonging to a subtask. You receive points for the subtask if  $Q$  is no greater than the specified limit.

Subtask	Constraints	Points
1	$n = 4, Q \leq 12$	5
2	$n = 50, Q \leq 1225$	7
3	$n = 50, Q \leq 300$	12
4	$n = 500, Q \leq 1500$	1-76

In subtask 4, you receive points according to the following formula:

$$\left\lfloor \frac{25\,000}{\max(750, Q) - 500} - 24 \right\rfloor$$

For example, if the average number of queries made by your program is  $Q = 1500$ , you get 1 point from the subtask. If  $Q = 1000$ , you get 26 points, and if  $Q = 750$ , you get 76 points.