

Problem C. Cloyster

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 512 megabytes

This is an interactive problem.

You just found yourself in a huge field of Cloyster specimens somewhere on the seabed off the coast of Japan. They're quite mad at you as you tried to catch one of them a moment ago. You must immediately find the leader of the pack and beg him for forgiveness. Where is he, though?

The field contains n^2 square cells grouped into n rows and n columns. Each cell contains a single Cloyster hidden in its own shell. Now, I can reveal to you a couple of little-known facts about Cloyster:

- The leader has the largest shell. *Obviously.*
- All Cloyster in the pack have shells of different sizes.
- Each Cloyster other than the leader has a neighbor with a larger shell in some cell adjacent to its own cell by an edge or by a corner.

You can query individual specimens for the sizes of their shells. Use this information to locate the leader of the pack. Be quick, though — if you don't find him in $3n + 210$ queries, Cloyster will run out of patience and they'll attack you!

Interaction Protocol

The first line contains a single integer n ($2 \leq n \leq 2000$) — the size of the field. Your solution should read it first. Afterwards, your program can make two types of requests:

- “? i j” ($1 \leq i, j \leq n$) — query the size of the shell of the specimen in the cell in the i -th row and the j -th column. You can use this query at most $3n + 210$ times. The response will be a single integer a_{ij} ($1 \leq a_{ij} \leq 10^9$) printed on a separate line.
- “! i j” ($1 \leq i, j \leq n$) — answer that the leader of the pack is in the i -th row and the j -th column. This should be your last query. Your solution should then terminate immediately.









The interactor isn't adaptive, i.e. the sizes of the shells are fixed before the run of your program and won't change between your queries. Remember to flush the output after printing each line!


























Examples

| standard input | standard output |
|----------------|-----------------|
| 3 | ? 1 1 |
| 1 | ? 2 3 |
| 4 | ? 3 2 |
| 8 | ? 3 3 |
| 9 | ? 2 2 |
| 5 | ! 3 3 |
| 5 | ? 4 4 |
| 2 | ! 1 1 |

Note

Here are the sizes of each Cloyster in both sample tests:

| | | |
|---|---|---|
|  |  |  |
| 1 | 2 | 3 |
|  |  |  |
| 7 | 5 | 4 |
|  |  |  |
| 6 | 8 | 9 |

| | | | | |
|--|---|--|--|--|
|  |  |  |  |  |
| 97 | 29 | 31 | 37 | 41 |
|  |  |  |  |  |
| 89 | 23 | 19 | 17 | 43 |
|  |  |  |  |  |
| 83 | 13 | 11 | 7 | 47 |
|  |  |  |  |  |
| 79 | 5 | 3 | 2 | 53 |
|  |  |  |  |  |
| 73 | 71 | 67 | 61 | 59 |

Note, that the communication in the second sample test is correct — your program is allowed to make a guess even if it isn't certain about the correctness of the answer.