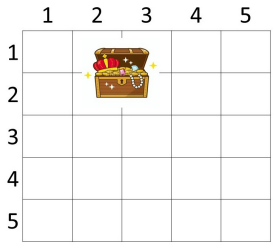
Problem K
Treasure Hunt
Time limit: 1 second

There is a 2×2 box of treasure hidden somewhere in a 5×5 grid. The rows are numbered from 1 to 5, as are the columns, with cell $(1, 1)$ being the top left corner of the grid. Your goal is to find the treasure within 5 queries. You can query a grid cell and ask if there is any part of the treasure chest in that grid cell. After at most 5 queries, you must output the upper left coordinate of the treasure box.



Interaction

Your program can make up to 5 queries for the location of the treasure. To make a query, print a new line of the form

`?` i j

where i and j are integers between 1 and 5. You are asking is the cell at the i^{th} row and j^{th} column has a part of the treasure box. After each query, the interactor will respond with a 0 if the cell does not contain any part of the treasure, or a 1 if the cell does.

After determining the correct answer, output the answer on a new line in the form

`!` i j

specifying the upper left coordinate of the treasure box. After outputting the answer, your program must exit. If you attempt to read a response after outputting an answer, you will receive an arbitrary verdict.

Do not forget to flush the output after each query you output.

If the interactor receives any invalid or unexpected input, the interactor will output -1 and then immediately terminate. Your program must cleanly exit in order to receive a correct verdict, otherwise the verdict that it receives may be an arbitrary verdict indicating that your submission is incorrect.

A testing tool is provided in python. Use it as follows (You may have to replace `python3` by just `python`):

```
python3 testing_tool.py However-you-run-your-program
```

The tool is provided as-is, and you should feel free to make whatever alterations or augmentations you like to it. The tool attempts to detect and report common errors, but it is not guaranteed that a program that passes the testing tool will be accepted.

Read	Sample Interaction 1	Write
		? 3 3
0		
		? 2 2
1		
		? 1 1
0		
		? 1 3
1		
		! 1 2