Problem D. Hidden Rook

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

This problem is interactive.

Roman hid a rook on an $n \times m$ chessboard. You need to find its exact position. You can ask Roman the following question **at most 4 times**: “How many cells (i, j), where $X_1 \le i \le X_2$ and $Y_1 \le j \le Y_2$, are under the hidden rook’s attack?” A rook attacks all cells in the same row or column, including its own cell.

Input

The first line contains an integer t, the number of test cases ($1 \le t \le 15\,000$).

Interaction Protocol

The interaction in each test case starts with two integers, n and m: the chessboard dimensions ($3 \le n, m \le 15$).

To ask Roman a question, print “`?` X_1 Y_1 X_2 Y_2” ($1 \le X_1 \le X_2 \le n$, $1 \le Y_1 \le Y_2 \le m$). After that, you will receive an integer K: the number of cells (i, j), where $X_1 \le i \le X_2$ and $Y_1 \le j \le Y_2$, that are under the hidden rook’s attack. You can ask at most 4 questions in each test case.

To report the answer, print “`!` X Y”, where (X, Y) is the hidden rook’s cell.

After making each query, do not forget to print the newline character and flush the output. You can use the following commands:

- fflush(stdout) or cout.flush() in C++;
- System.out.flush() in Java;
- flush(output) in Pascal;
- stdout.flush() in Python;

for other languages, see their documentation. You will get the “`Idleness limit exceeded`” verdict if you fail to do so.

Example

standard input	*standard output*
2	
6 6	
	? 1 1 3 6
8	
	? 2 2 2 3
2	
	! 2 3
7 5	
	? 1 1 7 5
11	
	? 1 1 1 4
4	
	! 1 4