

Worm Worries

You are looking for a place in the soil to put your pet worm, Maximus. You limit your search to a box-shaped region with dimensions $N \times M \times K$ centimeters which you have divided into a three-dimensional grid of cube-centimeter cells, labeled (x, y, z) after their position in the grid ($1 \leq x \leq N$, $1 \leq y \leq M$, $1 \leq z \leq K$). Each cell has a certain humidity $H[x, y, z]$ which is an integer in the range $1 \dots 10^9$. You can measure the humidity of any cell with a special sensor.

Maximus loves humid places, so you need to put him in a cell which is at least as humid as its neighboring cells, otherwise he goes away and you will have trouble finding him. In other words, you need to place Maximus in a local maximum. More precisely, you need to find a cell (x, y, z) , such that

$$H[x, y, z] \geq \max(H[x+1, y, z], H[x-1, y, z], H[x, y+1, z], H[x, y-1, z], H[x, y, z+1], H[x, y, z-1]),$$

where a value is treated as 0 when it is outside the box (because Maximus absolutely wants to stay in the box).

However, the number of cells can be very large, so you do not want to measure the humidity of all the cells. Therefore, in this task, you are allowed to interact with the grader, and ask for the humidity at given points. When you have found a suitable location for Maximus, give that location to the grader.

Interactivity

The first line of the input contains four positive integers: N , M , K and Q , where N , M and K are the box dimensions and Q is the maximum number of measurements you may perform.

After that, you can write at most Q lines of the form “? x y z” to standard output. This asks for the value of the humidity at point (x, y, z) . For each such line, the grader will in response write a single line with the integer $H[x, y, z]$, which can be read from standard input by your program.

After all these lines, your program must write out exactly one line of the form “! x y z” and terminate. This claims that the point (x, y, z) is a suitable location for Maximus according to the criterion above. The grader will provide no response to this output.

All values of x, y, z must obey $1 \leq x \leq N$, $1 \leq y \leq M$, $1 \leq z \leq K$. If they do not, or some line has an invalid format, or you ask for more than Q values, the grader will respond with -1 and exit. If this happens your program should also exit. If it continues, it may incorrectly get a verdict of Runtime Error or Time Limit Exceeded.

You *must* make sure to flush standard output before reading the grader’s response, or else your program will get judged as Time Limit Exceeded. This works as follows in the supported languages:

- Java: `System.out.println()` flushes automatically.
- Python: `print()` flushes automatically.
- C++: `cout << endl`; flushes, in addition to writing a newline. If using `printf`, `fflush(stdout)`.
- Pascal: `Flush(Output)`.

To help deal with this interaction, we provide optional helper code which you may copy into your program. A link to this code for all supported languages (C++, Pascal, Java, Python) can be found in the sidebar of the Kattis problem page. The helper code also uses fast input/output routines, which may be useful for Python and Java (only relevant for the last two test groups).

The grader will be *non-adaptive*; that is, each test case will have a fixed set of humidity values that do not depend on what measurements are performed by the program.

Constraints

Your solution will be tested on a set of test groups, each worth a number of points. Each test group contains a set of test cases. To get the points for a test group you need to solve all test cases in the test group. Your final score will be the maximum score of a single submission.

Group	Points	Limits
1	10	$M = K = 1, N = 1\,000\,000, Q = 10\,000$
2	22	$M = K = 1, N = 1\,000\,000, Q = 35$
3	12	$K = 1, N = M = 200, Q = 4\,000$
4	19	$K = 1, N = M = 1\,000, Q = 3\,500$
5	14	$N = M = K = 100, Q = 100\,000$
6	23	$N = M = K = 500, Q = 150\,000$

Sample dialogue

In Kattis there is one sample test case. In this sample, the box has dimensions $3 \times 1 \times 1$ and the humidity in the three cells is $\{10, 14, 13\}$. Below is an example dialogue, with the lines preceded by JUDGE being the output of Kattis (i.e. the input to your program), and the lines preceded by YOU being your program's output.

As 14 is indeed greater than or equal to the neighboring values (10 and 13), the location $(2, 1, 1)$ is a suitable location for Maximus, and you used three queries, which was the maximum allowed number ($Q = 3$). Thus, this dialogue gives Accepted on the sample.

```
JUDGE:  3 1 1 3
YOU:    ? 3 1 1
JUDGE:  13
YOU:    ? 2 1 1
JUDGE:  14
YOU:    ? 1 1 1
JUDGE:  10
YOU:    ! 2 1 1
```

Sample Input 1

```
3 1 1 3 123123 fixed 10 14 13
```

Sample Output 1

```
blah
```