Wow, It's Yesterday Six Times More

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

This is an interactive problem.

After the great success from 2018 to 2024, the Nanjing University of Aeronautics and Astronautics (NUAA) will host the *International Collegiate Programming Contest* (ICPC) Nanjing regional for the eighth time in a row.

Team ***Power of Two*** and team ***Three Hold Two*** won the champion title for Tsinghua University in 2018 and 2019. In 2020, 2021, and 2022, team ***Inverted Cross*** from Peking University won the three-peat champion titles, while in 2023, another team ***Reborn as a Vegetable Dog*** from Peking University won the title. In 2024, team ***Afterlife*** from Zhejiang University took the first place.

This year, around 335 teams are participating in the contest. At most 33 gold medals, 66 silver medals, and 99 bronze medals will be awarded (note that these numbers are for reference only). We are looking forward to seeing the participants' outstanding performance! We also want to express our gratitude for the hard work done by all staff and volunteers for this contest. Thank you all for your great contribution to this contest!

Photo taken in the 2023 ICPC Asia Nanjing Regional Contest

In the 2018 contest, problem K, ***Kangaroo Puzzle***, requires the contestants to construct an operation sequence for the game:

> The puzzle is a grid with n rows and m columns ($1 \le n, m \le 20$), and there are some (at least 2) kangaroos standing in the puzzle. The player's goal is to control them to get together. There are some walls in some cells, and the kangaroos cannot enter the cells with walls. The other cells are empty. The kangaroos can move from an empty cell to an adjacent empty cell in four directions: up, down, left, and right.
>
> There is exactly one kangaroo in every empty cell in the beginning, and the player can control the kangaroos by pressing the buttons U, D, L, R on the keyboard. The kangaroos will move simultaneously according to the button you press.
>
> The contestant needs to construct an operating sequence of at most 5×10^4 steps consisting of U, D, L, R only to achieve the goal.

In the 2020 contest, problem A, ***Ah, It's Yesterday Once More***, requires the contestants to construct an input map to hack the following code of the problem described before:

```
#include <bits/stdc++.h>
using namespace std;
string s = "UDLR";
int main()
{
  srand(time(NULL));
  for (int i = 1; i <= 50000; i++) putchar(s[rand() % 4]);
  return 0;
}
```

Furthermore, in the 2021 contest (Problem A, ***Oops, It's Yesterday Twice More***), the 2022 contest (Problem A, ***Stop, Yesterday Please No More***), the 2023 contest (Problem A, ***Cool, It's Yesterday Four Times More***), and the 2024 contest (Problem A, ***Hey, Have You Seen My Kangaroo?***), every year we have a problem related to the kangaroos! We would like to introduce all these problems to you, but if we do so every year, we may have a 500-page statement for one single problem in the 3025 contest. Therefore, we omit them this time. Besides, you may already have seen them in the practice contest.

Now, in the 2025 contest, as everyone expects, the kangaroo problem is back again! We don't know why problem setters are so obsessed with kangaroos, but the problem is as follows:

You are given a grid with n rows and m columns. There is a hole in the cell on the i_h-th row and the j_h-th column. All other cells are empty and there is one kangaroo standing in each cell.

Similarly, the kangaroos are controlled by pressing the button U, D, L, R on the keyboard. All kangaroos will move simultaneously according to the button pressed. Specifically, for any kangaroo located in the cell on the i-th row and the j-th column, indicated by (i, j):

1. Button U: it will move to $(i-1, j)$.
2. Button D: it will move to $(i+1, j)$.
3. Button L: it will move to $(i, j-1)$.
4. Button R: it will move to $(i, j+1)$.

If a kangaroo steps onto the hole (that is, $i = i_h$ and $j = j_h$) or steps out of the grid, it will be removed from the grid.

The problem is that, the exact value of i_h and j_h is not known. Your task is to find the position of the hole with at most $(n + m + 10)$ queries. For each query, you can press one button (U, D, L, or R). The interactor will output an integer t as the answer, indicating the number of kangaroos remaining after you press the button.

Note that the interactor is not adaptive, meaning that the answer for each test case is pre-determined.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 100$) indicating the number of test cases. For each test case:

The first line contains two integers n and m ($3 \le n, m \le 30$) indicating the number of rows and columns in the grid.

Interaction Protocol

To ask a query, output one line. First output `?` followed by a space, then print one upper-case English letter (U, D, L, or R). After flushing your output, your program should read a single integer t indicating the answer to your query. Recall that you can ask at most $(n + m + 10)$ queries for a test case.

If you want to guess the position of the hole, output one line. First output `!` followed by a space, then print two integers i_h and j_h ($1 \le i_h \le n$, $1 \le j_h \le m$) separated by a space indicating the position. After flushing your output, your program should continue processing the next test case, or exit immediately if there are no more test cases. Note that your guess does not count as a query.

To flush your output, you can use:

- `fflush(stdout)` (if you use `printf`) or `cout.flush()` (if you use `cout`) in C and C++.
- `System.out.flush()` in Java and Kotlin.
- `sys.stdout.flush()` in Python.

Example

standard input	standard output
2 3 4 7 3 4 3 8 5	 ? L ? D ! 2 3 ? U ? R ! 1 1