DMOPC '20 Contest 5 P3 - Bottom Row

Time limit: 2.0s **Memory limit:** 256M
Python: 512M

After typing continuously for three days and four nights, you've managed to increase your typing speed to a sizeable 35 WPM. However, you've gotten quite bored of transforming strings, so a quick scroll through the internet has led you to another typing game for a quick breather. In this game, you are given an $N \times N$ grid with K blocked cells and the others open. The cells are indexed with $(1, 1)$ as the bottom-left cell, and (N, N) as the top-right cell. All cases will satisfy the constraint that no two blocked cells are on the same row or column, and the cells $(1, 1)$ and (N, N) will not be blocked. There is currently a robot at cell $(1, 1)$, and you want to get it to cell (N, N). The robot reads a string of characters inputted by the user to determine its movements.

For a string of characters S, the robot will read the characters from left to right. If the i^{th} character is:

- `D`, the robot will move from cell (r, c) to cell $(r - 1, c)$.
- `U`, the robot will move from cell (r, c) to cell $(r + 1, c)$.
- `L`, the robot will move from cell (r, c) to cell $(r, c - 1)$.
- `R`, the robot will move from cell (r, c) to cell $(r, c + 1)$.

Your string should only contain the 4 characters listed above, and the robot should never attempt to move outside the grid or move into a blocked cell. As fast and efficient as ever, your goal is to find the shortest string of characters S which moves the robot from $(1, 1)$ to (N, N), or determine that no such string exists. If multiple shortest strings exist, print the lexicographically smallest one. A string x is lexicographically smaller than a string y of the same length if for some j, $x_i = y_i$ for all $i < j$, and $x_j < y_j$.

Constraints

$2 \le N \le 10^6$

$0 \le K \le N$

$1 \le r_i, c_i \le N$

Subtask 1 [30%]

$2 \le N \le 2 \times 10^3$

Subtask 2 [70%]

No additional constraints.

Input Specification

The first line contains 2 integers N and K, as described in the problem statement.

The next K lines each contain 2 integers r_i and c_i, representing that cell (r_i, c_i) is blocked. These cells will all be distinct and heed the constraints given in the statement. Specifically, no two blocked cells are on the same row or column, and the cells $(1, 1)$ and (N, N) will not be blocked.

Output Specification

If no string can move the robot from $(1, 1)$ to (N, N), output `IMPOSSIBLE`.

Otherwise output a string S, as described in the problem statement.

Sample Input 1

```
3 2
3 1
2 3
```

Sample Output 1

```
RUUR
```

Explanation for Sample 1

The following diagram depicts the given grid, with `#` denoting blocked cells and `.` denoting the rest.

```
#..
..#
...
```

It can be proven that `RUUR` is the shortest string that moves the robot from $(1, 1)$ to $(3, 3)$, and it is also the lexicographically smallest string among all shortest strings which achieve the same goal.

Sample Input 2

```
2 2
1 2
2 1
```

Sample Output 2

```
IMPOSSIBLE
```

Explanation for Sample 2

The following diagram depicts the given grid, with `#` denoting blocked cells and `.` denoting the rest.

```
#.
.#
```

It can be proven that no string can move the robot from $(1, 1)$ to $(2, 2)$.