Robot (robot)

There is a robot which is placed on a field modeled as a $n \times m$ grid. Some of these grid cells are walls.

The robot accepts 4 types of instructions: `up`, `down`, `left`, `right`.

Suppose the robot is currently at the coordinate (x, y). Then the effect of executing the instructions will be as follows:

- `up` : If $x = 0$ or $(x-1, y)$ is a wall, the robot does not move. Else, the robot moves to $(x-1, y)$
- `down` : If $x = n-1$ or $(x+1, y)$ is a wall, the robot does not move. Else, the robot moves to $(x+1, y)$
- `left` : If $y = 0$ or $(x, y-1)$ is a wall, the robot does not move. Else the robot moves to $(x, y-1)$
- `right`: If $y = m-1$ or $(x, y+1)$ is a wall, the robot does not move. Else the robot moves to $(x, y+1)$.

You know that the starting position of the robot is either (a, b) or (c, d). Find a sequence of at most q instructions such that the robot always ends up at $(0, 0)$ when the robot starts from either (a, b) or (c, d). It can be proven that there exists a solution for all inputs satisfying the problem constraint.

Implementation details

You should implement the following procedure:

```
void construct_instructions(bool[][] g, int q, int a, int b, int c, int d)
```

- g: a 2-dimensional array of size $n \times m$. For each $i, j (0 \leq i \leq n-1, 0 \leq j \leq m-1)$, $g[i][j] = 1$ if cell (i, j) is a wall, and $g[i][j] = 0$ otherwise.
- q: the maximum number of instructions that you are allowed to use.
- a, b, c, d: the possible starting location of the robot is either (a, b) or (c, d)

This procedure should call one or more of the following procedures to construct the sequence of instructions:

```
void up()
void down()
void left()
void right()
```

After appending the last instruction, `construct_instructions` should return.

Example

Consider the following call:

```
construct_instructions([[0,0],[0,1]], 700, 1, 0, 0, 1)
```

The grid has a single wall at $(1,1)$.

If the robot begins at $(1,0)$, the following happens when `up()` followed by `left()` is executed:

Action taken	New location	Remarks
`up()`	$(0,0)$	$(0,0)$ is not a wall
`left()`	$(0,0)$	Robot does not move left since $y = 0$

Similarly, if the robot begins at $(0,1)$, it remains at $(0,1)$ after `up()` and moves to $(0,0)$ after `left()`.

The program should call `up()` followed by `left()` before returning to output this construction.

Constraints

- $1 \leq n \leq 10$
- $1 \leq m \leq 10$
- $0 \leq a \leq n-1$
- $0 \leq b \leq m-1$
- $0 \leq c \leq n-1$
- $0 \leq d \leq m-1$
- $g[0][0] = g[a][b] = g[c][d] = 0$
- There exists a finite sequence of instructions to move the robot from (a,b) to $(0,0)$
- There exists a finite sequence of instructions to move the robot from (c,d) to $(0,0)$
- $q = 700$

Subtasks

1. (20 points) $n \leq 2$
2. (20 points) $g[i][j] = 0$ (for all $0 \leq i \leq n-1,\ 0 \leq j \leq m-1$)
3. (20 points) $a = c,\ b = d$
4. (40 points) No additional constraints

Sample grader

The sample grader reads the input in the following format:

- line 1 : $n\ m\ q\ a\ b\ c\ d$
- line $2 + i$ ($0 \le i \le n - 1$): $g[i][0]\ g[i][1] \ldots g[i][m-1]$

The sample grader prints the output in the following format:

- line 1: the number of instructions used
- line 2: a string containing all the instructions used. For every call to `up()`, `down()`, `left()`, `right()`, the grader will print a single character, `U`, `D`, `L` or `R` respectively.