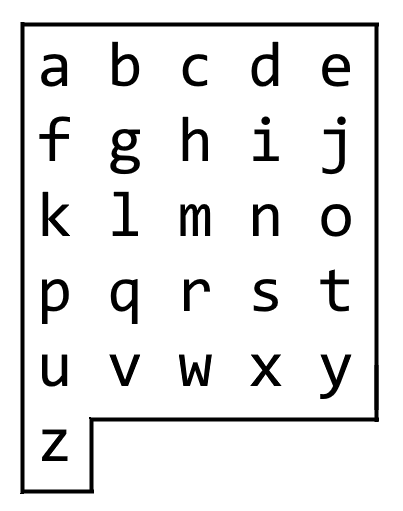
On an alphabet board, we start at position (0, 0), corresponding to character board[0][0].

Here, board = ["abcde", "fghij", "klmno", "pqrst", "uvwxy", "z"], as shown in the diagram below.



We may make the following moves:

* 'U' moves our position up one row, if the position exists on the board;
* 'D' moves our position down one row, if the position exists on the board;
* 'L' moves our position left one column, if the position exists on the board;
* 'R' moves our position right one column, if the position exists on the board;
* '!' adds the character board[r][c] at our current position (r, c) to the answer.

(Here, the only positions that exist on the board are positions with letters on them.)

Return a sequence of moves that makes our answer equal to target in the minimum number of moves.  You may return any path that does so.

**Example 1:**

**Input:** target = "leet"

**Output:** "DDR!UURRR!!DDD!"

**Example 2:**

**Input:** target = "code"

**Output:** "RR!DDRR!UUL!R!"

**Constraints:**

* 1 <= target.length <= 100
* target consists only of English lowercase letters.