Problem L. Labyrinth

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Dwarf the Puzzler is designing labyrinth puzzles for other dwarfs. A labyrinth is an $N \times N$ square with each cell being either a passage or a wall. In one of the passages is located a pawn, which faces one of four possible directions: right, up, left or down. In each step a dwarf first rotates the pawn and then moves it in the resulting direction. Of course the pawn cannot move to or through the wall! If the pawn moves out of the labyrinth, then the puzzle is solved.

Dwarfs love those puzzles, but Dwarf the Riddler, Puzzler's old rival, boasts that labyrinths are dull and that he can solve them using a simple tactic: *among the cells to the right, front, left and back (relatively to the direction faced by the pawn) choose the first one that is a passage, then rotate the pawn to this direction and move in this direction.* The Puzzler is not convinced that this indeed is a proper tactic and, as an engineer, wants to run some tests on it. Help him, for the labyrinths he devised and starting positions (and directions) of the pawn compute, whether the pawn will eventually leave the labyrinth and how many moves it will take.

Input

The first line of input contains two integers N and Q, representing the size of the labyrinth and the number of queries, respectively.

In the next N lines, there is a description of the labyrinth, with description of one labyrinth row per line, starting from top-most row. Each line contains N characters that are either a dot `.` or a hash `#`, representing a passage and wall, respectively. Rows and columns of the labyrinth are numbered from 1 to N, top-to-bottom and left-to-right.

The following Q lines contain descriptions of the subsequent queries. Each query consists of two integers r, c, and one character d. The numbers r and c represent the row and column where the pawn is initially located, and the character d indicates the direction the pawn is facing, where the letters `U`, `D`, `L` and `R` mean that the pawn is initially facing respectively up, down, left or right.

You can assume that the pawn is never located in a cell where there is a wall.

Limits $1 \le N \le 1\,000$, $1 \le Q \le 100\,000$, $1 \le r, c \le N$, $d \in \{\texttt{U}, \texttt{D}, \texttt{L}, \texttt{R}\}$.

Output

For each query, output one line containing a single integer: the number of moves in which the pawn will leave the labyrinth. If the pawn never leaves the labyrinth, output -1 instead.

Example

standard input	standard output
10 10 ########## #.#.#.#... #.....#.## ####.##..# #....##### #..#.#...# #.#..#.#.# #.#.##.#.# #.#......# ###.###### 2 10 U 2 10 L 4 8 U 10 4 U 8 7 U 8 9 U 6 2 U 5 2 U 3 5 D 7 4 R	1 11 7 61 -1 54 11 -1 33 4