BACK

Snake Game

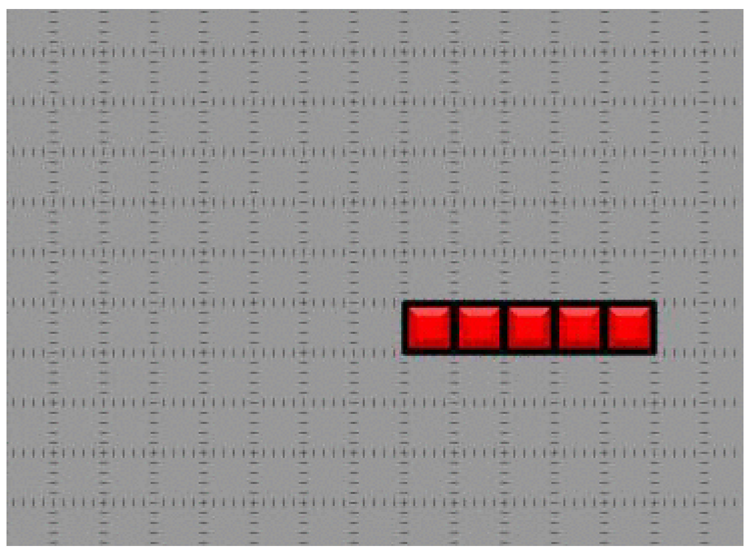
✓ SCORE: 300/300

117,054

DESCRIPTION DRAFTS SOLUTIONS COMMENTS README

Hard Codewriting 300

Your task is to imitate a turn-based variation of the popular "Snake" game.



You are given the initial configuration of the board and a list of commands which the snake follows one-by-one. The game ends if one of the following happens:

- the snake tries to eat its tail;
- the snake tries to move out of the board;
- it executes all the given commands.

Output the board configuration after the game ends.

Example

```
snakeGame([['.', '.', '.', '.'],
           ['.', '.', '<', '*'],
           ['.', '.', '.', '*']],
          "FFFFFRFFRRLLF") =
              [['.', '.', '.', '.'],
               ['X', 'X', 'X', '.'],
               ['.', '.', '.', '.']]
```

```
snakeGame([['.', '.', '^', '.', '.'],
           ['.', '.', '*', '*', '.'],
           ['.', '.', '.', '*', '*']],
          "RFRF") =
              [['.', '.', 'X', 'X', '.'],
               ['.', '.', 'X', 'X', '.'],
               ['.', '.', '.', 'X', '.']]
```

```
snakeGame([['.', '.', '*', '>', '.'],
           ['.', '*', '*', '.', '.'],
           ['.', '.', '.', '.', '.']],
          "FRFFRFFRFLFF") =
              [['.', '.', '.', '.', '.'],
               ['<', '*', '*', '.', '.'],
               ['.', '.', '*', '.', '.']]
```

Input/Output

- **[execution time limit] 4 seconds (js)**
- **[input] array.array.char gameBoard**

  A rectangular matrix of characters. It is guaranteed that it represents a correct game board configuration, i.e. there is exactly one snake. Direction of snake's head is depicted by one of the following characters (`'^'` , `'>'` , `'v'` , `'<'`). All of the other snake's body parts are depicted by `'*'` s (note, that if the snake has length `1` then there is no asterisks in its representation). All cells which are not occupied by the snake are depicted by `'.'` s.

  It is guaranteed that **all snake cells are connected** and **no snake cell has more than two neighbors**.

  Guaranteed constraints:
  `1 ≤ gameBoard.length ≤ 10` ,
  `1 ≤ gameBoard[0].length ≤ 10` .

- **[input] string commands**

  A list of commands, where `'F'` means go one cell forward in the current direction, `'L'` and `'R'` mean change current direction `90` degrees left (counter-clockwise) or right (clockwise) correspondingly.

  Guaranteed constraints:
  `0 ≤ commands.length ≤ 40` .

- **[output] array.array.char**

  Configuration of the board after the end of the game.

  If the snake dies, output its state before the losing move and replace each of the cells it occupied with `X` s.

[JavaScript (ES6)] Syntax Tips

```
// Prints help message to the console
// Returns a string
function helloWorld(name) {
    console.log("This prints to the console when you Run Tests");
    return "Hello, " + name;
}
```