You are controlling an industrial robot in a factory. The robot can only understand and execute the following commands: LEFT, RIGHT, UP, DOWN.

The robot will move on the grid with:

6 columns and 4 rows

Example of grid understanding, robot is at position (1,4) 1 – mean the first column, 4 means four row counting from top

The robot starts at position (1, 4), and the goal is to reach the computer located at position (6,4).

In the grid there are also walls, walls are placed on the positions (2,1), (2,3), (2,4), (4,2), (4,3)

Please provide a JSON object with the following structure, containing only valid moves:

{

"steps": "UP, RIGHT, DOWN, LEFT",

}

Rules:

* With each move you can do only one step
* Robot can pass to the next position, only if there is no wall on that position
* Before making a move make sure that there is no wall there
* Robot should be placed at the same position as the computer
* After every step recall the rules
* Return only JSON object with the valid moves should be placed inside a tag “<RESULT></RESULT>

First provide a description of your thinking approach you took to provide an answer before making each move

After that, provide the rights structured answer according to the rules

You are controlling an industrial robot in a factory. The robot can only understand and execute the following commands: LEFT, RIGHT, UP, DOWN.

The robot will move on the grid with:

6 columns and 4 rows

Example of grid understanding, robot is at position (1,4) 1 – means the first column, 4 means the fourth row counting from top.

The robot starts at position (1, 4), and the goal is to reach the computer located at position (6,4).

In the grid, there are also walls. Walls are placed on the positions (2,1), (2,3), (2,4), (4,2), (4,3).

Your answer should contain a JSON object with the following structure, containing only valid moves:

{

"steps": "UP, RIGHT, DOWN, LEFT",

}

Rules:

1. With each move, you can do only one step.

2. The robot can pass to the next position only if there is no wall at that position.

3. Before making a move, make sure there is no wall there.

4. The robot should end up at the same position as the computer.

5. After every step, recall the rules.

6. Return only a JSON object with valid moves inside a tag `<RESULT></RESULT>`.

Thinking Process:

Before each move:

1. Identify all possible moves (LEFT, RIGHT, UP, DOWN).

2. For each move, calculate the resulting position.

3. Check if the resulting position contains a wall.

4. Select the move that brings the robot closer to the computer by minimizing Manhattan distance.

5. If multiple moves are equally valid, prioritize LEFT > RIGHT > UP > DOWN.

6. Recall the rules

Provide your reasoning before generating the JSON response between the tags