```python
def display_Path_to_Princess(N, grid):
    # Find the bot's position (always in the center)
    bot_position = (N // 2, N // 2)

    # Find the princess's position (in one of the four corners)
    if grid[0][0] == 'p':
        princess_position = (0, 0)
    elif grid[0][N-1] == 'p':
        princess_position = (0, N-1)
    elif grid[N-1][0] == 'p':
        princess_position = (N-1, 0)
    elif grid[N-1][N-1] == 'p':
        princess_position = (N-1, N-1)

    # Calculate the differences in row and column indices
    row_diff = princess_position[0] - bot_position[0]
    col_diff = princess_position[1] - bot_position[1]

    # Output the moves to reach the princess
    moves = []

    if row_diff > 0:
        moves.extend(["DOWN"] * row_diff)
    elif row_diff < 0:
        moves.extend(["UP"] * abs(row_diff))

    if col_diff > 0:
        moves.extend(["RIGHT"] * col_diff)
    elif col_diff < 0:
        moves.extend(["LEFT"] * abs(col_diff))

    # Print all moves, each on a new line
    for move in moves:
        print(move)

```

STDIN

Input for the program (Optio

Output:

```
DOWN
LEFT
```

```
35
36 # Example usage:
37 N = 3
38 grid = [
39     "---",
40     "-m-",
41     "p--"
42 ]
43
44 display_Path_to_Princess(N, grid)
45
```