

Princess Peach is trapped in one of the four corners of a square grid. You are in the center of the grid and can move one step at a time in any of the four directions. Can you rescue the princess?

Input format

The first line contains an odd integer N ($3 \leq N < 100$) denoting the size of the grid. This is followed by an $N \times N$ grid. Each cell is denoted by '-' (ascii value: 45). The bot position is denoted by 'm' and the princess position is denoted by 'p'.

Grid is indexed using [Matrix Convention](#)

Output format

Print out the moves you will take to rescue the princess in one go. The moves must be separated by '\n', a newline. The valid moves are LEFT or RIGHT or UP or DOWN.

Sample input

```
3
---
-m-
p--
```

Sample output

```
DOWN
LEFT
```

Task

Complete the function `displayPathtoPrincess` which takes in two parameters - the integer `N` and the character array `grid`. The grid will be formatted exactly as you see it in the input, so for the sample input the princess is at `grid[2][0]`. The function shall output moves (LEFT, RIGHT, UP or DOWN) on consecutive lines to rescue/reach the princess. The goal is to reach the princess in as few moves as possible.

The above sample input is just to help you understand the format. The princess ('p') can be in any one of the four corners.

Scoring

Your score is calculated as follows : $(N \times N - \text{number of moves made to rescue the princess}) / 10$, where `N` is the size of the grid (3x3 in the sample testcase).

```
1  #!/usr/bin/python
2
3  def displayPathtoPrincess(n,grid):
4      #print all the moves here
5
6      m = int(input())
7      grid = []
8      for i in range(0, m):
9          grid.append(input().strip())
10
11     displayPathtoPrincess(m,grid)
```