Part 1: Navigation

As those of you in the online section learned in Module 1, a certain autonomous agent likes to fly around the house and interrupt video recordings at the most inopportune moments (Figure 1). Suppose that a house consists of a grid of $N \times M$ cells, represented like this:

```
....XXX
.XXX...
....X..
.X.X...
.X.X.X.
pX...X@
```

As you can see, the map consists of N lines (in this case, 6) and M columns (in this case, 7). Each cell of the house is marked with one of four symbols: `p` represents the agent's current location, `X` represents a wall through which the agent cannot pass, `.` represents open space over which the agent can fly, and `@` represents your location (presumably with video recording in progress).

Figure 1: The autonomous agent, after a bath.

Your goal is to write a program that finds the shortest path between the agent and you. The agent can move one square at a time in any of the four principal compass directions, and the program should find the shortest distance between the two points and then output a string of letters (L, R, D, and U for left, right, down, and up) indicating that solution. Your program should take a single command line argument, which is the name of the file containing the map file. For example:

```
[<>djcran@silo ~] python3 route_pichu.sh map1.txt
Shhhh... quiet while I navigate!
Here's the solution I found:
16 UUURRDDDRRUURRDD
```

You can assume that there is always exactly one `p` and one `@` in the map file. If there is no solution, your program should display path length `-1` and not display a path.