Path Navigation using BFS Traversal

Assume you are trapped in a mystical castle which you want to escape and have a map of the castle. The castle consists of NxM grid, represented as given below:

```
....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 castle is marked with one of four symbols: p represents your current location, X represents a wall through which you cannot pass, . represents open space over which you can walk, and @ represents the location through which a magic portal opens, from which you can escape.

Your goal is to write a program to find the shortest path between your location and the opening of the magic portal. You can move one square at a time in any of the four principal compass directions, and the program should output a number indicating the shortest distance between the two points, followed by 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 mystical_castle.py
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.