

Problem E

Emergency Exit

Patrick wants to organize a big event in the computer rooms at KTH. He has set up posters and sent out invitations, and it looks like quite a large crowd of people will show up. Now Patrick has to deal with an unexpected problem: fire safety regulations. People must be able to get out fast enough in case of an emergency, so only a certain number of people are allowed in the computer rooms, and it looks like



Patrick's event will exceed this number. But Patrick has not given up, he wants to prove that everyone can still get out fast enough if they behave optimally.

The building can be represented by an $N \times M$ grid where some cells are blocked and others are empty. There will be *exactly one* door to the building. This means that one cell on the edge of the grid is empty and the rest are blocked. Some empty cells have people in them. In one time step, a person can move to an empty cell that shares a side with the cell they are currently in. It is also possible to not move in a time step. Everyone moves simultaneously, and no two people may occupy the same cell at the end of a time step. When someone exits the grid (through the one door), they disappear and can no longer collide with other people.

Your task is to calculate the minimum time to get everybody out, and instruct the people to move in such a way to make this happen.

Input

The first line contains two integers N and M ($3 \leq N, M \leq 100$), the number of rows and columns in the grid.

The following N lines each contains a string of length M , representing each row of the grid. An empty cell without a person is denoted by “.”, a person is denoted by “P”, and a blocked cell is denoted by “#”.

Every cell on the edge of the grid will be blocked except for exactly one. The number of people in the grid will be between 1 and 100.

Output

If it is impossible for everyone to get out (i.e. someone is stuck inside the building), print one line with -1 .

Otherwise, print one line with the integer S , the number of seconds it takes to get everyone out. Let K be the number of people in the grid. Print K lines each containing one string of length S consisting of the characters “U” (up), “D” (down), “R” (right), “L” (left), “.” (wait). This string represents the actions the i -th person should take. The people are considered in the order they appear in the input, if the input grid is read row by row from left to right. When a person has exited the grid, give them wait instructions for the remaining time.

Sample Input 1

```

4 6
#####
#.P..#
#P.#P#
##P###

```

Sample Output 1

```

6
DDD...
.RDD..
ULLDDD
D.....

```

Sample Input 2

```

3 3
##.
#P#
###

```

Sample Output 2

```

-1

```

Sample Input 3

```

7 27
#####
#.....
#####.#####.#####.##
#.....#.....#.....#..#
#...#..#..P#...P#..P#..#..#
#...#..#P.P#P..P#...#.P#..#
#####

```

Sample Output 3

```

25
URUURRRRRRRRRRRRRRRR.....
LLLULUURRRRRRRRRRRRRR..
LLUUURRRRRRRRRR.....
.RRUURUURRRRRRRRRRRRRR.
UURUURRRRRRRRRRRRRRRR...
.UULUURRRRRRRRRRRRRRRR...
.LLLUULUURRRRRRRRRRRRRR
LUULLLLUURRRRRRRRRR.....

```