

5774 Robot Navigation

A robot has been sent to explore a remote planet. To specify a path the robot should take, a program is sent each day. The program consists of a sequence of the following commands:

- FORWARD X : move forward by X units.
- TURN LEFT: turn left (in place) by 90 degrees.
- TURN RIGHT: turn right (in place) by 90 degrees.

The robot also has sensor units which allow it to obtain a map of its surrounding area. The map is represented as a grid. Some grid points contain hazards (e.g. craters) and the program must avoid these points or risk losing the robot.

Naturally, if the initial location of the robot, the direction it is facing, and its destination position are known, it is best to send the shortest program (one consisting of the fewest commands) to move the robot to its destination (we do not care which direction it faces at the destination). You are more interested in knowing the number of different shortest programs that can move the robot to its destination. However, the number of shortest programs can be very large, so you are satisfied to compute the number as a remainder modulo 1,000,000.

Input

There will be several test cases in the input. Each test case will begin with a line with two integers N M

Where N is the number of rows in the grid, and M is the number of columns in the grid ($2 \leq N, M \leq 100$). The next N lines of input will have M characters each. The characters will be one of the following:

- '.' Indicating a navigable grid point.
- '*' Indicating a crater (i.e. a non-navigable grid point).
- 'X' Indicating the target grid point. There will be exactly one 'X'.
- 'N', 'E', 'S', or 'W' Indicating the starting point and initial heading of the robot. There will be exactly one of these. Note that the directions mirror compass directions on a map: 'N' is North (toward the top of the grid), 'E' is East (toward the right of the grid), 'S' is South (toward the bottom of the grid) and 'W' is West (toward the left of the grid).

There will be no spaces and no other characters in the description of the map. The input will end with a line with two 0's.

Output

For each test case, output two integers on a single line, with a single space between them. The first is the length of a shortest possible program to navigate the robot from its starting point to the target, and the second is the number of different programs of that length which will get the robot to the target (modulo 1,000,000). If there is no path from the robot to the target, output two zeros separated by a single space. Output no extra spaces, and do not separate answers with blank lines.

Sample Input

```
5 6
*...X
.....*
.....*
.....*
N...*
6 5
...X
.****
.****
.****
.****
N****
3 3
.E.
***
.X.
0 0
```

Sample Output

```
6 4
3 1
0 0
```