Submarine

October 2021
C++ — 2 sec — 512 MB

May Day! May Day! Radio Silence. A faint crackle. *This is HMS Sinky, anyone receiving? Repeat, anyone receiving?*

HMS Sinky, a nuclear submarine, has drifted off course into a field of sea-mines. One wrong turn will end in disaster. Everything is critical.

The crew aboard the submarine have managed to send you a 3D scan of the minefield and a set of directions. The minefield is represented by an **n** x **n** x **n** matrix of cells, which either contain a mine (#) or are empty (.) Below is a scan of a minefield with width 4. Each square shows a layer of the cube, starting at the top on the left.

```
#..#    ....    ####    #.#.
....    .#..    ....    ...#
....    ..#.    ....    ...#
#..#    ....    ####    #...
```

The directions you have received are the unit directions the submarine has moved in, either North (N), South (S), East (E), West (W), Up (U), or Down (D). However, due to interference, some directions could not be decoded properly (?), and at these moments the submarine could have moved in any of the 6 directions.

You do not know where the submarine started, or where it is now. Given that the submarine always stays in empty cells in the scan, you can determine a list of the submarine's current possible locations.

INPUT You will be given integers **n** and **s**, denoting the width of the scan, and the length of the set of directions respectively. The next **n** x **n** lines form a 3D grid of cells (either # or .) which represent the scan of the minefield. Each layer is given, one after the other, starting at the top. The last line of input will be a length **s** string, giving the directions for the submarine.

$4 \leq \mathbf{n} \leq 80$
$3 \leq \mathbf{s} \leq 1000$

OUTPUT Output a single integer, the number of possible ending locations for the submarine.

For example, suppose the submarine was somewhere in the minefield shown above, and moved in the directions: EUNWDS. The submarine could have ended in the second layer at (0, 2). Its movement is shown below, also starting in the second layer at (0, 2). The

numbers represent the position of the submarine after moving in the **m**th direction. Note that in this case, the submarine starts and ends at the same position.

```
#..#     ....     ####     #.#.
43..     5#..     ....     ...#
.2..     01#.     ....     ...#
#..#     ....     ####     #...
```

If the directions were `EU?WDS` there would be 9 different positions the submarine could end at, shown below.

```
#..#     ....     ####     #.#.
....     .#..     ....     ...#
....     X.#.     ..X.     XXX#
#..#     XX..     ####     #XX.
```

SAMPLES

INPUT

```
4 6
#..#
....
....
#..#
....
.#..
..#.
....
####
....
....
####
#.#.
...#
...#
#...
EU?WDS
```

OUTPUT

```
9
```