G Gluttonous Goop

Time limit: 3s

As a prominent researcher in the laboratorium for Breathtaking Agriculture through Petri dish Cultivation, you keep on looking for new organisms that might be the food of the future. Recently you have discovered a new fungus-type organism that seems to be nutritious and very efficient in converting energy from food to body mass. You have placed a small batch surrounded by food in a petri dish and watched it grow for a bit.

However, now that the weekend has arrived, you would rather spend some time with your loved ones than stare at the contents of this petri dish all the time (even though it is a *fun guy*). You cannot leave without taking the necessary precautions. What if the fungus grows too large and starts eating away at the rest of the laboratory?!

You model the situation as follows: you divide the plane into 1×1-squares, and draw where the fungus currently is. You know that every time step, if the fungus occupies a square, it will expand to all eight neighbouring squares (and still occupy the initial square). You know how many time steps you will be gone for over the weekend, and now you want to know how many squares the fungus will occupy when you get back.

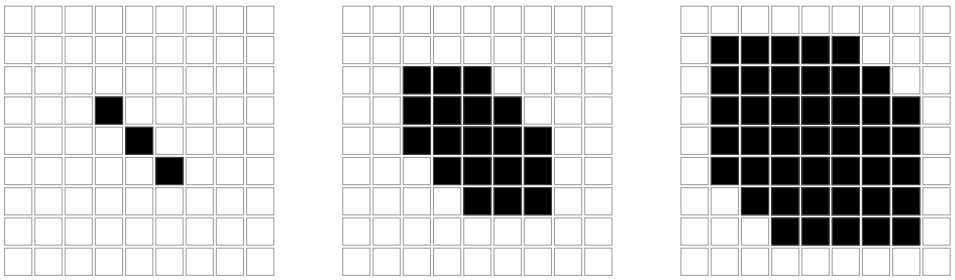


Figure G.1: Example of fungus growth: the fungus from sample 2 after 0, 1, 2 time steps. The middle image corresponds to the correct output for sample 2.

N.B.: In the input, the fungus will be given on a finite grid, but it can (and will!) grow beyond these boundaries. The fungus is not so easily contained.

Input

- First a line containing integers $1 \leq r, c \leq 20$, and $0 \leq k \leq 10^6$, denoting the number of rows and columns of the initial grid and the number of time steps.
- Then follow r lines of c characters, each character being either '`.`' or '`#`'. A '`#`' denotes that the fungus is occupying this square. The fungus need not be connected.

Output

- Output the number of squares the fungus occupies after k time steps have passed.

Sample Input 1

```
5 5 3
.....
.###.
.#.#.
.###.
.....
```

Sample Output 1

```
81
```

Sample Input 2

```
3 3 1
#..
.#.
..#
```

Sample Output 2

```
19
```

Sample Input 3

```
4 6 3
..##..
.#..#.
.#..#.
..##..
```

Sample Output 3

```
96
```

Sample Input 4

```
1 1 1000000
#
```

Sample Output 4

```
4000004000001
```