# The Bomberman Game                                    🔒 locked

👤 by **kevinsogo**

| Problem | Submissions | Leaderboard | Discussions |
| --- | --- | --- | --- |

**Bomberman** lives in a rectangular grid with $R$ rows and $C$ columns. Each cell in the grid either contains a bomb or nothing at all.

Each bomb can be planted in any cell of the grid but, once planted, it will detonate after *exactly 3 seconds*. Once a bomb detonates, it's destroyed — along with anything in its four neighboring cells. This means that if a bomb detonates in cell $i, j$, cells $(i \pm 1, j)$ and $(i, j \pm 1)$ are cleared (note that cells at the edge of the grid have less than four neighboring cells). If there is a bomb in a neighboring cell, the neighboring bomb is destroyed *without* detonating (i.e., no chain reaction occurs).

Bomberman is immune to bombs, so he can move freely throughout the grid. Here's what he does:

1. Initially, Bomberman arbitrarily plants bombs in some of the cells.

2. After one second, Bomberman does nothing.

3. After one more second, Bomberman plants bombs in all cells without bombs, thus filling the whole grid with bombs. It is guaranteed that no bombs will detonate at this second.

4. After one more second, any bombs planted exactly three seconds ago will detonate. Here, Bomberman stands back and observes.

5. Bomberman then repeats steps 3 and 4 indefinitely.

Note that during every second Bomberman plants bombs, the bombs are planted simultaneously (i.e., *at the exact same moment*), and any bombs planted at the same time will detonate at the same time.

Given the initial configuration of the grid with the locations of Bomberman's first batch of planted bombs, determine the state of the grid after $N$ seconds.

### Input Format

The first line contains three space-separated integers denoting the respective values of $R$, $C$, and $N$.
Each line $i$ of the $R$ subsequent lines describes row $i$ of the grid's initial state as a single string of $C$ characters; the `.` character denotes an empty cell, and the `O` character (ascii 79) denotes a bomb.

### Constraints

- $1 \le R, C \le 200$

- $1 \le N \le 10^9$

### Subtask

- $1 \le N \le 200$ for $40\%$ of the maximum score.

### Output Format

Print the grid's final state. This means $R$ lines where each line contains $C$ characters, and each character is either a `.` or an `O` (ascii 79). This grid must represent the state of the grid after $N$ seconds.

### Sample Input

```
6 7 3
.......
...O...
....O..
.......
```

```
OO.....
OO.....
```

**Sample Output**

```
OOO.OOO
OO...OO
OOO...O
..OO.OO
...OOOO
...OOOO
```

**Explanation**

The initial state of the grid is:

```
.......
...O...
....O..
.......
OO.....
OO.....
```

Bomberman spends the first second doing nothing, so this is the state after 1 second:

```
.......
...O...
....O..
.......
OO.....
OO.....
```

Bomberman plants bombs in all the empty cells during his second second, so this is the state after 2 seconds:

```
OOOOOOO
OOOOOOO
OOOOOOO
OOOOOOO
OOOOOOO
OOOOOOO
```

In his third second, Bomberman sits back and watches all the bombs he planted 3 seconds ago detonate. This is the final state after **3** seconds:

```
OOO.OOO
OO...OO
OOO...O
..OO.OO
...OOOO
...OOOO
```

f   y   in

**Submissions:** 5
**Max Score:** 40
**Difficulty:** Medium

**Rate This Challenge:**
☆ ☆ ☆ ☆ ☆

More

**Current Buffer** (saved locally, editable)

Python 3

1

Upload Code as File ☐ Test against custom input

Run Code Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature