Death Star's Demise

In another Galaxy Far Far away, Luke Skywalker is on his trusted X-Wing in a dogfight of the century with the Empire. Their prime target : The Death Star. But The Dark Lord Darth vader learnt from his last mistake. The Energy source has been spread out over a rectangular grid having square shaped rooms.

With Luke as your eyes in the sky, you must commandeer a Y-Wing carrying Detonation Charges. A charge can be dropped in any room of the grid, but explodes 3 seconds after landing. Once a charge explodes, all the room attached with it in all 4 directions are destroyed.

This means that if a charge detonates in cell i, j, any valid cells (i +- 1, j) and (i, j +- 1) are cleared and no other rooms are destroyed.

Your Y-Wing is being defended by Luke and the rest of the resistance, so that you can move freely above the grid. These are the instructions given to you by Leia and Luke :-

(i). Plant charges arbitrarily in some of the rooms, the initial state.

(ii). After one second, do nothing.

(iii). After one more second, plant charges in empty rooms. This fills the entire grid with charges. None of the charges explode at this point.

(iv). After one more second, any charges set exactly three seconds ago will explode. Here, you have to stand and confirm detonation.

(v). Repeat steps (iii) and (iv) indefinitely.

Any charge planted at the same time will expode at the same time. You have to determine the state of the grid after t seconds

Input Format

- The first line inputs three space-separated integers a(number of rows), b(number of columns), and t(seconds to stimulate).
- Each of the next a lines contains a row of the matrix as a single string of b characters.
- The "." character represents an empty room, and the "O" character (ascii 79) represents a charge.

Constraints

$$1 \le a, b \le 200$$
$$1 \le t \le 10^{09}$$

Output Format

Print the final state of grid. This means a, lines where each line contains b, characters, and each character is either a "." or an "O". This grid gives the state of the grid after t seconds.

Sample Input 0

```
6 7 5
.......
...O.O.
....O..
..O....
OO...OO
OO.O...
```

Sample Output 0

```
.......
...O.O.
...OO..
..OOOO.
OOOOOOO

OOOOOOO
```

Expected Solution :

```
import java.util.Scanner;

public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        int R = in.nextInt();
        int C = in.nextInt();
        int N = in.nextInt();

        char[][] first = new char[R + 2][C + 2];

        for (int i = 1; i <= R; i++) {
            String row = in.next();

            for (int j = 1; j <= C; j++)
                first[i][j] = row.charAt(j - 1);
        }

        if (N == 1) {
            for (int i = 1; i <= R; i++) {
                for (int j = 1; j <= C; j++)
                    System.out.print(first[i][j]);
                System.out.println();
            }
            return;
        } else if (N % 2 == 0) {
            for (int i = 1; i <= R; i++) {
                for (int j = 1; j <= C; j++)
                    System.out.print('O');
                System.out.println();
            }
            return;
        }

        int[][] third = new int[R + 2][C + 2];
        for (int i = 1; i <= R; ++i) {
            for (int j = 1; j <= C; ++j)
                if (first[i][j] == 'O')
                    third[i][j] = third[i + 1][j] = third[i][j + 1] = third[i - 1][j] = third[i][j - 1] = 1;
        }
        if (N % 4 == 3) {   // 3 7 11..
            for (int i = 1; i <= R; i++) {
                for (int j = 1; j <= C; j++) {
                    if (third[i][j] == 1)
                        System.out.print('.');
                    else
                        System.out.print('O');
                }
                System.out.println();
            }
            return;
        }

        int[][] fifth = new int[R + 2][C + 2];
        for (int i = 1; i <= R; i++) {
            for (int j = 1; j <= C; j++)
                if (third[i][j] == 0)
                    fifth[i][j] = fifth[i + 1][j] = fifth[i][j + 1] = fifth[i - 1][j] = fifth[i][j - 1] = 1;
        }

        if (N % 4 == 1) {   // 5 9 13..
            for (int i = 1; i <= R; i++) {
                for (int j = 1; j <= C; j++) {
                    if (fifth[i][j] == 1)
                        System.out.print('.');
                    else
                        System.out.print('O');
                }
                System.out.println();
            }
        }
    }
}
```