



The Eccentric Vacuum Cleaner

Problem Statement

Oski recently bought an automatic vacuum cleaner to clean his room, represented as an $N \times M$ grid. Rows are numbered 1 through N from top to bottom, and columns are numbered 1 through M from left to right. Some cells are empty (.), while others contain obstacles (#).

However, Oski noticed that the robot's navigation unit is completely faulty and follows a very weird movement protocol. Curious about its behavior, Oski places the robot at an empty cell (r_0, c_0) facing **Right** (East) and lets it run for K steps.

At each step, the robot operates according to the following rules:

1. It inspects the cell directly ahead in its current facing direction.
2. **Successful Move:** If the target cell is within the grid boundaries and does not contain an obstacle (#), the robot moves into that cell and immediately rotates **90 degrees counter-clockwise (CCW)**.
3. **Blocked Move:** If the target cell is outside the grid boundaries or contains an obstacle (#), the robot stays in its current cell and immediately rotates **90 degrees clockwise (CW)**.

Oski wants to know two things after the robot executes exactly K steps:

1. The number of distinct empty cells visited at least once (including the starting position).
2. The final position (r_{last}, c_{last}) of the robot.

Help Oski compute these values!

Input Format

- The first line contains three integers N , M , and K ($1 \leq N, M \leq 1000$; $1 \leq K \leq 10^{18}$).
- The second line contains two integers r_0 and c_0 ($1 \leq r_0 \leq N$; $1 \leq c_0 \leq M$), representing the starting coordinates of the robot.
- The next N lines each contain a string of M characters (. or #), representing the grid. It is guaranteed that cell (r_0, c_0) is . .

Output Format

- Print the first line containing a single integer — the number of distinct cells visited by the robot.
- Print the second line containing two space-separated integers — the final row r_{last} and column c_{last} of the robot after K steps.

Sample Input 1

```
3 3 5
1 1
...
.#.
...
```

Sample Output 1

```
3
1 3
```

Explanation of Sample 1

Orientations: 0 = Right, 1 = Up, 2 = Left, 3 = Down

- **Start:** (1, 1), facing Right (0).
- **Step 1:** Moves to (1, 2), turns CCW → facing Up (1).
- **Step 2:** Blocked by top boundary! Stays at (1, 2), turns CW → facing Right (0).
- **Step 3:** Moves to (1, 3), turns CCW → facing Up (1).
- **Step 4:** Blocked by top boundary! Stays at (1, 3), turns CW → facing Right (0).
- **Step 5:** Blocked by right boundary! Stays at (1, 3), turns CW → facing Down (3).

Visited cells: (1, 1), (1, 2), (1, 3) (Total: 3).

Final position: (1, 3).

Sample Input 2

```
4 4 1000
1 1
....
.##.
.##.
....
```

Sample Output 2

```
12
4 1
```