Snake Move

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	1024 mebibytes

Putata is playing a famous snake game on his laptop, where a snake moves around on a grid of size $n \times m$. There may be obstacles in some cells of the grid. The snake can be represented as a sequence of coordinate pairs that determine where its body is located: $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$. Here, k denotes the length of the snake. The head of the snake is at (x_1, y_1), the tail is at (x_k, y_k), and neighboring parts of the body are located in cells that share a side.

In this game, the snake is programmed with a series of commands represented as a string. There are 5 types of commands that you can use:

- '`L`': Command the snake to move one step left. The head will then move to $(x_1, y_1 - 1)$.
- '`R`': Command the snake to move one step right. The head will then move to $(x_1, y_1 + 1)$.
- '`U`': Command the snake to move one step up. The head will then move to $(x_1 - 1, y_1)$.
- '`D`': Command the snake to move one step down. The head will then move to $(x_1 + 1, y_1)$.
- '`S`': Shorten the length of the snake by one. The tail of the body will be erased. The length will become $k - 1$. Note that you can not do this when $k = 1$.

When the head moves, each part of the body also moves accordingly. Specifically, the i-th part of the body ($2 \le i \le k$) moves to the position where the $(i-1)$-st part was before the command. The snake can not move into a cell with an obstacle, and can not move outside the grid. Besides, the snake cannot collide with itself. So you should guarantee that no two parts of the body will share the same location.

Consider the following corner case: The head is at (x_1, y_1), and the tail is at (x_k, y_k). If the head is moving to (x'_1, y'_1), then it is **allowed** to move to $(x'_1, y'_1) = (x_k, y_k)$: if we think about a real-world scenario, the head moves into the cell just as the tail moves out. In a similar fashion, it is **allowed** to swap the head and the tail by using a single command when $k = 2$.

You will be given the map of the grid and the body sequence of the snake. Let $f(i, j)$ denote the minimum number of commands that Putata needs to use such that the head of the snake will arrive at (i, j), or 0 if it is impossible. You have to calculate:

$$\left(\sum_{i=1}^{n} \sum_{j=1}^{m} f(i,j)^2 \right) \bmod 2^{64}.$$

Input

The first line of the input contains three integers n, m and k ($1 \le n, m \le 3000$, $1 \le k \le \min\{nm, 10^5\}$) denoting the size of the grid and the length of the snake.

In the next k lines, the i-th line contains two integers x_i and y_i ($1 \le x_i \le n$, $1 \le y_i \le m$, $|x_i - x_{i+1}| + |y_i - y_{i+1}| = 1$) denoting the location of the i-th part of the body. It is guaranteed that all the k pairs (x_i, y_i) are pairwise distinct. It is also guaranteed that each part is in a cell without an obstacle.

In the next n lines, the i-th line contains a string of length m. If cell (i, j) is empty, the j-th character in the i-th of these lines is '`.`'. If cell (i, j) is occupied by an obstacle, the character is '`#`'.

Output

Print a single line containing an integer: the answer to the problem.

Examples

standard input	*standard output*
4 5 5 3 5 3 4 3 3 3 2 4 2	293
2 2 4 1 1 1 2 2 2 2 1	14
5 5 3 1 2 1 1 2 1 ..... .###. .#.#. .###.	407