You have a rectangular board consisting of n rows, numbered from 1 to n, and m columns, numbered from 1 to m. The top left is $(1,1)$ and the bottom right is (n,m). Initially - at time 0 - there is a coin on the top-left cell of your board. Each cell of your board contains one of these letters:

- *: Exactly one of your cells has letter '*'.
- **U**: If at time t the coin is on cell (i,j) and cell (i,j) has letter 'U', the coin will be on cell $(i-1,j)$ at time $t+1$, if $i > 1$. Otherwise, there is no coin on your board at time $t+1$.
- **L**: If at time t the coin is on cell (i,j) and cell (i,j) has letter 'L', the coin will be on cell $(i,j-1)$ at time $t+1$, if $j > 1$. Otherwise, there is no coin on your board at time $t+1$.
- **D**: If at time t the coin is on cell (i,j) and cell (i,j) has letter 'D', the coin will be on cell $(i+1,j)$ at time $t+1$, if $i < N$. Otherwise, there is no coin on your board at time $t+1$.
- **R**: If at time t the coin is on cell (i,j) and cell (i,j) has letter 'R', the coin will be on cell $(i,j+1)$ at time $t+1$, if $j < M$. Otherwise, there is no coin on your board at time $t+1$.

When the coin reaches a cell that has letter '*', it will stay there permanently. When you punch on your board, your timer starts and the coin moves between cells. Before starting the game, you can make operations to change the board, such that you are sure that at or before time k the coin will reach the cell having letter '*'. In each operation you can select a cell with some letter other than '*' and change the letter to '*U*', '*L*', '*R*' or '*D*'. You need to carry out as few operations as possible in order to achieve your goal. Your task is to find the minimum number of operations.

For example, given a grid of $n = 2$ rows and $m = 3$ columns:

```
UDL
RR*
```

the goal is to get from $(1,1)$ to $(2,3)$ in as few steps as possible. As the grid stands, it cannot be done because of the `U` in the cell at $(1,1)$. If $(1,1)$ is changed to `D`, the path $(1,1) \to (2,1) \to (2,2) \to (2,3)$ is available. It could also be changed to `R` which would make the path $(1,1) \to (1,2) \to (2,2) \to (2,3)$ available. Either choice takes 1 change operation, which is the value sought if $k \geq 3$. A lower value of k would result in a return value of -1 because the shortest path is 3 steps, starting from $(1,1)$.

Function Description

Complete the *coinOnTheTable* function in the editor below. It should return an integer that represents the minimum operations to achieve the goal, or -1 if it is not possible.

coinOnTheTable has the following parameters:

- *m*: an integer, the number of columns on the board
- *k*: an integer, the maximum time to reach the goal
- *board*: an array of strings where each string represents a row of the board

Input Format

The first line of input contains three integers, n, m, and k, the number of rows, the number of columns and the maximum time respectively.

The next n lines contain m letters each, describing your board.

Constraints

$n, m \leq 51$
$k \leq 1000$

Output Format

Print an integer which represents the minimum number of operations required to achieve your goal. If you cannot achieve your goal, print -1.

Sample Input

```
2 2 3
RD
*L
```

Sample output :

```
0
```

Sample input :

```
2 2 1
RD
*L
```

Sample output :

```
1
```

Explanation :

In the first example, a valid path exists without making any changes. In the second example, the letter of cell (1,1) must be changed to 'D' to make a valid path. In each example, a path length $\leq \boldsymbol{k}$ is available.