Problem A: Appearance Analysis

Time limit: 1 s
Memory limit: 512 MiB

You have taken a photo of the Faculty of Electrical Engineering and Computing "C" building here in Zagreb with its equal sized windows neatly arranged in rows and columns. Some of the windows are painted in curious designs and you are trying to analyze the photo and determine the number of distinct designs.

We represent the photo as a rectangular grid of characters with r rows and c columns. Every window is a rectangular area and all windows have the same dimensions. Each cell in a window is either clear (denoted by the "." character) or painted (denoted by the "+" character). Two windows are considered to be of the *same design* if one can be rotated a multiple of 90 degrees and placed on top of the other so that they perfectly match. When comparing designs, we are not allowed to flip a window inside out.

Windows are regularly aligned in rows and columns with exactly one row of brick cells (denoted by the "#" character) framing each window. More precisely, there is a single row of "#" characters between two consecutive window rows as well as before the first window row and after the last window row. Similarly, there is a single column of "#" characters between two consecutive window columns as well as before the first window column and after the last window column. The exact number of window rows and window columns is arbitrary. The window dimensions are also arbitrary. However, a window consists of at least one cell and, again, all windows in the photo have the same dimensions.

Find the number of different window designs in the photo.

Input

The first line contains two integers r and c ($3 \leq r, c \leq 111$) — number of rows and the number of columns of the photo. Each of the following r lines contains a string consisting of c characters — one row of the photo.

Output

Output a single integer — the number of different window designs in the photo.

Example

input

```
11 16
################
#....#++++#+...#
#....#++.+#+...#
#....#.++.#++.+#
#....#....#++++#
################
#....#.+..#++++#
#..++#.+..#++.+#
#+...#....#.++.#
#+...#..++#....#
################
```

output

```
4
```

input

```
9 21
####################
#...+#++++#+...#..+.#
#..+.#.++.#.+..#..+.#
#.+..#....#..+.#..+.#
####################
#.+..#....#..+.#.+..#
#..+.#.++.#.+..#.+..#
#...+#++++#+...#.+..#
####################
```

output

```
4
```