



COMPETITIVE

PROG-A-THON

8: Window designs

Time limit: 1.000 seconds

We will get a new building at Science Park! Of course, the building has yet to be built and they are now busy with designing the building. The design of the outside has just been finished. When you look at the design, you notice that they have a lot of different kind of windows. Although all windows have the same size, some are painted in a different way. You are wondering how many distinct kinds of windows there are in the design you are given.

The design is given to you 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 windows is either clear (denoted by the '.' character) or painted (denoted by the '+' character). Two windows are considered to be of the same kind if one can be rotated by a multiple of 90 degrees and placed on top of the other so they are completely matched. You are not allowed to flip a window inside out.

The windows are aligned regularly 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 design have the same dimensions.

Input

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

Output

Output a single integer — the number of different kinds of windows in the design.

Sample input 1

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

Sample output 1

4

Sample input 2

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

Sample output 2

4