Problem D. Dog

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Let us talk about the big elephant in the room: you've had a enormous dog in your room for a while now and you need to hide it without raising suspicion, since you want to keep the animal. Hiding an animal of this size is difficult: if you use a lot of space, it is obvious that you are hiding something from your landlord. Hence, you want to use as little space as possible to hide the dog.

When the dog was placed against the wall, you took a black and white picture of the animal. Looking around in the house, the only tools you found to hide the dog with were empty tin cans. The dimensions of the tin cans correspond with 2×2 pixels in the picture and these cans cannot overlap. So, you can make a *dog* and if landlord asks why you have cans in the shape of a dog, you simply say it is a bad joke of yours.

The position of each can has to exactly correspond to a block of 2×2 pixels in the picture, and they cannot be shifted or rotated to only partially cover some pixels. Furthermore, cans cannot float in the air, so every can has to be supported either by the floor, which is just below the bottom row of the picture, or by another can, for which at least one of the left and right half must directly rest on another can. The structure does not otherwise need to be balanced.

What is the minimum number of cans needed to hide the dog?

Input

The input consists of:

- One line containing two integers n ($2 \leq n \leq 100$) and m ($2 \leq m \leq 10$), the height and width of your room. Both n and m are even.
- n lines, each containing m characters that are either '`.`' or '`#`', where '`#`' marks a position that needs to be hidden by a can.

Output

Output the minimal number of 2×2 cans that is required to hide the dog in the room.

Examples

standard input	*standard output*
4 4 .... ..#. .##. ##..	3
4 4 #.#. .... #...	4
14 8 ....##.. ...###.. ....##.. .....#.. ...####. ..#####. .######. .#####.. .###.... .##..... ..##.... ...#.... .###....	15