Problem H. Fence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Consider a rectangular $n \times m$ grid consisting of symbols '.' and '*'. A *house* is a connected component of '*' cells. Two cells are connected if they share a common **side**. A *fence* is a closed polyline without self-touchings and self-intersections which consists of segments corresponding to cell sides.

You have to build a fence with the following three properties:

1. The fence must not have common points with grid borders.
2. The fence must not have common points with houses which are **outside** the fence.
3. Each segment of the fence must be outside any house, in other words, there must be '.' on at least one side of each segment.

Calculate the maximum **number** of houses which can be enclosed by the described fence. If the fence cannot be build, this number has to be 0.

Input

The first line contains two integers n and m, the dimensions of the grid ($1 \le n, m \le 10^3$).

Each of the next n lines is a string of length m consisting of characters '.' and '*'. Together, these lines describe the grid.

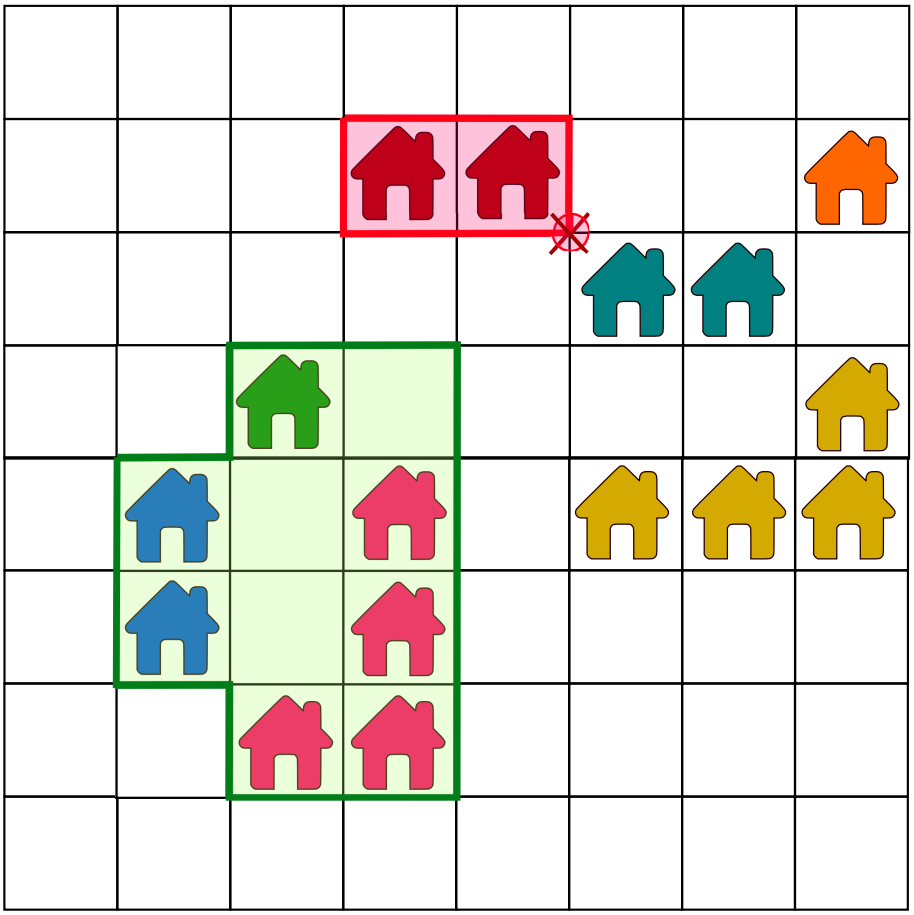
Output

Print one number: the maximum number of houses which can be enclosed by a fence with the three required properties.

Examples

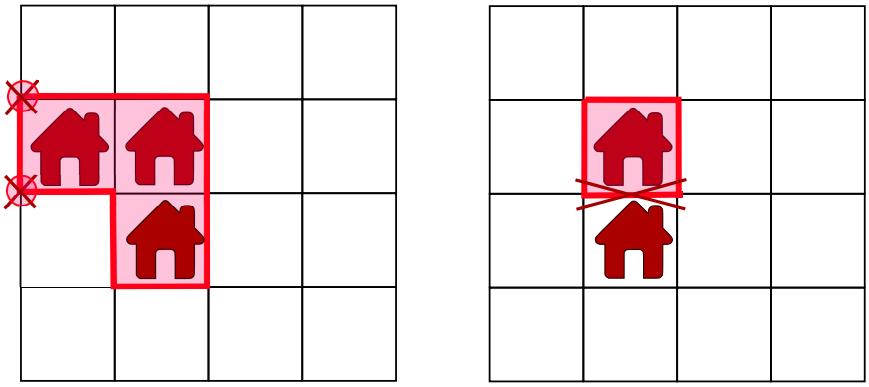
standard input	standard output
8 8 ....**..* .....**. ..*....* .*.*.*** .*.*.... ..**....	3
3 4 .*.. **** ..*.	0

Note



The green polyline near the center which encloses three houses is one of the ways to build a fence optimally.

The red polyline near the top is a bad fence because it touches a house which is outside the fence.



These two pictures show invalid ways to construct a fence. The left picture shows the case where the fence touches the outside boundary. On the right picture, the fence goes through a house, which is unacceptable.