

F. Pasha and Pipe

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

On a certain meeting of a ruling party "A" minister Pavel suggested to improve the sewer system and to create a new pipe in the city.

The city is an $n \times m$ rectangular squared field. Each square of the field is either empty (then the pipe can go in it), or occupied (the pipe cannot go in such square). Empty squares are denoted by character '.', occupied squares are denoted by character '#'.

The pipe must meet the following criteria:

- the pipe is a polyline of width 1,
- the pipe goes in empty squares,
- the pipe starts from the edge of the field, but not from a corner square,
- the pipe ends at the edge of the field but not in a corner square,
- the pipe has at most 2 turns (90 degrees),
- the border squares of the field must share **exactly two** squares with the pipe,
- if the pipe looks like a single segment, then the end points of the pipe must lie on distinct edges of the field,
- for each non-border square of the pipe there are **exactly two** side-adjacent squares that also belong to the pipe,
- for each border square of the pipe there is **exactly one** side-adjacent cell that also belongs to the pipe.

Here are some samples of **allowed** piping routes:

....##	.*. .#
*****	****.	.***.
..#..	..#*.	..#*.
#...#	#...*#	#...*#
.....	...*.	...*.

Here are some samples of **forbidden** piping routes:

.**.#	*...#	.*.*#
.....	****.	.*.*.
..#..	..#*.	.*#*.
#...#	#...*#	#*.*#
.....	...*.	.***.

In these samples the pipes are represented by characters '*'.

You were asked to write a program that calculates the number of distinct ways to make exactly one pipe in the city.

The two ways to make a pipe are considered distinct if they are distinct in at least one square.

Input

The first line of the input contains two integers n, m ($2 \leq n, m \leq 2000$) — the height and width of Berland map.

Each of the next n lines contains m characters — the map of the city.

If the square of the map is marked by character '.', then the square is empty and the pipe can through it.

If the square of the map is marked by character '#', then the square is full and the pipe can't through it.

Output

In the first line of the output print a single integer — the number of distinct ways to create a pipe.

Examples

input
3 3# ...
output
3

input
4 2
output
2

input
4 5 #...# #...# ###.# ###.#
output
4

Note

In the first sample there are 3 ways to make a pipe (the squares of the pipe are marked by characters ' * '):

.*.

.*#

.*.

.*.

**#

...

...

**#

.*.