## 

# todo

In general to found specific strings in other strings.

## **Goal**

There are plenty of animals in the forest. Given a visual representation of the forest, count wolfs and owls that are present.

A wolf looks like this:

|\\_/|

An owl looks like this:

(oo)

**Input**

**Line 1:** two integers W and H separated by a space.

**Next** H **lines of length** W**:** rows describing the forest map

**Output**

**Line 1:** An integer representing the number of wolfs found

**Line 2:** An integer representing the number of owls found

**Constraints**

2 ≤ W ≤ 60

2 ≤ H ≤ 16

**Example**

**Input**

7 2

/|\\_/|\

/|(oo)\

**Output**

1

1

Goal

There are plenty of animals in the forest. Given a visual representation of the forest, count wolfs and owls that are present.

A wolf looks like this:

|\\_/|

An owl looks like this:

(oo)

Input

Line 1: two integers W and H separated by a space.

Next H lines of length W: rows describing the forest map

Output

Line 1: An integer representing the number of wolfs found

Line 2: An integer representing the number of owls found

Constraints

2 ≤ W ≤ 60

2 ≤ H ≤ 16

Example

Input

7 2

/|\\_/|\

/|(oo)\

Output

1

1

# CODE

/\*\*

\* Auto-generated code below aims at helping you parse

\* the standard input according to the problem statement.

\*\*/

var inputs = readline().split(' ');

const W = parseInt(inputs[0]);

const H = parseInt(inputs[1]);

for (let i = 0; i < H; i++) {

const ROW = readline();

}

TESTS:

**01** Test 1

Input

7 2

/|\\_/|\

/|(oo)\

Output

1

1

TEst 2

02 Test 2

**Input**

11 3

/|\ /|\ /|\

/|\ /|\ /|\

/|\ /|\ /|\

**Output**

0

0

**Test 3**

**Input**

30 10

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|o/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\\_/|/|\/|oo|\/|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/\_\/|\/|\/|\/|\/|\/|\/|\

/|\(oo)|\/|\/|\|\\_/|\/|\o|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\/|\/(oo)\/|\/|\/|\

/|\|\\_/|\/|\/|\/|\/|\/|\_/|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

Output

3

2

Test 4 :

Input:

60 16

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|o/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/\_\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\(oo)|\/|\/|\|\\_/|\/|\o|\/|\/|\/|\|\\_/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\/|\/(oo)\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\|\\_/|\/|\/|\/|\/|\/|\_/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\|\\_/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\(oo)|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\/|\

4

3

Output:

4

3