

# Battleships

---

**Time limit:** 0.6s    **Memory limit:** 16M

---

You and a friend are playing the classic game of Battleships. You each have a grid consisting of  $M$  rows of  $N$  cells ( $1 \leq N, M \leq 2000$ ). Each cell is either empty or contains a player's ship (in this version of the game, all ships are the size of one cell). The goal of the game is to destroy all of the opponent's ships by hitting individual cells.

You and your friend have bet tons of CompSci points on this game. Unfortunately, your friend is completely owning you. So desperate times call for desperate measures.

You know for a fact that you can distract your friend for a brief moment by telling him that a famous programmer is behind him, but this trick will only work exactly once (programmers are so predictable). While he isn't looking, you'll have time to snatch up some of his ships with one hand. Your hand can cover a square of exactly  $S \times S$  cells ( $1 \leq S \leq \min(N, M)$ ), and you can gather all the ships within such a square at once.

Of course, your friend is no fool, so he's got his grid well concealed. As such, you don't know anything about it except its size, so when the time comes, you'll just choose a random square of size  $S \times S$  that's completely within the grid.

As usual, these bets attract large crowds. One of the bystanders who can see your opponent's grid knows your plan, and is curious as to the expected number of ships that you will grab (in other words, the average number of ships out of all the possible snatches you could make). Nerdy though he is, he can't calculate it in his head, so he runs to a computer and codes up a program...

## Input Specification

---

Line 1: 3 integers -  $M$ ,  $N$ , and  $S$

The next  $M$  lines:  $N$  characters each, representing your opponent's grid - an  represents a ship, while a  represents an empty cell.

## Output Specification

---

A single number - the expected number of ships that you'll grab. It must be within  $10^{-8}$  of the correct answer.

## Sample Input

---

```
3 4 2
XX.X
XX..
.X..
```

## Sample Output

---

2

## Explanation

---

There are 6 possible areas you could pick, yielding this many ships each:

4 2 1  
3 2 0

This is a total of 12 ships, for an average of exactly 2.