

Escape

Input file: **standard input**
Output file: **standard output**
Time limit: 4 seconds
Memory limit: 512 megabytes

After dealing with numbers, Evirir the dragon is too tired and he decides to play a video game.

Evirir the dragon is playing a video game named **Escape!**

The game takes place on a grid with N rows and M columns. The rows are numbered $1, 2, \dots, N$ from top to bottom and the columns are numbered $1, 2, \dots, M$ from left to right. The cell on row i and column j is denoted by (i, j) . Two different cells are **adjacent** if they share a side (so each cell is adjacent to at most 4 cells: up, down, left, and right). Formally, cell (i, j) is adjacent to $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$, and $(i, j + 1)$ (if they exist).

Each cell in the grid is one of the four types below, and each cell type is represented by a character:

- ‘.’ denotes an empty cell
- ‘d’ denotes a cell with a dog
- ‘e’ denotes an escape door
- ‘#’ denotes a wall cell

Evirir wants to reach an escape door and avoid dogs. Initially, Evirir has **2 HP** (health points) and is on a non-wall cell. Every second, the following happens in order:

1. If Evirir has less than or equal to 0 HP, then he loses the game immediately.
2. If Evirir is at an escape door (and has at least 1 HP), then he wins the game immediately.
3. Evirir does not move or moves to an adjacent cell that is **not** a wall cell.
4. Then, every dog does not move or moves to an adjacent cell that is **not** a wall cell. Multiple dogs can move to the same cell.
5. Then, for every dog in the same cell as Evirir, the dog bites Evirir and Evirir’s HP decreases by 1. Then, the dog disappears from the grid forever.

Furthermore, **before starting the game, Evirir must first decide the sequence of cells he will move to**. That is, Evirir must fix his moves without knowing the dogs’ moves. The dogs may then look at Evirir’s plan and move accordingly.

We call a cell **winning** if and only if it is not a wall cell and Evirir can win if he starts at the cell, moves according to his plan, and wins **regardless of how the dogs move**. Note that Evirir can start at escape gates or on a cell with a dog.

Since Evirir the dragon is lazy, he asks you to determine the number of winning cells in the grid.

Input

The first line consists of two integers, N and M ($1 \leq N, M \leq 3000$).

Then, N lines follow, describing the grid cells. Each line consists of M characters. The j -th character in line i denotes the cell type of cell (i, j) .

Output

Print one integer, the number of **winning** cells.

Scoring

Subtask 1 (11 points): There is at most one dog on the grid.

Subtask 2 (13 points): $N = 1$

Subtask 3 (15 points): $N, M \leq 10$

Subtask 4 (15 points): $N, M \leq 40$

Subtask 5 (17 points): There is at most one escape door.

Subtask 6 (19 points): $N, M \leq 2000$

Subtask 7 (10 points): No other constraints.

Example

standard input	standard output
4 5 ...d. .e#e. ..d#d .e.d.	14

Note

Let us say we are starting at cell (4,5) and our path is shown below.

```
...d.  
.e#e.  
..d#d  
.XXXX
```

X here shows the path that is chosen.

With this, the dog at (4,4) remains stationary and dog at (3,3) goes downwards immediately after the player makes the first move.

Before the game starts, player has 2 HP at is currently at (4,5).

```
...d.  
.e#e.  
..d#d  
.e.dP
```

P shows the position of the player.

Now he moves to the left, there is a dog at cell (4,4), so he has only 1 HP remaining. At the same time the dog at cell (3,3) moves down and the grid now looks like this:

```
...d.  
.e#e.  
...#d  
.edP.
```

It is obvious that the player will lose at the next move, immediately after the player moves to the left. It can be observed that cell (4,5) is not a winning cell after considering all possibilities of routes.