Jumppy and the Grid

Jumppy likes to jump! One day Jumppy went to a park. Jumppy can jump all over the park.The park can be thought of as a square grid with square cells of side length 1. The contents of the grid is either 0(zero) or X. There are certain things which Jumppy likes. They are:

->Jumppy likes rectangles.
->Jumppy likes X.

Therefore Jumppy will jump only in the rectangles consisting of X. A rectangle is defined as follows:

1)The whole rectangular region should contain only X.
2)The rectangle should be surrounded with 0 or the boundary of the grid.
3)The diagonally adjacent cell(see the definition) of the corner of the rectangle may be X or 0. (Refer to the first example).

Diagonally adjacent cell: Suppose the given cell has coordinates (p,q) then its diagonally adjacent cells would have coordinates (p+1,q+1) , (p+1,q-1) , (p-1,q+1) , (p-1,q-1).

Now Jumppy is curious how many rectangles are there in the park. Help Jumppy find the number of rectangle.

INPUT:
An integer n denoting the side of the grid.
Then n lines follow each containing a string of n characters describing the square grid. All the characters will be either 0 or X.

OUTPUT:
Output the number of rectangles in the given grid.

CONSTRAINTS:
0 < n <= 1000

EXAMPLES:

INPUT:

```
4
XX00
XX00
00XX
00XX
```

OUTPUT:

```
2
```

INPUT:

```
5
00000
0XXX0
0XXX0
0XXX0
00X00
```

OUTPUT:

```
0
```

INPUT:

```
5
00000
0XXX0
0X0X0
0XXX0
00000
```

OUTPUT:

```
0
```

INPUT:

```
3
X0X
0X0
X0X
```

OUTPUT:

```
5
```

EXPLANATION:

Case 1: As can be seen there are two rectangles as highlighted.
Case 2: The grid contains no rectangles because it violets the second condition of definition.
Case 3: The grid contains no rectangles because it violets the first condition of definition.
Case 4: The individual X in this case can be considered as rectangles.