Problem J. CntSQ

Input file	`stdin`
Output file	`stdout`

Squares are fun. Counting them is even more fun.

You're given a binary matrix (containing only 0s and 1s), and your task is to count how many square submatrices have the property that every cell on their border is a 1.

Task

Formally, a square is defined by its top-left corner (i_1, j_1) and bottom-right corner (i_2, j_2), where $i_2 - i_1 = j_2 - j_1$. The border of such a square is the set of all cells (x, y) that satisfy at least one of the following:

- $(x = i_1 \text{ or } x = i_2)$ and $j_1 \le y \le j_2$
- $(y = j_1 \text{ or } y = j_2)$ and $i_1 \le x \le i_2$

Your task is to count how many such squares exist such that all the cells on their border contain the value 1.

Input Data

The first line of the input contains two integers N and M, representing the number of rows and columns in the matrix.

Each of the next N lines contains a string of M digits (0 or 1), with no spaces, representing one row of the matrix.

Output Data

Output a single integer: the number of square submatrices whose borders contain only 1s.

Restrictions and Clarifications

- $1 \le N, M \le 1000$
- A square of side length 1 (a single cell) is also valid if that cell is 1.

Examples

Input file	Output file

3 3
111
101
111
	9

7 7
0000000
0111100
0101111
0100101
0111111
0000011
0000011
	27