Correct squares identification

We are given a rectangular matrix filled with 0's and 1's. We want to detect all so called correct squares in the matrix.

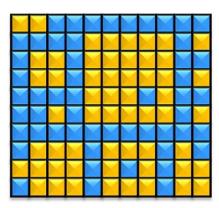A **square** in the matrix is any rectangular shaped area of cells which height is equal to its width.

The **border** of a square is the set of all cells in the square which are adjacent either to the border of the matrix or which are ajacent to at least one cell outside the area. The cell in the square which are not at its border are in its interior.

A square Q is considered to be a **correct square** if

- Q is filled entirely with 1's or there is exactly one 0 in its interior.
- The border Q contains only 1's, it does not contains any 0.
- All cells, which are not part of Q and are adjacent horizontally, vertically or diagonally to the border of Q, contain 0's.

The area of a square is equal to the number of cells in the square.

Image 1. A matrix in which 0's are represented by blue (darker) cells, 1's are represented by yellow (lighter) cells. The matrix contains 4 correct squares with area 1, one correct square with area 4, one correct square with area 9 and one correct square with area 25. The image illustrates Example 1 below.

The task

Find the number of all correct squares in the given matrix and determine the total area they occupy.

Input

The first input line contains two integers M and N representing the number of rows and the number of columns of the input matrix. Next, there are exactly M lines. Each line contains N values, the values correspond to the values in a particular matrix row. Each value is 0 or 1. All values are separated by single space.
It holds $2 \leq M, N \leq 2000$.

Output

The output contains one text line with two integers *S*, *A* separated by space. The value *S* represents number of correct squares in the input matrix, the value *A* represents the total of the areas of all correct squares in the matrix.

Example 1

Input

```
10 11
1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 0 1
0 1 1 0 1 1 1 1 1 0 0
0 1 1 0 1 1 1 1 1 0 1
0 0 0 0 1 1 1 0 1 0 0
0 0 1 0 1 1 1 1 1 0 1
0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 1 1 1 0 1 0
1 1 1 1 0 1 0 1 0 1 1
0 1 1 1 0 1 1 1 0 1 1
```

Output

```
7 42
```

Example 2

Input

```
9 12
0 0 0 0 0 0 1 1 1 1 1 0
0 1 1 1 0 0 1 1 1 1 1 0
0 1 0 1 0 0 1 1 1 1 1 0
0 1 1 1 0 0 1 1 1 0 1 0
0 0 0 0 0 0 1 1 1 1 1 0
1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 1 1 1 0 0 0 1 1
1 1 0 0 1 1 1 0 1 0 1 1
```

Output

```
6 49
```

Example 3

Input

```
15 24
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
1 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 0 1 0
1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0
1 1 1 1 1 1 0 1 1 1 0 1 1 0 0 0 0 0 1 1 1 1 1 0
1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0
1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0
0 0 0 0 0 1 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1
```

Output

```
3 6
```

Public data

The public data set is intended for easier debugging and approximate program correctness checking. The public data set is stored also in the upload system and each time a student submits a solution it is run on the public dataset and the program output to stdout and stderr is available to him/her.
Link to public data set