# Secure Matrix Areas

In this problem, we consider a matrix consisting of $M \times N$ cells.
Each matrix cell is either empty or it is marked by some symbol. The marked cells are named **secure cells**.
A **secure area** in the matrix is a square submatrix $Q$ with the following properties:

- 1. $Q$ consists of at least two rows (or columns).
- 2. There are exactly three secure cells in $Q$.
- 3. The cell in the upper left corner of $Q$ is a secure cell.

The **size of a secure area** is equal to the number of rows (columns) in it.

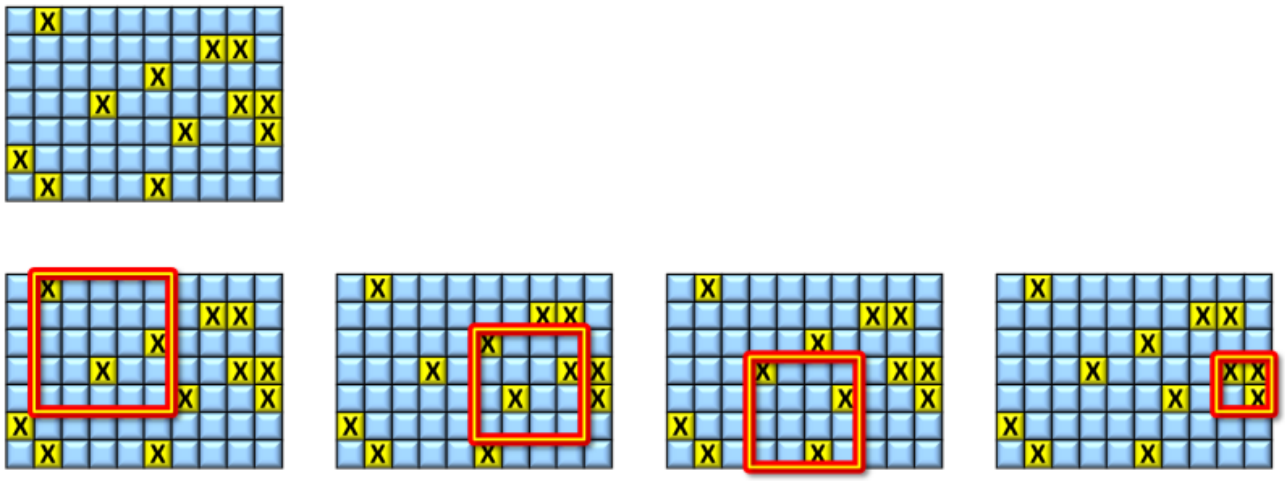

**Image 1.** The top row of the image depicts a 7×10 matrix with 12 secure cells marked by 'X' and highlighted in yellow.
The bottom row of the image shows all four secure areas in this matrix.
Secure areas are highlighted by red bounding boxes. The sizes of the areas are, from left to right, 5,4,4,2.
The image illustrates Example 1 below.

## The task

Determine the number of all secure areas of different sizes in the given matrix.

---

## 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. Value 0 represents empty cell, value 1 represents a secure cell. All values are separated by single space.
It is guaranteed that the input matrix always contains at least one secure area.
It holds $2 \leq M, N \leq 1000$.

## Output

The output contains one or more text lines. Each line contains two integers $s$, $C(s)$, separated by space. Value $s$ represents the size of a secure area, value $C(s)$ represents the number secure areas which size is exactly $s$. The lines are sorted in ascending order of values $s$. Only positive values of $C(s)$ are printed. When $C(s) = 0$ neither $s$ nor $C(s)$ is printed. The output contains no empty line.

### Example 1

**Input**

```
7 10
0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 1 0 0 0 0
0 0 0 1 0 0 0 0 1 1
0 0 0 0 0 0 1 0 0 1
1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 1 0 0 0 0
```

**Output**

### Example 2

**Input**

```
8 9
1 1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 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 0
0 0 0 0 0 0 0 0 0
```

**Output**

### Example 3

**Input**

```
6 6
1 0 0 0 1 0
0 1 0 1 0 1
0 0 1 0 0 0
0 1 0 1 0 0
1 0 0 0 1 0
0 1 0 0 0 1
```

**Output**

```
3 5
```

```
2 1
4 2
5 1
```

```
2 1
3 1
4 1
5 1
6 1
7 2
```

---

## 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**