The Kabbalah Returns (`kabbalah2`)

Luca heard from George about the powers of the *Kabbalah* and now he wants to try for himself. He promptly enrolled in the same course as George, where students routinely spend days and nights studing the wisdoms hidden in the Bible.

Luca learned that in order to become the master of all masters it is crucial to list all the numbers contained in all pages of the Bible, and put those numbers in a $N \times M$ grid. Doing so will make the numbers entagle together and release the hidden power!

After preparing the matrix, Luca needs to find a clue: the year of the edition of the *master book*, a particular Bible which is thought to be the most powerful one. The year is defined as the 4-digit number (without leading zeros) that appears most frequently in the grid. When looking for these occurrences, all sequences of 4 adjacent positions in the 8 cardinal directions are considered (see examples).

> ☞ Among the attachments of this task you may find a template file `kabbalah2.*` with a sample incomplete implementation.

Input

The first line contains 2 integers N and M, the number of rows and the number of columns. The next N lines contain M characters long numbers.

Output

You need to write a single line with two integers: the four digit number with the highest number of occurrences (without leading zeros) and the number of occurrences of that number in the grid.

Constraints

- $4 \leq N, M \leq 1000$.

- Each digit in the grid is between 0 and 9.

- If there are multiple solutions, choose the smallest one (the smallest edition year).

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.
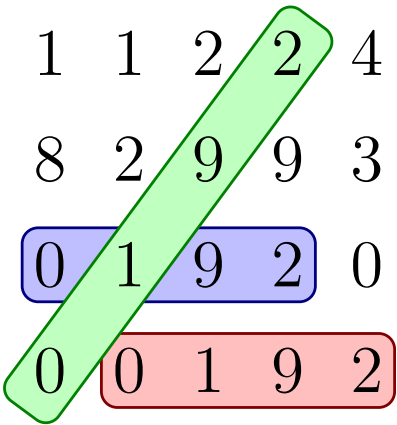
– **Subtask 1** (0 points)　　　Examples.

– **Subtask 2** (10 points)　　　$N = M = 4$ and all the digits are equal.

– **Subtask 3** (10 points)　　　All the digits are equal.

– **Subtask 4** (15 points)　　　All the digits are zeros except exactly one.

– **Subtask 5** (30 points)　　　$N \leq 100$.

– **Subtask 6** (35 points)　　　No additional limitations.

Examples

input	output
4 5 11224 82993 01920 00192	2910 3
10 10 8747832883 2423768395 6346456432 5324532466 3246326365 2564564634 3456346534 4563464365 6435634456 4536346545	6666 8

Explanation

In the **first sample case** the searched numbers are highlighted in the following picture.

$$
\begin{array}{ccccc}
1 & 1 & 2 & 2 & 4 \\
8 & 2 & 9 & 9 & 3 \\
0 & 1 & 9 & 2 & 0 \\
0 & 0 & 1 & 9 & 2
\end{array}
$$

In the **second sample case** there are 8 occurences of 6666, note that many of them overlap and are counted twice because 6666 can be read also from right to left.