

C. The Labyrinth

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given a rectangular field of $n \times m$ cells. Each cell is either empty or impassable (contains an obstacle). Empty cells are marked with '.', impassable cells are marked with '*'. Let's call two empty cells *adjacent* if they share a side.

Let's call a *connected component* any non-extendible set of cells such that any two of them are connected by the path of adjacent cells. It is a typical well-known definition of a connected component.

For each impassable cell (x, y) imagine that it is an empty cell (all other cells remain unchanged) and find the size (the number of cells) of the connected component which contains (x, y) . You should do it for each impassable cell independently.

The answer should be printed as a matrix with n rows and m columns. The j -th symbol of the i -th row should be "." if the cell is empty at the start. Otherwise the j -th symbol of the i -th row should contain the only digit — the answer modulo 10. The matrix should be printed without any spaces.

To make your output faster it is recommended to build the output as an array of n strings having length m and print it as a sequence of lines. It will be much faster than writing character-by-character.

As input/output can reach huge size it is recommended to use fast input/output methods: for example, prefer to use `scanf/printf` instead of `cin/cout` in C++, prefer to use `BufferedReader/PrintWriter` instead of `Scanner/System.out` in Java.

Input

The first line contains two integers n, m ($1 \leq n, m \leq 1000$) — the number of rows and columns in the field.

Each of the next n lines contains m symbols: "." for empty cells, "*" for impassable cells.

Output

Print the answer as a matrix as described above. See the examples to precise the format of the output.

Examples

input
3 3 *.* .*. *.*
output
3.3 .5. 3.3

input
4 5 **.* .**** .*.* *.*.*
output

4	6	.	.	3
.	.	7	3	2
.	6	.	4	.
5	.	4	.	3

Note

In first example, if we imagine that the central cell is empty then it will be included to component of size 5 (cross). If any of the corner cell will be empty then it will be included to component of size 3 (corner).