Problem 3

THE GAME OF LIFE

The Game of Life is not a game in the conventional sense. There are no players, and no winning or losing. Once the "pieces" are placed in the starting position, the rules determine everything that happens later. Nevertheless, Life is full of surprises! In most cases, it is impossible to look at a starting position (or *pattern*) and see what will happen in the future. The only way to find out is to follow the rules of the game.

Rules of the Game of Life
Life is played on a grid of square cells--like a chess board but extending infinitely in every direction. A cell can be *live* or *dead*. A live cell is shown by putting a marker on its square. A dead cell is shown by leaving the square empty. Each cell in the grid has a neighborhood consisting of the eight cells in every direction including diagonals (except the cells along the border).

To apply one step of the rules, we count the number of live neighbors for each cell. What happens next depends on this number.
- A dead cell with exactly three live neighbors becomes a live cell (birth).
- A live cell with two or three live neighbors stays alive (survival).
- A live cell, with four or more neighbors OR one or less neighbors, dies (overcrowding or loneliness).

Note: *The number of live neighbors is always based on the cells before the rule was applied. In other words, we must first find all of the cells that change before changing any of them. Sounds like a job for a computer!*

The input file (DATA3) will contain the information for the starting generation on the grid. The first line will contain two integers, r and c, that represent the number of rows and columns of the grid, respectfully, $3<=r,c<=50$. The next r lines will contain c characters, either a '.' (period), representing a dead cell, or an 'X' representing a live cell.

The output file (OUT3) will contain five lines of data, each representing the number of live cells after the first, fifth, tenth, fiftieth and one hundredth generation, respectfully.

Sample Input	Sample Output

<pre>10 10
..XX..XXXX
.X.XX.XX..
XX....X..X
X...XX..XX
X..X.X....
....
.XXX.....X
.......XXX
......XX..
X.X.X.X...</pre>
	<pre>38
	30
	27
	0
	0</pre>