Input

import java.util.Scanner;

import java.io.File;

class Input {

public static void main(String[] args) throws Exception {

Scanner readFile = new Scanner(new File("inp5"));

int m, n;

m = readFile.nextInt();

n = readFile.nextInt();

readFile.nextLine(); // The first line will contain ''

String[] inputState = new String[m];

Cells.setRows(m);

Cells.setCols(n);

for (int i = 0; i < m; i++) {

inputState[i] = readFile.nextLine();

}

Cells.setState(inputState);

for (int i = 0; i < 100; i++) {

Cells.nextState();

Cells.printState();

System.out.println("State: "+(i+1));

Thread.sleep(600);

if (i != 99)

for (int j = 0; j < 60; j++) System.out.print('\f');

}

}

}

class Cell {

private char life = '-';

private int rowIndex;

private int colIndex;

public Cell(char life, int rowIndex, int colIndex) {

this.life = life;

this.rowIndex = rowIndex;

this.colIndex = colIndex;

}

public char getState() {

return life;

}

public void setState(char life) {

this.life = life;

}

public char getNextState() {

int count = Cells.getNeighborCount(this, rowIndex, colIndex);

if ((count ==2 || count == 3) && this.life == '\*')

return '\*';

else if (count == 3 && this.life == '-')

return '\*';

return '-';

}

}

Cells

class Cells {

private static int rows;

private static int cols;

private static Cell[][] life;

public static void setRows(int rows) {Cells.rows = rows;}

public static void setCols(int cols) {Cells.cols = cols;}

public int getRows() {return rows;}

public int getCols() {return cols;}

public static void setState(String[] state) {

life = new Cell[rows][cols];

for (int i = 0; i < rows; i++)

for (int j = 0; j < cols; j++) {

life[i][j] = new Cell(state[i].charAt(j), i, j);

}

}

public static void printState() {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++)

System.out.print(life[i][j].getState());

System.out.println();

}

}

public static void printState(Cell[][] life) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++)

System.out.print(life[i][j].getState());

System.out.println();

}

}

public static void nextState() {

Cell[][] newLife = new Cell[rows][cols];

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

newLife[i][j] = new Cell(life[i][j].getNextState(), i, j);

}

}

life = newLife;

}

public static int getNeighborCount(Cell key, int rowIndex, int colIndex) {

if (life[rowIndex][colIndex] != key) return -1; // Trying to access other cell's row, column

int count = 0;

for (int i = Math.max(rowIndex-1, 0); i <= Math.min(rowIndex+1,rows-1); i++) {

for (int j = Math.max(colIndex-1, 0); j <= Math.min(colIndex+1, cols-1); j++) {

if (!(i == rowIndex && j == colIndex)) {

if (life[i][j].getState() == '\*') {

count++;

}

}

}

}

return count;

}

}