CSCD 240
Assignment 1
1D & 2D Arrays

PROGRAM SPECIFICATIONS

An asteroid has crashed to earth, and everyone that comes into open contact with it has died. The government has tasked you with saving the world. You think the comet carried a bacterium but you are not sure. You do know how big the organism is but you don't know what will kill it. You will run tests on the organism that tests it growth rate.

The experiment culture dish is a square, divided into 100 smaller squares (10x10). Population in each small square is measured on a four point scale (from 0 to 3). The control information is represented as an array D, indexed from 0 to 15, of integer values and is interpreted as follows:

In any given culture dish square, let K be the sum of that square's growth rate and the growth of the four squares immediately to the left, right, above and below that square (squares outside the dish are considered to have growth 0). Then, by the next day, that dish square's growth will change by D[K] (which may be a positive, negative, or zero value). The total growth cannot, however, exceed 3 nor drop below 0.

Right now any solution will cause an immediate population explosion ([3,3,3, …, 3]). Your task is to find some solution which will cause all the bacteria to die off ([-3, -3, …, -3]), or not grow at all ([0, 0, …, 0]).

Write a program to test and simulate the growth, reading in the number of days to be simulated, the control rules, and the initial population densities of the dish.

For this assignment you will need at least 3 files. The file that contains main will be called AndromedaTester.c.

Input Format

Input to this program consists of:

1. The first line will contain a single integer denoting the number of solutions to be run.
2. The second line will contain a single integer denoting the number of days to be simulated, for that control solution.
3. The next line will contain the solution rule D as 16 integer values, ordered from D[0] to D[15], separated from one another by one or more blanks. Each integer will be in the range -3…3, inclusive.
4. The remaining ten lines of input will describe the initial population density in the culture dish. Each line describes one row of squares in the culture dish, and will contain 10 integers in the range 0…3, separated from one another by 1 or more blanks.

The input file is guaranteed to be properly formatted and contain valid data. You will need to prompt the user for the name of the input file.

Output Format:

The program will produce 10 lines of output for each generation, describing the population growth in the culture dish at the end of that generation. Each line represents a row of squares in the culture dish, and will consist of 10 characters, plus the usual end-of-line terminator. Each generation should be labeled starting with generation 0 (which is the generation read in from the input file). Output will be to a file, via redirection, each control solution run will be clearly denoted.

Each character will represent the population growth at a single dish square, as follows:

Density	Character
0	.
1	!
2	X
3	#

Sample Input

```
2
2
0 1 1 1 2 1 0 -1 -1 -1 -2 -2 -3 -3 -3 -3
3 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 0 0
0 0 0 0 3 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 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
1
0 1 1 1 2 1 0 -1 -1 -1 -2 -2 -3 -3 -3 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 0 0 0 0
0 0 0 0 3 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 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

SAMPLE RUN

```
Solution:

0 1 1 1 2 1 0 -1 -1 -1 -2 -2 -3 -3 -3 -3

Generation: 0
#.........
..........
..........
..........
....#.....
..........
..........
..........
..........
..........


Generation: 1

#!........
!.........
..........
....!.....
...!#!....
....!.....
..........
..........
..........
..........


Generation: 2
##!.......
#!........
!...!.....
...!#!....
..!#X#!...
...!#!....
....!.....
..........
..........
..........
```

```
Solution:

0 1 1 1 2 1 0 -1 -1 -1 -2 -2 -3 -3 -3 0

Generation: 0
..........
..........
..........
..........
....#.....
..........
..........
..........
..........
..........

Generation: 1

..........
..........
..........
....!.....
...!#!....
....!.....
..........
..........
..........
..........
```

CODING SPECIFICATIONS

- You will have a hard coded two dimensional array. Meaning int array[10][10];
- You must use your fileutil.c class – don't forget a header file for it
- In your functions you must pass any array with the square brackets
- In your functions you must pass the array as hardcoded 2 dimensional – meaning function(int array[][10]);
- There is no need for terms like malloc or calloc so you are not allowed to use them.

TO TURN IN

A zip that contains:

- All C files necessary to compile and run your program
  - At least 5 H/C files - cscd240hw1, fileutil.c fileutil.h, your other.h and other.c
  - The file that contains main will be called cscd240hw1.c
- All input files
- All output files.
- Sample run named AndromedaOut.txt
- Makefile with a target of hw1

Name your zip your lastname first letter of your first name hw1.zip (Example: steinershw2.zip)

GET STARTED ASAP