INTRODUCTION TO COMPUTER SCIENCE I
LAB 7
Game of Life prep

1 Conway's Game of Life

We will begin work on Conway's *Game of Life* (read about it!). particular, this will be our first use of *matrices* (a.k.a., *two-dimensional arrays*). The *Game of Life* isn't really a game, but rather a simulated system of cells that "live" and "die" according to some simple rules. The result of these simple rules is arbitrarily complex behavior, which is a bit surprising.

2 Getting started

Begin like so:

1. Make a directory for `lab-7` and change into it.
2. Copy source code from my public directory:
   ```
   $ cp ~sfkaplan/public/COSC-111/lab-7/*.java .
   ```
3. Open and edit source code file `Life.java` with `emacs`. (There is also a file, `Support.java`, that has handy methods used by code in `Life.java`, but you don't need to do anything to this code.)

3 Your assignment

The given code reads in an *initial grid file* that describes the size of a grid of cells, as well as giving the coordinates of the live cells. A grid file is just a text file (you can make one with `emacs`) that looks like this:

```
4 6
0 1
1 2
3 2
3 3
3 4
3 5
```

Specifically, the *first* line of the file specifies the size of the grid, given as as a number of *rows* followed by a number of *columns*. The remaining lines provide the coordines (with the *row* first and the *column* second) of the live cells in the grid.

The code given to you will read this file, construct the grid, and print it, yielding something like:

```
$ java Life foo.txt

. + . . . .
. . + . . .
. . . . . .
. . + + + +


0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

The grid of cells is shown as plus-signs (for *live* cells) and periods (for *dead* cells). The grid below is supposed to contain the count of *live neighbors*—that is, of the eight immediately adjacent cells, how many are themselves alive. Each position in this grid is the *live neighbor count* for the cell in the same position in the cell grid.

Of course, all of these values are `0` because the final method of the source code, `countNeighbors()`, is incomplete. **Your task is to complete this method so that the grid of live neighbor counts is correct.** For example, the above should look like this following:

```
java Life foo.txt

. + . . . .
. . + . . .
. . . . . .
. . + + + +


1 1 2 1 0 0
1 2 1 1 0 0
0 2 3 4 3 2
0 1 1 2 2 1
```

4 How to submit your work

Use the CS submission systems to submit your work. Specifically, you will need to submit your `Life.java` file. You may use either of the following two methods, while connected to `remus` or `romulus`, to use the submission system:

- **Web-based:** Visit the submission system web page.
- **Command-line based:** Use the `~lamcgeoch/submit` command at your shell prompt.

This assignment is due on Thursday, Apr-21, 11:59 pm.