HOMEWORK #1

Title: *Hive mind*

Due Date: <u>Friday, 8 September 2017</u>

Administrative:

- **Naming:**
  - For this assignment, you will submit a single C++ file called 'hivemind.cpp'.
  - Remember to put your name and section at the top of your program file.
- **Submission:**
  - Submit using the 'cssubmit 1575 <section> 1' command, available on the campus Linux machines. Remember to execute the command in the same folder where your homework files exist. There are two sections of Data Structures, so make sure you get the right one!

Robots always work better in groups

Problem and goal:

You have been hired by the Mom's Friendly Robot Company as the chair of the new institutional committee to bring awareness to a grave new problem, robot anxiety disorder, or RAD. Robotic citizens of the hive mind communicate via wireless RF transmissions. All robots are unique. Each robot has a different signal strength for this communication capability, and thus different radii within which they can broadcast to other robot citizens of the hive mind. Outgoing signal strength is defined by a radius parameter, the radius of one robot's outgoing communication range. Robots experience symptoms of RAD if they have **fewer than 2** other robots within range of <u>bi-directional</u> communication. Your goal is to calculate the number of robots who are at risk of RAD, in a given 2D grid of the hive mind. To determine the distance between two robots in this digital 2D hive-grid, at you will need to recall from grade-school geometry how to calculate Euclidean distance between two points on a grid.
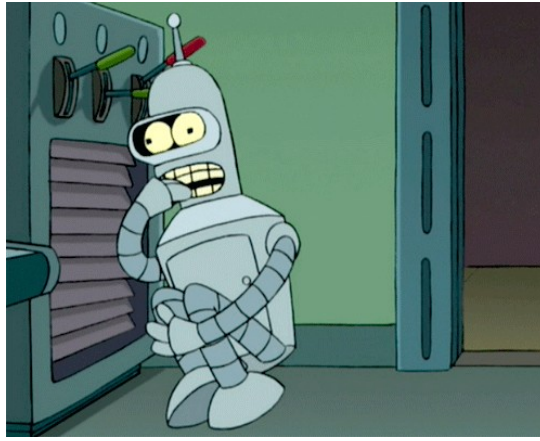
The effects of RAD

Input:

- The **first line of the input states the number of test cases T**, then T test cases follow.
- Each test case consists of a data grid and the locations where the robots currently reside. **A test case begins with the numbers H and W**; the height and width of the RF data grid. **H rows of W data points follow**, describing the number of fish detected by the sonar.

The provided file: *sampleinput.txt,* contains what you would type as input to the program, line by line. Input is supposed to be via `cin`, despite what you might infer by us giving you a text file. You can type or copy each line of input in, or if you wanted to test your program more efficiently, you could use **bash** to redirect the *sampleinput.txt* to `cin`, while remembering the rules for spacing and `cin`, e.g.,

$ ./yourexecutablefile < sampleinput.txt

This makes your program think that the input in the file is being typed out.

Output:

You should output the result after each round of diagnosis, bolded below in the context of the execution sequence, for example:

1. First, you should accept input for the number of rounds of diagnosis.
2. First round
    a. Then, you should accept the size of the hive mind grid for the first round (h w)
    b. Then, you should accept the 2D array of signal strengths (a matrix, row by row)
    c. **Then you should output via cout, the result for the first round.**
3. Second round
    a. Then, you should accept the size of the hive mind grid for the first round.
    b. Then, you should accept the 2D array of signal strengths.
    c. **Then you should output via cout, the result for the second round.**
4. ...

Implementation and execution:

- Your program should expect all input to come from 'cin', and all your output should be to 'cout'.
- Given that you do not know beforehand how large a hive-mind grid is, your program should dynamically allocate a 2D Array after the width and height of a grid is read.
- Make sure to de-allocate the 2D Array after you find the answer and before your program moves on to process the next grid.

Grading:

- You will be graded first and foremost on a program that generates the correct output, in answer and format.

Sample input and output:

```
$./hivemind < sampleinput.txt     // execute
3                                 // input: number of rounds
5 8                               // input: size of hive mind
1 0 0 0 1 1 0 1                   // input: first row of signal strengths
0 0 0 0 1 1 0 0
0 1 0 2 0 2 0 0
0 0 0 0 0 0 0 0
1 0 0 2 0 2 0 1
Hive mind 1 has 5 isolated members.          // output: via cout
6 6                                           // input: size of hive mind
0 0 0 5 0 0                                    // input: first row of signal strengths
0 1 0 0 0 0
0 0 0 3 0 5
5 0 3 0 0 0
0 0 0 0 1 0
0 0 5 0 1 0
Hive mind 2 has 3 isolated members.          // output: via cout
7 12                                          //...
1 0 0 3 0 0 0 0 0 3 0 0
0 3 0 0 0 3 0 3 0 0 0 0
0 0 0 0 0 0 0 0 0 2 0 3
3 0 0 2 0 0 3 5 4 0 0 0
0 0 0 0 0 0 0 0 6 3 0 0
0 3 0 0 0 3 0 3 1 1 0 0
1 0 0 3 0 0 0 0 3 0 1 0
Hive mind 3 has 4 isolated members.
```