CMPE 436: Assignment 2
Fall 2017 Instructor: Alper Sen
Due Date: midnight, Oct 18, 2017 Demo Date: TBD

Question 1- (50 points)
Write a multi-class **multi-threaded** Java program that simulates the *game of life,* which was described in Assignment 1. Each cell will have its own thread dedicated to it to compute the cell's value in the next generation. This thread will be embedded in a class that is instantiated M*N times by the driver, once for each cell in the grid. Thread synchronization will be done with semaphores. Do not create new threads for generations after the first.

The primary problem you have to solve is coordinating all the cell threads during each generation. A cell thread cannot start computing the new cell value for the next generation until all other cells have completed their computation for the current generation. Use semaphores, that is, `P()` and `V()` to solve this problem. You can use BinarySemaphore and CountingSemaphore implementations discussed in class (Do not use Java concurrency library). You will find it extremely helpful to use **barrier synchronization**.

Question 2- (20 points)
Write a **multithreaded Java** program with 3 threads that has a deadlock in it. Demonstrate the deadlock.

Question 3- (20 points)
Write a **multithreaded Java** program with 3 threads that has a race condition in it. Demonstrate the race condition.

Guidelines:

1- Email your assignment solution.
2- Add the following to the start of your programs.
    // your name // your student ID // your email address
    // CMPE436-Assignment *n* - where *n* is the assignment number (1, 2, ...)
3- Add comments to your programs. Program clarity is very important. You get graded on this.
4- Also add a README.txt file to explain your programs.
5- Demo your homework to the instructor. Bring your laptop.
6- DO NOT DISCUSS WITH YOUR CLASSMATES. DO NOT USE SOLUTIONS FROM OTHERS. CHEATING WILL NOT BE TOLERATED.