Assignment 4 Programming Hints

## compiling on Linux with thread included:

You may encounter errors compiling your source code when you include thread library on Linux. That's because on Linux the thread library should be dynamically linked through your compile command. You should use:

g++ -std=c++11 source.cpp -lpthread -o cellculture

## sleep_for() - use chrono library and milliseconds instead of seconds:

e.g., A genesis cell lives half its life

std::this_thread::sleep_for(std::chrono::milliseconds((int)(0.5 * 1000 * lifeTime)));

## call thread.join() for your monitor thread:

You should join your monitor thread to your main thread so that the main thread will wait for the monitor thread

## call thread.detach() after you create a cell thread:

You should detach your cell threads to let them run in the background

## use a Counter class which has a counter and a mutex to serve as the global counter:

```
class Counter
{
    int number;
    std::mutex mutex;
public:
    Counter() :number(0){}
    int getNumber()   {     return number; }
    void increase()
    {
        mutex.lock();
        number++;
        mutex.unlock();
    }
    void decrease()
    {
        mutex.lock();
        number--;
        mutex.unlock();
    }
};
```

use std::ref(counter/mutex) to pass an object or a mutex to thread:

e.g.,

```
Counter c;
std::thread cellThread(…, std::ref(c));
```