CSE 120: Homework 2

Merrick Qiu

1.
```
struct lock {
    bool* flag = false;
}

void acquire (struct lock *) {
    bool lockedFlag = true;
    while (true) {
        if (lock == 0) {
            XCHG(lockedFlag, lock->flag);
            break;
        }
    }
}

void release (struct lock *) {
    (lock->flag)* = false;
}
```

2. (a) main, A, B, A
   (b) fee foe foo far fie fum fun
   (c) The currentThread will be main and the ready and wait queues will be empty.

3. (a)
```
monitor Barrier {
    int doneThreads = 0;
    Condition barrier = new Condition();

    void Done(int n) {
        doneThreads++;
        if (doneThreads = n) {
            doneThreads = 0;
            barrier.wakeAll();
        } else {
            barrier.wait();
        }
    }
}
```

   (b)
```
class Barrier {
    int doneThreads = 0;
```

```
        Lock lock = new Lock();
        Condition barrier = new Condition();

        void Done(int n) {
            acquire(lock);
            doneThreads++;
            if (doneThreads = n) {
                doneThreads = 0;
                barrier.wakeAll();
            } else {
                barrier.wait();
            }
            release(lock);
        }
    }

class Surfing {
    enum State { calm, breaking; }
    enum Direction { LEFT, RIGHT, BOTH; }

    State oceanState = calm;
    Direction waveDirection = BOTH;
    Condition leftDirection = new Condition();
    Condition rightDirection = new Condition();
    Lock lock = new Lock();

    void paddle (Direction dir) { // invoked by surfer threads
        acquire(lock);
        if (oceanState == breaking && waveDirection == dir OR BOTH) {
            // successful paddle
            release(lock);
            return;
        }
        (dir == left ? leftDirection : rightDirection).wait();
        release(lock);
    }
    void wave (Direction dir) {   // invoked by the ocean thread
        acquire(lock);
        oceanState = breaking;
        waveDirection = dir;
        if (dir == left OR BOTH) {
            left.wakeAll();
        }
        if (dir == right OR BOTH) {
            right.wakeAll();
        }
```

```
    release(lock);
}
void done () {                    // invoked by the ocean thread
    acquire(lock);
    oceanState = calm;
    release(lock);
}
```

4. From the diagram, we can see that having the students work in the order of Bertrand, Dag, Chloe, then Annabelle lets everyone get their work done.