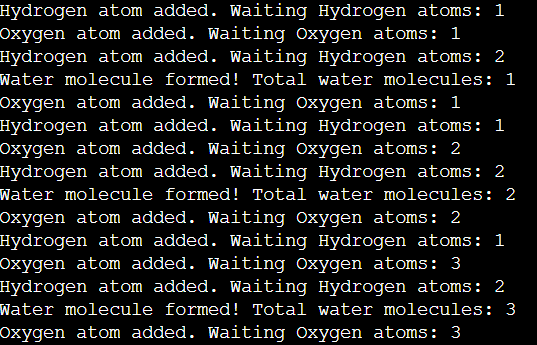
**Question:**

1. Building H2O
   1. Correctness constraints
      1. Each hydrogen thread waits to be grouped with one other hydrogen and oxygen before returning
      2. Each oxygen thread waits for two other hydrogens before returning
      3. Only one thread access shared state at a time
   2. There is only one condition any thread will wait for, i.e. a water molecule being formed. However, it will be necessary to signal hydrogen and oxygen threads independently, so  we will use two condition variables, waitingH and waitingO.
   3. It will be necessary to know the number of hydrogen and oxygen threads in the monitor.   But it would be more useful to know how many hydrogen and oxygen threads have been assigned and have not been assigned to water molecules; let these be int wH (number of waiting hydrogens), wO (number of waiting oxygens), aH (number of assigned hydrogens), and aO (number of assigned oxygens). These are all initialized to 0.
   4. Algorithm:

Hydrogen() {  
  lock.acquire();  
  wH++;  
  
  // while not allowed to leave  
  while (aH == 0) {  
    // try to make a water molecule  
    if (wH >= 2 && wO >= 1) {  
      wH-=2; aH+=2;  
      wO-=1; aO+=1;  
      waitingH.signal();  
      waitingO.signal();  
    }  
    // else wait for somebody else to  
    else {  
      waitingH.wait();  
    }  
  }  
  aH--;  
  lock.release();  
}  
  
Oxygen() {  
  lock.acquire();  
  wO++;  
  
  // while not allowed to leave  
  while (aO == 0) {  
    // try to make a water molecule  
    if (wH >= 2 && wO >= 1) {  
      wH-=2; aH+=2;  
      wO-=1; aO+=1;  
      waitingH.signal();  
      waitingH.signal();  
    }  
    // else wait for somebody else to  
    else {  
      waitingO.wait();  
    }  
  }  
  aO--;  
  lock.release();  
}

**Sample Output:**



**Note: Please do not use struct{};.**