# Operating Systems

Homework 5

## Question 1

// shared memory  
semaphore match\_sem(0);  
semaphore paper\_sem(0);  
semaphore tobacco\_sem(0);  
Semaphore agent\_sem(1);

// process with matches (needs paper and tobacco)  
while (true) {  
 if (paper\_sem > 0 && tobacco\_sem > 0) {  
 paper\_sem.wait();  
 tobacco\_sem.wait();  
 smoke();  
 agent\_sem.signal();  
 }  
}

// agent  
while (true) {  
 agent\_sem.wait();  
 int random = rand() % 3;  
 if (random != 2) {  
 match\_sem.signal();  
 }  
 if (random != 1) {  
 paper\_sem.signal();  
 }  
 if (random != 0) {  
 tobacco\_sem.signal();  
 }  
}

## Question 2

// shared memory  
semaphore mutex(1);  
semaphore barber(0);  
semaphore full\_chairs(0);  
queue<customer> chairs;{  
int empty\_chairs = 3;

// barber code  
while (true) {  
 full\_chairs.wait();  
 customer current\_customer = chairs.dequeue();  
 mutex.wait();  
 ++empty\_chairs;  
 mutex.signal();  
 cut\_hair(current\_customer);

// customer code  
mutex.wait();  
if (empty\_chairs) {  
 --empty\_chairs;  
 mutex.signal();  
 chairs.enqueue(a\_cusomer);  
 full\_chairs.signal();  
} else {  
 mutex.signal();  
}