CS341 #17. Producer Consumer, Semaphores, Condition Variables. Barriers & Reader Writer Problem

1. Producer Consumer & Counting Semaphores (review)

Assume buffer is an array of length 16. Better names for s1?s2?

```
void add(value) {
 sem_wait(&sem_s1)
 buffer[ (in++) & 15 ] = value;
 sem_post(&sem_s2);
}
```

```
remove() {
 sem_wait(&sem_s2);
 result = buffer[ (out++) & 15 ];
 sem_post(&sem_s1);
 return result;
}
```

Q. What are 'sem_s1' and sem_s2? When do they block?

Q. What should be their initial values?

Q. What if sem_s1 was only initialized to 7? Would the producer consumer still work? to 32?

Q. What is missing from the above code? When would it matter?

Q. Could you implement a producer consumer queue using condition variables instead?

2. Fix the following multithread code to be thread safe, and use condition variables to avoid busy waiting

```
#define N (8)
pthread_cond_t cvs[N];
pthread_mutex_t locks[N];
int data[N];
int quit;

void init() {
  for(int i =0; i < N;i++) {

    pthread_cond_init(cvs + i, NULL);
    pthread_mutex_init(locks + i, NULL);
  }
}

// Wait until data[i] > 1, then subtract 2 and increment data[i+1]
void runner(void*arg) { // For N-1 threads. Each thread gets a value 0 to N-2
  int i = (int) i;
  while(!quit) {
     while(data[i] < 2) {
        sleep for a bit
     }
     data[i] -= 2;
     data[i+1] ++;
   }
}

void modify(int index, int amount) {


  data[index] += amount;



}
```

3.Counting Semaphore Quick Review I. choose {will always / may / will never} :

sem_post ___________ block   sem_wait ___________ block.

3.Counting Semaphore Quick Review II

10 threads call `sem_wait`. 3 threads immediately continue, the other 7 are blocked. Then `sem_post` is called twice (2). How many additional threads will continue?

4. Three classic / well known synchronization problems:

Barrier

Producer Consumer

Reader-Writer Problem

5. pthread barriers

```
pthread_barrier_init( &barrier, _______);
pthread_barrier_destroy(&barrier)

pthread_barrier_wait( &barrier)
```

Return values?

```
0
PTHREAD_BARRIER_SERIAL_THREAD
```

6. Use a CV to implement a single-use barrier until all 8 threads have reached the barrier.

7. Post-lecture challenge:

i) Can you make a barrier using only counting semaphores?

ii) Can you make a barrier using only mutex locks?