ECSE 427
Saif Elkholy
260606967
Assignment 3

Part 1

1. The two semaphores are used to synchronise the producer (airplane dropping passengers in the queue) and the consumer (taxis taking passengers from the queue) threads. This means that the semaphores send a signal when each thread can add or remove a passenger from the queue. Notice that the job of the semaphore does not include protecting a shared resource (the queue). This is where the mutex plays a role. A mutex ensures that no matter what type of thread (producer or consumer), only one thread is modifying the queue at a given moment. This is to prevent a possible race condition or any other anomalies with multiple threads accessing a shared resource.
2. No. Eventually, the producer will have to wait for a consumer semaphore signal. This means that the consumer thread will eventually get CPU time as the producer must wait.
3. Mutexes ensure that multiple threads don't access a shared resource at once. Mutexes prevent a race condition, semaphores don't
4. We need to have two semaphores to be able to signal to the producer thread when there exists an empty spot in the queue. Conversely, we need to be able to signal to the consumer thread when there exists a passenger in the queue. With one semaphore, you would only be able to signal one type of thread (producer/consumer). This would mean that the other thread would have to employ a busy waiting approach to know when it can run.