6.15 Servers can be designed to limit the number of open connections. For example, a server may wish to have only *N* socket connections at any point in time. As soon as *N* connections are made, the server will not accept another incoming connection until an existing connection is released. Explain how semaphores can be used by a server to limit the number of concurrent connections.

A.1 The Producer/Consumer (or Bounded-Buffer) Problem: In multithreaded programs there is often a division of labor between threads. In one common pattern, some threads are producers and some are consumers. Producers create items of some kind and add them to a data structure; consumers remove the items and process them.

- While an item is being added to or removed from the buffer, the buffer is in an inconsistent state. Therefore, threads must have exclusive access to the buffer.
- If a consumer thread arrives while the buffer is empty, it blocks until a producer adds a new item.

In the kernel of the operating system, though, there are limits on available space. Buffers for things like disk requests and network packets are usually fixed size. In situations like these, we have an additional synchronization constraint:

- If a producer arrives when the buffer is full, it blocks until a consumer removes an item.

Write a program that can correctly coordinate the producers and consumers and their depositing and retrieving activities.

A.2 Readers/Writers Problem: Reader-Writer Problem, pertains to any situation where a data structure, database, or file system is read and modified by concurrent threads. While the data structure is being written, or modified it is often necessary to bar other threads from reading, in order to prevent a reader from interrupting a modification in progress and reading inconsistent or invalid data.

As in the producer-consumer problem, the solution is asymmetric. Readers and writers execute different code before entering the critical section. The synchronization constraints are:

1. Any number of readers can be in the critical section simultaneously.
2. Writers must have exclusive access to the critical section.

In other words, a writer cannot enter the critical section while any other thread (reader or writer) is there, and while the writer is there, no other thread may enter. Use semaphores to enforce these constraints, while allowing readers and writers to access the data structure, and avoiding the possibility of deadlock.

CHAPTER 7 (Silberschatz – Deadlocks)

7.14 A single-lane bridge connects the two Vermont villages of North Tunbridge and South Tunbridge. Farmers in the two villages use this bridge to deliver their produce to the neighboring town. The bridge can become deadlocked if both a northbound and a southbound farmer get on the bridge at the same time (Vermont farmers are stubborn and are unable to back up). Using semaphores, design an algorithm that prevents deadlock. Initially, do not be concerned about starvation (the situation in which northbound farmers prevent southbound farmers from using the bridge, and vice versa).

7.15 There is a bridge which is aligned along the east-west direction. This bridge is too narrow to allow cars to go in both directions. Hence, cars must alternate going across the bridge. The bridge is also not strong enough to hold more than three cars at a time. Find a solution to this problem which does not cause starvation. That is, cars that want to get across should eventually get across. However, we want to maximize use of the bridge. Cars should travel across to the maximum capacity of the bridge (that is, three cars should go at one time). If a car leaves the bridge going east and there are no westbound cars, then the next eastbound car should be allowed to cross. We don't want a solution which moves cars across the bridge three at a time, i.e., eastbound cars that are waiting should not wait until all three cars that are eastbound and crossing the bridge have crossed before being permitted to cross.