

5. Describe the readers-writers problem in general, and explain how the specific pseudo-code shown below uses semaphores to ensure (1) multiple reader threads may access shared data while locking out all writer threads, and (2) a single writer thread is allowed to access the shared data at any one time. The variables `mutex` and `rw_mutex` are binary semaphores initialized to 1.

Writer thread

```
do {
    down(rw_mutex);
    ...
    /* writing is performed */
    ...
    up(rw_mutex);
} while (true);
```

Reader thread

```
do {
    down(mutex);
    read_count++;
    if (read_count == 1)
        down(rw_mutex);
    up(mutex);
    ...
    /* reading is performed */
    ...
    down(mutex);
    read_count--;
    if (read_count == 0)
        up(rw_mutex);
    up(mutex);
} while (true);
```

6. Compare and contrast (a) locks vs. semaphores and (b) condition variables vs. semaphores.