CIS 415 Operating Systems: Worksheet 5 - Semaphore Usage

1. Pizza Synchronization

Processes A, B, ..., Z (a total of 26 processes) are using the semaphore S in order to enforce mutually exclusive use of the shared variable COUNT. S is initialized to 1 and COUNT is initialized to zero. All processes execute the following code once each.

```c
main() {
    P(S);       /* wait(S) */
    COUNT++;    /* increment the counter */
    /* I can eat a whole pizza */
    V(S);       /* signal(S) */
}
```

Subsequently, Prof. Lo orders the number of pizzas specified in the variable COUNT.

a. Which line or lines of code make up the critical section code?
b. What is the maximum number of processes that can be executing in critical section at the same time?
c. What is the maximum number of processes that can be blocked on the semaphore S?
d. What is the maximum number of processes that can be in the ready queue at the same time (considering only these 26 processes)?
e. If each process executes the above code exactly once, how many pizzas will be ordered?
f. If the P and V calls were accidentally omitted, what is the lowest value that COUNT could receive after all 26 processes executed the COUNT++ instruction exactly once? (How many students will get a pizza in this case?)
g. If the P and V calls were accidentally omitted, what is the highest value that COUNT could receive after all 26 processes executed the COUNT++ instruction exactly once? (How much will Professor Lo have spent on pizza assuming each pizza costs $10. Will she be happy about this?)

====================================================================

2. Write pseudo-code to control the number of students who get to watch the special preview of Avatar 4D version. At most 50 students will be allowed to simultaneously watch this video streamed on the Internet to avoid traffic congestion. Define the semaphore, initialize it, and write the code for each student to execute where the critical section code is "DOWNLOAD Avatar". In reality, this situation would not be controlled with semaphores. Why not?

====================================================================

3. Write pseudo-code to ensure that processes A and B both complete execution before process C begins. Define the semaphore (s), initialize, and write the code for each student to execute. Note, there is no critical section here. We are using semaphores to ENFORCE PRECEDENCE - to coordinate processes. Can this be done with flag variables? Why or why not?