Quiz 5

Which of the following variables are shared between the processes in Peterson's solution?

1. int turn
2. boolean flag[2]
3. **both turn and flag**
4. No variables are shared.

What is the correct order of operations for protecting a critical section using a binary semaphore?

1. release() followed by acquire()
2. acquire() followed by release()
3. **wait() followed by signal()**
4. signal() followed by wait()

In _____, the process requests permission to access and modify variables shared with others.

1. entry section
2. **critical section**
3. exit section
4. remainder section

Which of the following critical-section problem's requirements ensures programs will cooperatively determine what process will next enter its critical section?

1. mutual exclusion
2. **progress**
3. bounded waiting
4. none of the rest

Which of the following critical-section problem's requirements limits the amount of time a program will wait before it can enter its critical section?

1. mutual exclusion
2. progress
3. **bounded waiting**
4. none of the rest

The counting semaphore is initialized to _____.

1. 0
2. 1
3. **the number of resources available**
4. none of the rest

Which of the following is NOT true regarding semaphore's queue-based implementation?

1. **It suffers from the busy waiting problem**
2. Semaphore has a waiting queue associated with the semaphore

3. When a process executes the *wait()* operation and finds that the semaphore value is not positive, it will suspend itself
4. A process that is suspended, waiting on the semaphore, should be restarted when some other process executes a signal() operation.

Which of the following is NOT true for Peterson's solution?

1. Mutual exclusion is preserved
2. The progress requirement is satisfied
3. The bounded-waiting requirement is met
4. **Peterson's solution works for synchronization among more than two processes**

Busy waiting refers to the phenomenon that while a process is in its critical section, any other process that tries to enter its critical section must loop continuously in the call to acquire the mutex lock.

1. **True**
2. False

The value of a counting semaphore can range only between 0 and 1.

1. True

**2. False**