1. Consider the following program. Can this program solve the mutual exclusion problem for two processes?

```
boolean blocked[2];
int turn;
void P(int id)
{
     while (true) {
          blocked[id] = true;
          while (turn != id) {
               while (blocked[1-id])
                       ;/* do nothing */
               turn = id;
          }
          /* critical section */
          blocked[id] = false;
          /* remainder */
     }
}
void main()
{
     blocked[0] = false;
     blocked[1] = false;
     turn = 0;
     parbegin (P(0), P(1));
}
```

2. Someone tries to solve the infinite buffer for the Producer/Consumer problem using binary semaphores as follows. The semaphore s is used to enforce mutual exclusion and the semaphore delay is used to force the consumer to wait if the buffer is empty. Do you think it works?

```
int  n=0;                                  /* number of data items in buffer */
binary_semaphore s  = 1, delay = 0;

void  producer ()
{
    while (true)  {                        /* repeat forever */
          produce();                       /* generate next data */
          semWaitB(s);                     /* enforce mutual exclusion */
          append();                        /* put data in buffer */
          n++;                             /* increment count of data in buffer */
          if  (n == 1) semSignalB(delay);   /* buffer was empty */
          semSignalB(s);
     }
 }

void  consumer()
{
    semWaitB(delay);                       /* if buffer is empty, wait */
    while (true)  {                        /* repeat forever */
          semWaitB(s);                     /* enforce mutual exclusion */
          take();                          /* take data out of buffer */
          n--;                             /* decrement count of data in buffer */
          semSignalB(s);
          consume();                       /* consume the data */
          if (n==0) semWaitB(delay);       /* if buffer is empty, wait */
    }
}
```

3. Refer to the solution to the readers/writers problem using semaphore with writers have priority. Assume that a reader is reading and no writer and reader are waiting for the time being.

a) What will be the values of the semaphores when a writer wants to write while the first reader is reading?

b) Continue with a), what will be the values of the semaphores when a second reader wants to read while the first reader is still reading?

c) Continue with b), what will be the values of the semaphores when a third reader wants to read while the first reader is still reading?

d) Continue with c), what will be the values of the semaphores when a second writer wants to write while the first reader is still reading?

e) Which one will resume first when the first reader finishes reading, assuming all the semaphores are ***strong semaphores***?

Self-test

1. ________ is when the sequence of instruction is guaranteed to execute as a group, or not execute at all, having no visible effect on system state.

A. Critical section
B. Mutual exclusion
C. Atomic operation
D. Starvation

2. The requirement that when one process is in a critical section that accesses shared resources, no other process may be in a critical section that accesses any of those shared resources is _________ .
A. starvation
B. deadlock
C. mutual exclusion
D. atomic operation

3. A _________ is an integer value used for signaling among processes.
A. semaphore
B. message
C. deadlock
D. critical section

4. The three operations that may be performed on a semaphore are initialize, _______ and __________ .

5. A semaphore that does not specify the order in which processes are removed from the queue is a _________ semaphore.
A. weak
B. general
C. strong
D. binary

6. Refer to the solution to the bounded-buffer producer/consumer problem using semaphore. Assume that the size of the buffer is 10. What is the value of each semaphore when a producer is inserting data into an empty buffer while no consumer is waiting?
A. s=1; n=0; e=10
B. s=0, n=0; e=9
C. s=0; n=1; e=10
D. s=0; n=1; e=9