CIS 415 Operating Systems: Worksheet 5 - Semaphore Basics

1. Indicate whether synchronization is needed in the following areas of OS resource management. Is semaphores the appropriate mechanism for each area?
   a. management of dedicated devices such as printer.
   b. movement of PCBs among blocked, ready, and running queues -- i.e. doing the linked list manipulations.
   c. managing one buffers for disk I/O; managing two buffers for disk I/O.
   d. coordinating two processes that are currently executing the Unix ls command on the same directory.
   e. coordinating read and write access to an array of integers shared between a parent and a child process.
   f. coordinating students who want to come to Ginnie's office hours when she only has four chairs.

2. Look at each sample code below and indicate whether it is possible for processes A and B to violate mutual exclusion. CS = critical section.

   a. semaphore mutex = 1;
      A:  P(mutex);  CS;  V(mutex);
      B:  P(mutex);  CS;  V(mutex);

   b. A: disable interrupts;  CS;  enable interrupts;
      B: disable interrupts;  CS;  enable interrupts;

   c. semaphore mutex = 1;
      A: disable interrupts;  CS;  enable interrupts;
      B:  P(mutex);  CS;  V(mutex);

   d. semaphore mutex = 1;  semaphore cutex = 1;
      A:  P(mutex);  CS;  V(mutex);
      B:  P(cutex);  CS;  V(cutex);

   e. semaphore mutex = 2;
      A:  P(mutex);  CS;  V(mutex);
      B:  P(mutex);  CS;  V(mutex);

   f. semaphore mutex = 1;
      A:  V(mutex);  CS;  P(mutex);
      B:  V(mutex);  CS;  P(mutex);

   /* test-and-set atomically reads flag and sets it to not-clear */
   g. flag = clear;
      A: while (test-and-set(flag) != clear) noop;  CS;  flag = clear;
      B: while (test-and-set(flag) != clear) noop;  CS;  flag = clear;

   h. flag = clear;
      A: while (flag != clear) noop; flag = not-clear;  CS; flag = clear;
      B: while (flag != clear) noop; flag = not-clear;  CS; flag = clear;

   i. flag = 1;
      A:  P(flag);  CS;  V(flag);
      B:  P(flag);  CS;  V(flag);