Quiz 2
Grace Livingston

Question 1:
A system has three task, Task A has highest priority, Task B medium, and Task C has lowest priority.
Current states:

Task C: Running.
Task A and B: Blocked
Then TaskB becomes unblocked.

What are the new task states now?
After that TaskA becomes unblocked, what are the new task states now?

Answer:
Task A: Running
Task B: Blocked
Task C: Blocked

Question 2: Fix the bugs in this code?

```
void Task1 ()
{
    PendSemaphore(A);
    PendSemaphore(B);
    a = b;
    PostSemaphore(B);
    PostSemaphore(A);
 }

void Task2 ()
 {
      PendSemaphore(A);
      PendSemaphore(B);
      b = a;
      PostSemaphore(B);
      PostSemaphore(A);
}
```

Answer:

Question 3: True or False

Tasks can pend on more than one event.

Answer:

True. Unlike Semaphores, the Event_pend() function accepts a list of Events to wait for.

An event can unblock more than one task.

Answer:

False. Only a single task may pend on any given Event at a time.

Holding a semaphore for too long may cause other tasks to miss deadlines

Answer:

True. If other tasks are pending on the held semaphore, they will not execute until the semaphore is released, regardless of any deadlines.

RTOS knows which semaphore protects which data?

Answer:

False. RTOS only knows that a Semaphore is being held or being waited on, but not the data protected by that Semaphore.

Using a Semaphore inside an Interrupt Service Routine is allowed.

Answer:

False. Semaphores disable interrupts to check the list of waiting tasks, and to acquire the semaphore, so an ISR waiting on a Semaphore would leave interrupts disabled, including the interrupt for scheduling tasks.

Question 4: Explain the difference between a hard and a soft real time system? Give an example for each.

Answer:

Hard Real Time: The deadlines on operations must be satisfied or damage to critical systems, injury, and/or loss of life will occur. Examples include aircraft flight control software (DO-179 systems), power generation systems, chemical plant controls, elevator controls.

Soft Real Time: Lost deadlines on operations may not be satisfied, but missing them will not result in critical failures. Examples include cell phone call management (except emergency calls), in-vehicle and in-flight entertainment systems, pager systems, home automation, household appliances such as microwaves, refrigerators, environmental controls.

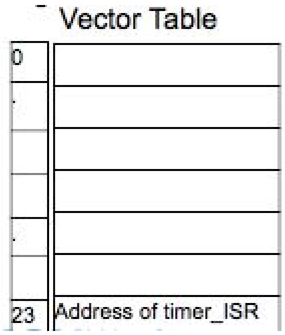
Question 5: For the last Timer Interrupt lab assignment, show the sequence of events that happen when an Timer overflow interrupt occurs?

(I am looking for what steps that happen from the time interrupt happens to the time ISR executes; things to consider vector table, usage of the stack etc)

main()

```
{
   timer_init();
   while(1)
   {
     ;
   }
}

timer_init()
{
..
}
timer_ISR()
{
.
}
```



Vector Table

0	
.	
.	
23	Address of timer_ISR

Answer:

1. The timer peripheral asserts the timer IRQ line.
2. The CPU core waits for interrupts to be enabled if they're disabled.
3. The CPU completes the current instruction.
4. The CPU saves the program counter stack.
5. The CPU then looks up the ISR address in the ISR vector table, and loads this into the program counter.
6. The CPU resumes executing instructions from the new program counter address.