SYSC 3310 Exercise Questions – part 1

"For these exercise questions, you may consult the reference manual and the lecture slides while solving the problems. In the exam, supporting information will be given with each question as needed"

1) If an input pin connected to an active low button is a source of interrupts, these should be configured as:
   a. level interrupts
   b. falling edge interrupts
   c. rising edge interrupts

2) To test whether an active low button (P2.1) is pressed, we would code
   a. if(~(P2->IN & (uint8_t)(1 << 1)))
   b. if(!(P2->IN & (uint8_t)(1 << 1)))
   c. if((P2->IN & (uint8_t)(1 << 1)))
   d. if((P2->IN & (uint8_t)(0 << 1)))

3) If we didn't have a flag register for Port interrupts, on a Port ISR, software would:
   a. potentially not be able to determine which pin caused the interrupt
   b. read the value of the pins (PxOUT), and know precisely which pin caused the interrupt
   c. read the value of the pins (PxIN), and know precisely which pin caused the interrupt

4) In our TimerA configured in up mode, assuming a hypothetical 500Hz clock, if we wanted to count precisely 200ms, we would set TA0CCR0 to the value:
   a. 100
   b. 99
   c. 199
   d. 20

5) If you were responsible for building a timer, triggered at 2Hz, what would be the minimum number of bits in the timer so it could count 1 minute?
   a. 8
   b. 9
   c. 7
   d. 10

6) Assuming our TimerA is initially configured to generate an interrupt every second, what does the following code in the Timer interrupt routine do?

```
TA0_N_IRQHandler()
{
        TA0CTL &= ~(uint16_t)(1 << 0);
        TA0CCR0 >>= 1;
}
```

   a. prevent any further timer interrupts
   b. stop the timer
   c. disable timer interrupts
   d. make the next interrupt occur in half the previous time

e. make the next interrupt occur in twice the previous time

7) What is wrong with the following code

```
void Port1_IRQHandler(void)
{
        if(P1IN & (uint8_t)(1 << 1))
        {
                P1IN &= ~(uint8_t)(1 << 1);
        }
}
```

a. we're writing to an invalid register
b. we're testing the wrong register
c. nothing
d. we're testing the wrong register and we're writing to an invalid register

8) Ignoring all configurations, and assuming we have no variables in our code, a system that has a timer interrupt which acts on 3 output LEDs, and no other interrupt, can be modeled by a state machine with at most:

a. 5 states
b. 3 states
c. 1 state
d. 2 states
e. 8 states
f. 9 states

9) If we modeled the following code as a simplified state machine (ignore configurations), it would have:

```
TA0_N_IRQHandler()
{
        TA0CTL &= ~(uint16_t)(1 << 0);
        P2->OUT ^= (uint8_t)(1 << 1);
}
PORT1_IRQHandler()
{
        if(P1->IFG & (uint8_t)(1 << 2))
        {
                P1->IFG &= ~(uint8_t)(1 << 2);
                if(P2->OUT & (uint8_t)(1 << 1))
                        P2->OUT &= ~(uint8_t)(1 << 6);
                else
                        P2->OUT |= (uint8_t)(1 << 6);
        }
}
```

a. 2 states
b. 5 states
c. 4 states
d. no states
e. 3 states