Review for Exam 2

1. C Programming
   - (a) Setting and clearing bits in registers
     - `PORTA = PORTA | 0x02;`
     - `PORTA = PORTA & ~0x0C;`
   - (b) Using pointers to access specific memory location or port.
     - `* (unsigned char *) 0x0400 = 0xaa;`
     - `#define PORTX (* (unsigned char *) 0x400)`
       `PORTX = 0xaa;`
2. Interrupts
   - (a) Interrupt Vectors (and reset vector)
     - How to set interrupt vectors in C
   - (b) How to enable interrupts (specific mask and general mask)
   - (c) What happens to stack when you receive an enabled interrupt
   - (d) What happens when you leave ISR with RTI instruction?
   - (e) What setup do you need to do before enabling interrupts?
   - (f) What do you need to do in interrupt service routine (clear source of interrupt, exit with RTI instruction)?
   - (g) How long (approximately) does it take to service an interrupt?
3. Timer/Counter Subsystem
   - (a) Enable Timer
   - (b) Timer Prescaler
     - How to set
     - How it affects frequency of timer clock
   - (c) Timer Overflow Interrupt
   - (d) Input Capture
   - (e) Output Compare
   - (f) How to enable interrupts in the timer subsystem
   - (g) How to clear flags in the timer subsystem

(h) Be able to look at registers and determine timer is set up
   - Which channels are being used
   - Which are being used for Input Capture, which for Output Compare
   - How to time differences from Timer count registers

4. Real Time Interrupt
   (a) How to enable
   (b) How to change rate
   (c) How to enable interrupt
   (d) How to clear flag

5. Pulse Width Modulation
   (a) How to get into 8-bit, left-aligned high-polarity mode
   (b) How to set PWM period (frequency)
      - Using Clock Mode 0
      - Using Clock Mode 1

   (c) How to set PWM duty cycle
   (d) How to enable PWM channel
   (e) Be able to look at PWM registers and determine PWM frequency and duty cycle