# Lab 3:

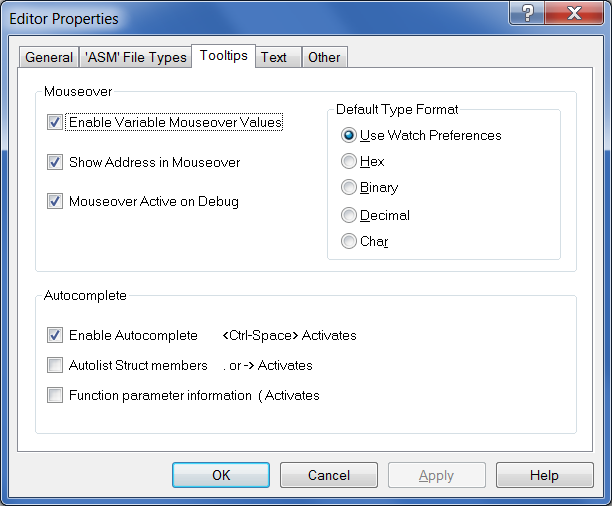
**Program Branching, Lookup Tables, Timers, Interrupts**

**Name:**

**Give brief answers to the following questions. You can edit this document and insert your answers after each question,**

**Due dates:**

**MW – Wed, Feb 21, beginning of class  
TTH – Tue, Feb 20, beginning of class**

1. (1 pt) There is a discrepancy between one of the labels in Figure 7-1 on page 55 of the datasheet and the register bit table for Register 7-1 on the same page. What is the discrepancy? Which is correct, the figure or the register bit table? (Hint: the correctness can be determined looking in the P16F877\_inc\_copy.txt file.) Moral of the story: data sheets can have errors.  
     
   **Ans.**
2. (1 pt) Which register is copied to W with the instruction movf PIR1, W if:  
     
   a) STATUS = 0x00  
   b) STATUS = 0x20  
   c) STATUS = 0x40  
   d) STATUS = 0x60.  
     
   **Ans.**
3. (1 pt) What is value of the W register (in hex) after the following instruction: movlw  D’381’ ? What warning number and message does the assembler generate when it assembles this instruction?  
     
   **Ans.**
4. (1 pt) Suppose PORTC = 0011 1100 and STATUS<C> = 1. What will be the contents of PORTC and STATUS<C> after the instruction rrf PORTC, F?   
   **Ans.**
5. (1 pt) In the lab03\_LookupTable.asm code, the following line occurs at the beginning of the code:  
     
   list p = 16f877  
     
   Why can we use the “=” symbol even though it is not one of the PIC’s instructions? (Hint: See Table 3.4 in the MPASM User Guide.)  
     
   **Ans.**
6. (1 pt) Build the lab03\_LookupTable.asm code. Under the Edit/Properties menu, make sure the following dialog is set as shown below..   
     
     
     
   a) Note that TableSize is set equal to 4. Mouseover TableSize. What message do you see?  
     
   **Ans.**  
     
     
   b) Open the File Registers dialog and notice that TableSize is not listed. In which register is 0x90 being stored? Why?  
     
   **Ans.**
7. (2 pts) In the Instruction Summary the instruction decf is listed with two arguments:  
     
   decf f, d  
     
   If the destination bit is omitted, the code will still compile. In the lab03\_LookupTable.asm code, insert the instructions below after the INIT label, build the code, and program the chip. Step the program until the two instructions are executed. Mouseover TableSize and compare the value of TableSize with the value of W in the status bar at the bottom of the window.  
     
   movlw 0x5  
     
   decf TableSize  
     
   a) What values are in TableSize and the W register (in hex)? What is the default destination, F or W?  
     
   **Ans.** TableSize =   
    W =   
    Default =   
     
   b) Now include the W destination bit in the decf instruction and repeat Part (a). What values are in TableSize and the W register?  
     
   **Ans.** TableSize =   
    W =
8. (1 pt) In the lab03\_LookupTable.asm code,   
     
   a) What would the W register contain (in hex) after the instruction movlw TRISA?   
     
   **Ans.**  
     
   b) What would the W register contain (in hex) after the instruction movf TRISA, W ?   
     
   **Ans.**
9. (1 pt) In the lab03\_LookupTable.asm code, insert the following dt directive after the INIT label and build the code:  
     
   dt 10, 20, 30  
     
   Open the Program Memory dialog. What code did the dt directive produce?  
     
   **Ans.**
10. (1 pt) What is the opcode for the instruction movlw 0xE5 ? Give your answer in binary and hex. For the binary, separate into nibbles for easier reading. (Hint: See the instruction description for movlw.)  
      
    **Ans.**
11. (1 pt) What is wrong with the syntax of the instruction incf PORTC, 2?  
      
    **Ans.**
12. (1 pt) What are the contents of the W register (in hex and binary) and STATUS<Z> after the instructions in the following code?   
      
    movlw 0xD8   
    xorlw 0xFF  
      
    **Ans.**   
    0xD8 =  W =   
     Z =
13. (1 pt) What are the contents of the Count and W registers (in binary) and STATUS<Z> after the instructions in the following instructions?  
      
    movlw 0xD8   
    movwf Count  
    comf Count, W  
      
    **Ans.**   
      
    Count =   
     W =  
     Z =
14. (1 pt) What are the contents of the W register (in binary) and STATUS<Z> after the following instructions execute?   
      
    movlw D’100’  
    andlw 0x88  
    **Ans.**
15. (1 pt) What instruction will be executed after the btfsc instruction in the following code? What are the values of W and STATUS<Z> after the following code executes?  
      
    movlw 0x81  
    addlw 0x7F  
    btfsc STATUS, Z  
    goto label\_A  
    goto label\_B  
    **Ans.**
16. (1 pt) Suppose Bank 2 is selected.   
      
    a) What are the contents of W (in hex) after the following instruction? (Hint: see the PIC16F877.inc on the resources folder.)  
      
    movlw EEDATA  
      
    b) Suppose Bank 3 is selected. What are the contents of W (in hex) for the same instruction?  
      
    **Ans.**
17. (1 pt) Suppose the oscillator frequency is 8 MHz, PR2 = 117, and T2CON = 0101 0011. What is the time (in milliseconds) between Timer2 interrupts?  
      
    **Ans.**
18. (1 pt) Suppose W = 5, and program memory address 0x008E executes an addwf PCL, F instruction. What program memory address (in hex) will execute next?  
      
    **Ans.**
19. (1 pt) In the lab03\_Timer2.asm project, put a breakpoint in the interrupt service routine. Notice that the program never reaches the breakpoint.  
      
    We have apparently not set up the Timer2 interrupt correctly in lab03\_Timer2.asm. Note that the INIT routine attempts to set T2CON = 0000 1101. Add T2CON to a watch window and step through the INIT routine to see if everything is working. Is T2CON being set to 0000 1101? Why not? What register is getting set to 0000 1101 instead of T2CON?  
      
    **Ans.**
20. (5 pts) Demonstrate the Watch window with the incorrect value of T2CON to the instructor or TA as described in the previous problem. Fix the error in the lab03\_Timer2.asm program by adding one instruction. Demonstrate a correctly working lab03\_Timer2.asm program and explain what you did.  
      
    **Student Name**:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Instructor or TA**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_

1. (10 pts) Create a new MPASM project called lab03\_LookupTable\_2.asm using the lab03\_LookupTable.asm program. Modify the program so that the LEDs continually repeat the following sequence of blinks: 7, 5, 3, 1, 0, 2, 4, 6. Explain the modified code to the instructor or TA and demonstrate this program.

**Instructor or TA**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_