ECE 4175 QUIZ 9-23-09 (1 of 2)
FIVE PROBLEMS
Open book, open notes

High	99
Upper quarter	87
Median	77
Lower quarter	56
Low	24

1. Write C code to turn on the Qwik&Low board's LED if bit 3 of **PORTB** (which has already been set up as an input) is set; otherwise turn off the LED. Be sure to leave all other output pins unaffected.

[10%]
```
if (PORTBbits.RB3)
{
   PORTDbits.RD4=1;
}
else
{
   PORTDbits.RD4=0;
}
```
Andrew Matteson, Tri Pho, and Terance Wu provided a simpler alternative (that I didn't know would work until I checked it):
```
PORTDbits.RD4=PORTBbits.RB3;
```

2. Consider Figure 3-2 on page 34 and the execution of the following two lines of code by the Qwik&Low board.

[20%]
```
NEW  = PORTE & 0b00000010;
NEW += PORTB & 0b00000100;
```

a. The resulting value of **NEW** depends upon code executed before these lines.
What is the function of this preceding code?
It powers up the two pullup resistors driven from RE2 and RE0.

b. Given the appropriate preceding code, what are the possible values of **NEW**, expressed as binary numbers?
```
0b00000000
0b00000010
0b00000110
0b00000100
```

3. For this problem, you are to write a Time function, called each time around the main loop. Time uses a static local unsigned char variable called **TIMECNT** to count the 100 loop times making up one second. It also uses another static local unsigned char variable called **SECONDS** that is to be incremented once every second (and rolled over from 59 back to zero). Each time **SECONDS** is incremented, use the **ASCII** function listed at the top of page 87 to break out the two digit ASCII-coded representation of **SECONDS**. Use the **Display** function listed at the bottom of page 86 to display the resulting value of time, left-justified on the LCD display, leaving whatever was last displayed in the other six character positions unchanged.

[40%]
```
void Time()
{
   static unsigned char TIMECNT=0;
   static unsigned char SECONDS=0;
   if (++TIMECNT>99)
   {
      TIMECNT=0;
      if (++SECONDS>59)
      {
         SECONDS=0;
         NUMBER=SECONDS;
         ASCII();
         LCDSTRING[0]=TENS;
         LCDSTRING[1]=ONES;
         Display();
      }
   }
}
```

ECE 4175 QUIZ 9-23-09 (2 of 2)

4. [20%] Consider the T1.c template program listed on pages 44-46. Also consider the Watchdog timer information of page 24. Coming out of reset, the SWDTEN bit is automatically cleared to zero, disabling the watchdog timer initially.

a. Describe the operation of T1.c if the line

```
WDTCONbits.SWDTEN = 1;
```

is moved from the end of the Initial function to the first line of the Initial function. Be sure to explain your answer clearly.

The watchdog timer will time out while the CPU is awake and executing

```
Delay(50000);     // Pause for half a second
```

Because the watchdog timer times out while the chip is awake, it resets the chip. The CPU repeats this sequence endlessly, never getting any farther into the program code.

b. Also describe the effect of this change on the blinking of the LED. Again, be sure to explain your answer clearly.

Since the CPU never gets beyond the execution of the Delay macro, and since the initialization occurring before the Delay macro clears RD4 to zero, the LED never blinks at all.

5. [10%] The TXascii macro definition listed on page 83 is:

```
#define TXascii(in)  TXREG = in; while(!TXSTAbits.TRMT)
```

However, one group left out the ! as they typed this line into their code:

```
#define TXascii(in)  TXREG = in; while(TXSTAbits.TRMT)
```

What would be the effect of this error upon what is displayed in QwikBug's console window when their code is executed? Explain your answer clearly.

This macro tests the TRMT bit of Figure 6-4 (on page 76) which remains cleared while a byte is being transferred. Consequently, after the first write to TXREG, there is no delay before a subsequent byte in a string is written to TXREG. TXREG is overwritten with all the bytes in a string before the 500 microseconds it takes for the first byte to be transferred out of the TSR register. Because of this, no more than the first byte of a string will appear on the QwikBug console. Furthermore, if the rest of the main loop code takes less than 500 microseconds before the chip is put to sleep by the CPU, not even this first byte will be completely transferred, as the UART's clock (derived from the Fosc) is stopped. In this case, the QwikBug console should display either a garbled single character or no character at all.