GE 420 Lab 2 Extra Prelab:　　　**A small Binary/ Hexadecimal number review (Due at the beginning of Lab 2)**

　　　　On the DSP that we are going to use, and on any processor, numbers are stored as groups (bytes) of 1's and 0's. If the number is displayed with just these 1's and 0's that representation is called binary. The C programs that you write can display these binary numbers in different formats: octal, decimal, hexadecimal. Most of the time your code will display this variable (or integer) in decimal format because that is what we are most used to. There are times though when you for example are programming a low level device driver. Your code will be interested in changing or finding the state of an individual bit in a variable. In that case it is much easier to manipulate the bits if the variables are displayed (or thought of) in hexadecimal format. This is why in some of the code that is given to you this semester you will see numbers written in hexadecimal format.

　　　　In the C/C++ programming languages, hexadecimal numbers are differentiated from decimal numbers by adding 0x to the beginning of the hexadecimal number.

　　　　Do these few exercises to jar your memory on hexadecimal numbers. Remember as you are doing these exercises that the goal of using hexadecimal numbers in your code is to manipulate/change the state (1 or 0) of individual bits in an integer variable.

1.　Write 0xE87C in binary format.

2.　What is the state of Bit 12 in the above number (Question 1)? Remember that the least significant right bit is bit 0.

3.　Write 1011100011011110 in hexadecimal format. Remember that each four bits of a binary number represents on digit (character) of the hexadecimal number.

4.　Write 0x86BD in binary format.

5.　What is the state of Bit 8 in the above number (Question 4)?

6.　Write 1110001110100010 in hexadecimal format.

Binary Math
| = Binary "OR"
& = Binary "AND"
~ = Binary "NOT"
^ = Exclusive OR "XOR" (We will not use this one here)

7. Solve　　　0x94AF | 0xA008 =　　　　　　　　　Make sure to show your work doing it by hand.

8. Solve　　　0xA5C3 & 0x 1824 =　　　　　　　　Make sure to show your work doing it by hand.

9. Solve　　　~0xB346 =

A little hexadecimal algebra:

10.　Solve for "a" (here "a" only has one of its bits set high (1) all other bits are low (0))

0x5F13 | a = 0x5F53　　　　　In other words, what do you "OR" with this number to keep all of its bits the same except for changing "bit 6" from 0 to 1.

11.　Solve for "a" (here "a" only has one of its bits set low (0) all other bits are high (1)

0x324C & a = 0x304c　　　　　In other words, what do you "AND" with this number to keep all of its bits the same except for changing "bit 9" from 1 to 0.