EECE 2160 - Embedded Design: Enabling Robotics

Fall 2016 – Homework #1
Due: October 9 (end of day)

Instructions:

- Please type your solutions into a document and convert it into a PDF file. Your solution document should contain your name, student ID, the course name, and homework number. Submit (on Blackboard) a single PDF file with all your work including **commented** C programs and the output screen captures.
- You may discuss concepts with your classmates. This fosters group learning and improves the class' progress as a whole. However, make sure to submit your own independent and individual solutions.

---

1. (10 points) Convert the following numbers to 8-bit **unsigned** binary. The subscript indicates the base (show your work).
   a) 53_{10}
   b) FA_{16}

2. (20 points) Using 8-bits for the following 2's complement numbers, convert each to the indicated base. The subscript indicates the base (show your work).
   a. $19_{10} \rightarrow (\ ?\ )_2$
   b. $-13_{10} \rightarrow (\ ?\ )_2$
   c. $-23_{10} \rightarrow (\ ?\ )_{16}$
   d. $ED_{16} \rightarrow (\ ?\ )_{10}$

3. (10 points) Perform the following logical operations. Express your answers in hexadecimal notation (show your work).
   a. 0xABCD **OR** 0x9876
   b. 0xFEED **AND** (**NOT**(0xBEEF))

4. (10 points) Decode the following ASCII strings that are shown in binary ($B_7 \ldots B_0$):
   a. 01000010 01100101 01110011 01110100 00100000 01101111 01100110
   b. 01001100 01110101 01100011 01101011 00100001

5. (20 points) Express -14 as a 2's complement integer using 5 bits. Repeat the same using 6 and 7 bits. What does this tell you with respect to the property of sign extension in 2's complement representation?

6. (30 points) Write a C (or C++) program that prompts the user for two positive integers. Provide individual program functions, called from `main()`, that compute the following:
   a.) Compute their sum

b.) Compute their difference

In these problems, you can only use bit-wise logical operations (AND, OR, XOR, NOT). You cannot use +, -, *, or / operators (standard math syntax in C).

From the `main()` function, print out the results for functions a) and b) in **signed integer**, and also **two's complement binary format**. There is no specifier for printing in binary format. You can instead make a simple function that takes in as an argument a signed integer, and prints out the binary (1's and 0's) equivalent of the integer, e.g. `void printBinary(int x)`. Use the `sizeof()` function to determine the correct number of bits your output should have.

Include your **commented** code and sample test run of your program in your submission.