INTRODUCTION TO COMPUTER ENGINEERING
ASSIGNMENT 1: DATA REPRESENTATION DUE DATE: FRIDAY FEBUARY 12TH, 2016 AT 5:00PM

- Please submit this assignment by 5:00 pm on the due date via *MyCourses*. Your assignment must be packaged as follows: one compressed file (.zip) containing **exactly two files**: *If* this does not work. Email the material to the course email: mcem352@yahoo.ca
- Answers to Q1-Q3 as well as discussions and the output of your "C" program for Q4 (obtained either through screen capture or output re-direct) in a Microsoft Word (doc/docx) or portable document format (pdf) file
- The source code for Q4 in a text document (called `num_2_bin.c`).

A penalty of 10% will be applied if you do not follow these guidelines.

This assignment will be marked out of 100 points. **Late submissions will NOT be accepted.**

QUESTION 1 (16 POINTS)

Convert the following unsigned numbers into their numerical equivalents in the indicated bases. Be sure to use the correct number of significant figures for each case and show how the correct number of significant figures was obtained. **Show all your work**. The final answer on its own is not worth much – how you get there is what matters.

(a) 0.89_{10}
- (i) base 2
- (ii) base 16
- (iii) base 5
- (iv) base 8

(b) 101111011101_2
- (i) base 10
- (ii) base 16
- (iii) base 3
- (iv) base 4

(c) $BA2F.A3_{16}$
- (i) base 2
- (ii) base 10
- (iii) base 5
- (iv) base 8

(d) 17.341_8
- (i) base 10
- (ii) base 4
- (iii) base 3
- (iv) base 16

QUESTION 2 (20 POINTS)

(a) The following 32-bit binary number is the IEEE-754 single precision representation of a real number X. Compute X. Show all your work.

0010 0010 0001 1110 1100 1110 0000 0000

(b) The following 8-digit hexadecimal number is the IEEE-754 single precision representation of a real number Y. Compute Y. Show all your work.

0xAC0396ED

(c) Compute the binary representation of the number $3.4219087.10^{12}$. Show all your work; justify in particular any choice in number of significant digits that you make. Does representation entail representational error? If so, how much?

QUESTION 3 (14 POINTS)

Perform the following calculations assuming that all numbers are stored in 16-bit registers as 2's complement binary numbers with no overflow provision. Convert each of the numbers to their binary equivalent, and do all your calculations in binary to produce a binary result. Convert the result to hexadecimal and decimal (base-10) formats. If a computation produces incorrect results, explain why. Show all of your work.

(a) $E421_{16} - 1ACE_{16}$
(b) $4530_{16} + A123_{16}$

QUESTION 4 (50 POINTS)

You will write a C program to convert a base 10 real number into a binary equivalent and the hexadecimal equivalent. The number (base 10) is typed in and the code will output the binary equivalent with correct number of bits.

Examples of running the code below calls are given below:

```
> 12.25
  1100.01 (bin)
  C.4   (hex)

> 56.625
  111000.101
  38.A (hex)
```

Your program should

- produce an error message if input arguments are out of range (negative numbers, infinity etc…)
- produce an error if no arguments are given on the command line;
- not use any use the computer's built-in IEEE 754 conversion capabilities rather the computation should be done by your code (no special libraries).

Your program does not need to handle floating point input that are not number (`NaN` or $\pm\infty$).

Describe any limitations that may exist for your program. Full marks will only be given to programs capable of handling the complete range of inputs, however, a proper explanation of any limitations your program may have will minimize any deductions. Only properly commented code will receive full marks.

To test your code, select appropriate test cases to show that it works throughout the possible input range. Choosing a test set is a non-trivial problem and should be done with care, concentrating on number boundaries (maximum, minimum) and other code specific areas which may result in problems for your code. Include a printout of the results for all of your tests.

The "C" source code should contain a header which has the following information:

- Full name
- Student number
- Course number
- Date

Put all your functions and main routine into a single C language file, and only use standard C libraries (e.g. `stdio.h, stdlib.h`).