CS 429 Homework 4

Name: ______________________ Section #: ______________________

Instructions: As usual, you may collaborate with your classmates and ask for assistance from the TA. But don't copy anyone else's answer. Each problem is worth the same number of points (more or less). Submit your code in a hw4.c file and pipe the output to a separate file that you also submit. Don't forget to include your name in your files.

The questions from this homework are related to (or identical to) questions from a previous version of Exam 1. Write C programs to compute and print the answers. For each also output a label: e.g., "Problem 1a:" For some of them, you may find useful the `show_bytes` program from the slides. You should have a single `main` program that does all of the questions below, but you can have as many subsidiary functions as you need. For many of these you can just use `printf` with an appropriate format string to do most of the work.

1. For each of the following, write a fragment of code that will print out the answer. For those that say "Compute" use C to do the computation. For those that say "Show" you can just print the answer using an appropriately formatted `printf` statement, or your binary print routine. For those where the answer is binary, write a subsidiary function to produce a binary string, save it to a string variable, and print that.

   (a) Compute -123_{10} in 32-bit two's complement hex.

   (b) Compute `0x01234567 & 0x0F0F0F0F` (hex)

   (c) Show (in binary) `10.1111` rounded to even (at fourths position)

   (d) Compute `0x01234567 ^ 0x0F0F0F0F` (hex)

   (e) Show (in binary) TMin for a 9-bit two's complement system

   (f) Show (in decimal) TMax for a 9-bit two's complement system

   (g) Show (in decimal) UMax for a 7-bit system

   (h) Compute `0x01234567 | 0x0F0F0F0F` (hex)

   (i) Compute `~0x01234567` (hex)

   (j) Show decimal 7.3 to binary (indicate repeating bits); you don't have to compute this in C, just print the answer.

2. What decimal value is represented by the following bit string when interpreted as a single precision floating point value: `0xD1000000`. (Implement with a subsidiary function that takes a hex value and returns the corresponding decimal float value. You can use a union to make this easy.)

3. Implement the following functionality in C, supplying an expression to swap the two bytes in the unsigned short. Print the value in hex. Test it on `x == 0x1234`; it should return `0x3412`.

```
unsigned short x;
unsigned short swapbytes;
swapbytes = ______________________________________ ;
```

4. Write a function that takes a character string value and prints out the bytes stored in memory of its representation (including the terminating 0). Test it on a string containing your full name.