TCSS333
C for System Programming
Programming Assignment 8
Binary Representation of 32 bit Floating Point Values

DUE: See Canvas Home Work 8 link.

Create a file called Hw8.c based on the following:

Write a program that will read a series of lines from an input file named hw8in.txt. Each line will contain 3 values:

A whole number
A numerator
A denominator

Collectively, these 3 values represent a floating point number. For example, a possible line of input might be:

3 3 4

This will become: 3.0 + 3.0 / 4.0 or 3.75

Your program will store this value in a float (Do NOT use double) and will display to the console, both its float and floating point binary values. Again using the above float from the input file, the output would be:

```
Floating point format of    3.750000000000000:
 0 10000000 11100000000000000000000
```

Your program must create the same output as shown in the sample run that is included below. This includes breaking the 32 bits into 3 separate fields (sign, exponent, and fraction). To facilitate grading, your program must follow this format.

NOTE: Due to the input format of whole, numerator, denominator, you will need to use 0 1 for the last 2 digits if there is no fractional portion to the entire value to be evaluated. E.g. 5 0 1 will evaluate to 5.0.

Design your own input file (Hw8in.txt) and include as many variations of input as needed to fully test your program.

Write the entire program in one .c file (Hw8.c). Organize your solution into a sensible set of functions. As always, turning in code you may find on the internet is not acceptable. The goal of the assignment is for you to practice manipulating bits.

The float data type of C usually requires 32 bits (or 4 bytes). To ensure this is true for your machine, try this: printf("float size in bytes: %d", sizeof(float)); If the value printed out is not 4, please talk to me about how to do the assignment.

If you missed class, there is an explanation of the floating point format at:

http://en.wikipedia.org/wiki/Single-precision_floating-point_format

The program can be completed in less than 100 lines of code.

Be sure to test both negative and positive numbers.

A sample run appears on the next page.

As with HW6 and HW7, your program should compile at the command prompt. Save your program as Hw8.c and upload only this single source code file to the link for Home work 8 (Do NOT upload your testing input file). Also, follow proper documentation rules.

Sample Run - - - - >

Sample input file:

```
3 3 4
13 0 1
6 1 8
-5 1 2
-1 9 32
-6 7 9
0 0 1
15 0 1
```

Output displayed at the console:

```
Floating point format of    3.750000000000000:
       0 10000000 11100000000000000000000
Floating point format of   13.000000000000000:
       0 10000010 10100000000000000000000
Floating point format of    6.125000000000000:
       0 10000001 10001000000000000000000
Floating point format of   -5.500000000000000:
       1 10000001 01100000000000000000000
Floating point format of   -1.281250000000000:
       1 01111111 01001000000000000000000
Floating point format of   -6.777777671813965:
       1 10000001 10110001110001110001110
Floating point format of    0.000000000000000:
       0 00000000 00000000000000000000000
Floating point format of   15.000000000000000:
       0 10000010 11100000000000000000000
```