Due: Monday, June 27th.

Filename: converter.cpp, authors.csv

Format of authors.csv: 

author1_email, author1_last_name, author1_first_name
author2_email, author2_last_name, author2_first_name

For example:
simpson@ucdavis.edu, Simpson, Homer
potter@ucdavis.edu, Potter, Harry

Written (45 points): Due at 4pm in the homework box in 2131 Kemper
Chapter 1 Exercises: 2 b,e; 3 a,b,c,e; 4 b,e,i; 5 c,d; 6 b,d,e; 7 a,c; 8 b; 9 a vi), b vi); 10 c i), c iii), d i), d iii); 11 a,b; 12.
Chapter 2 Exercises: 2 a,b; 3; 4; 5; 6; 8; 10.

Programming (~2 hours, 35 points) : Due at 11:59pm in p1 in cs50 account using handin. Filename: converter.cpp
Write a program that: 1) converts an unsigned integer of a specified base to an unsigned integer of another specified base (45 minutes); 2) converts a signed decimal integer to its 32-bit signed magnitude and 32-bit 2’s complement hexadecimal representations (30 minutes), and 3) converts a float into its base 2 scientific notation (30 minutes). You may use class(es), if you wish, but they are not required. Please place the name(s) of the author(s) in a comment in the first line of the file. You will find a copy of my executable in ~ssdavis/50/p1.

Additional Specifications

1. For parts 1 and 2
   1.1. You must read the input numbers into character arrays.
   1.2. All conversions must be done explicitly by your program. You may not call atoi, atof, sprintf(), hex, %x, cast, or any other built-in converting functions.
2. You may assume that all inputs will be valid. There is no need for error checking.
3. The only header file you may #include is iostream.
4. The format of your output must match mine exactly. We use diff to compare your output to mine to determine your score.
5. Unsigned integer numbers.
   5.1. You may assume that the unsigned integers will fit into a 64-bit unsigned long long.
   5.2. The possible bases are 2-16 inclusive. For bases 11 to 16, you should use the letters A-F for digits valued greater than 9.
   5.3. Hint: Use a base 10 representation as an intermediate form for all combination of bases.
   6.1. The input format will be [-][0-9]*, with no leading zeroes.
7. Float.
   7.1. IEEE 32 bits (float): 31 sign bit, 23-30 exponent (with 127 offset), 0-22 mantissa (with implicit leading 1).
   7.2. Your mantissa should have the implicit leading 1 prepended.
   7.3. Hint: To make use of the bitwise operators you will need to convert your float to an integer. I used “unsigned temp = *((unsigned*) &floatValue);”

[ssdavis@lect1 p1]$ converter.out
Menu
0. Done.
1. Convert unsigned integer.
2. Convert signed integer.
3. Convert float.
Your choice: 1
Number: 13
Source base: 5
Destination base: 10
114

Menu
0. Done.
1. Convert unsigned integer.
Menu
0. Done.
1. Convert unsigned integer.
2. Convert signed integer.
3. Convert float.
Your choice: 1
Number: FF
Source base: 16
Destination base: 10
255
Two's complement: 00000001
Menu
0. Done.
1. Convert unsigned integer.
2. Convert signed integer.
3. Convert float.
Your choice: 2
Signed number: -57893
Signed magnitude: 8000E225
Two's complement: FFFF1DDB
Menu
0. Done.
1. Convert unsigned integer.
2. Convert signed integer.
3. Convert float.
Your choice: 3
Float: 13.5
1.10110000000000000000000 E11
Menu
0. Done.
1. Convert unsigned integer.
2. Convert signed integer.
3. Convert float.
Your choice: 2
Signed number: -1
Signed magnitude: 80000001
Two's complement: FFFFFFFF
Menu
0. Done.
1. Convert unsigned integer.
2. Convert signed integer.
3. Convert float.
Your choice: 3
Float: -40.1
-1.01000000110011001100110 E101
Menu
0. Done.
1. Convert unsigned integer.
2. Convert signed integer.
3. Convert float.
Your choice: 0
Bye
[ssdavis@lect1 p1]$