CST 8233 – F17 - Assignment #2

Floating Point Spy

Due Date: 23:59 12 November 2017 **Earnings:** 8% of your final grade

NOTE: Plan to finish a few days early to avoid last minute hardware/software issues for which there is no allowance.

NOTE: The code in this assignment must be your own work. It must not be code taken from another student or written for you by someone else, even if you give a reference to the person you got it from (attribution); if it is not entirely your own work it will be treated as plagiarism and given a fail mark, or less.

Purpose: A utility application to show a real float number in its normalized hexadecimal and binary representations together with its exponent and normalized mantissa in memory, to automate what is done manually in class.

Algorithm

Write a utility program in C (**not C++**) named *ass2* that will enable you to view the internal fields used to represent float numbers.

While the user wishes to continue, the user is asked to enter the next float number. The binary and hexadecimal representations of that number are then presented together with the exponent and normalized mantissa (output as a float with 7 decimal places). You may find bitwise operations in C useful for this.

Example output is shown at the end. Yours must look the same. If dynamic memory allocation is used, it must be released when the application terminates.

See the Marking Sheet for how you can lose marks, but you will lose at least 60% if: it fails to build in Visual Studio 2015, it crashes in normal operation, and it doesn't produce the example output.

Set up a Win32 **console** project in Visual Studio 2015 named *ass2* with the single source code file ass2.c. Write the code to implement the application as described above.

What to Submit :

Use the Submission on Blackboard to submit this assignment as a zip (not RAR not 7-Zip not 9 Zip) file containing only the source code ass2.c file. The name of the zipped folder **must** contain your name as a prefix so that I can identify it, for example using my name the file would be tyleraAss2CST8233.zip. It is also vital that you include the Cover Information (as specified in the Submission Standard) as a file header in your source file so the file can be identified as yours. Use comment lines in the file to include the header.

Before you submit the code,

- *check that it builds and executes in Visual Studio 2015 as you expect - if it doesn't build for me, for whatever reason, you get a deduction of at least 60%.*
- *make sure you have submitted the correct file – if I cannot build it because the file is wrong or missing from the zip, even if it's an honest mistake, you get 0.*

There's a late penalty of 25% per day. Don't send me file(s) as an email attachment – it will get 0.

Example output - yours must look identical:

```
1 = convert a float
2 - quit
1

Please enter the number to convert: 12.5

float number is         1.250000e+01
Binary:                 0100 0001 0100 1000 0000 0000 0000 0000
Big-endian Hex:         41 48 00 00
```

```
Little-endian Hex:      00 00 48 41
exponent = 130
normalised mantissa (including the hidden bit) = 1.5625000


1 = convert a float
2 - quit
1

Please enter the number to convert: -1.0

float number is         -1.000000e+00
Binary:                 1011 1111 1000 0000 0000 0000 0000 0000
Big-endian Hex:         BF 80 00 00
Little-endian Hex:      00 00 80 BF
exponent = 127
normalised mantissa (including the hidden bit)  = 1.0000000


1 = convert a float
2 - quit
1

Please enter the number to convert: 3.141593

float number is         3.141593e+00
Binary:                 0100 0000 0100 1001 0000 1111 1101 1100
Big-endian Hex:         40 49 0F DC
Little-endian Hex:      DC 0F 49 40
exponent = 128
normalised mantissa (including the hidden bit) = 1.5707965


1 = convert a float
2 - quit
1

Please enter the number to convert: 1.0e37

float number is         1.000000e+37
Binary:                 0111 1100 1111 0000 1011 1101 1100 0010
Big-endian Hex:         7C F0 BD C2
Little-endian Hex:      C2 BD F0 7C
exponent = 249
normalised mantissa (including the hidden bit) = 1.8807909


1 = convert a float
2 - quit
1

Please enter the number to convert: 1.0e-37

float number is         1.000000e-37
Binary:                 0000 0010 0000 1000 0001 1100 1110 1010
Big-endian Hex:         02 08 1C EA
Little-endian Hex:      EA 1C 08 02
exponent = 4
normalised mantissa (including the hidden bit) = 1.0633824


1 = convert a float
2 - quit
```