Lab8–Floating Point Practice
CPSC 2311- Spring 2021

Introduction

The goal of this lab is to introduce/practice Floating Point conversion. We will also cover this concept in class this week along with working examples.

Due:

Monday, March 29, 2021, midnight.

Lab Instructions

While I know this would be much easier to do by hand. I have found grading to be easier when your answers are typed. Therefore**, must** type your answer in RED on this document and submit the document through canvas as a **PDF**. Please read the entire document. Points will be deducted if you do not follow directions.

Part 1:
Watch the following videos pertaining to Floating Point conversion from decimal to binary and binary to decimal.

https://www.youtube.com/watch?v=tx-M_rqhuUA
https://www.youtube.com/watch?v=4DfXdJdaNYs

Part 2:

Following the instructions in the above video. Convert the following floating-point number to binary.

64.48

First, we need to extract from the number the two binary values for the whole-number portion (left of the decimal separator) and the fraction portion (right of the decimal separator):

64 =	32 * 2 + 0	0.48 * 2 =	0.96	
32 =	16 * 2 + 0	0.96 * 2 =	1.92	
16 =	8 * 2 + 0	0.92 * 2 =	1.84	
8 =	4 * 2 + 0	0.84 * 2 =	1.68	
4 =	2 * 2 + 0	0.68 * 2 =	1.36	
2 =	1 * 2 + 0	0.36 * 2 =	0.72	|
1 =	0 * 2 + 1	0.72 * 2 =	1.44	|
		0.44 * 2 =	0.88	|
		0.88 * 2 =	1.76	|
		0.76 * 2 =	1.52	| repeating
		0.52 * 2 =	1.04	|
		0.04 * 2 =	0.08	|
		0.08 * 2 =	0.16	|
		0.16 * 2 =	0.36	|

$64.48 = 1000000.01111\underline{010111000} = 1.00000001111\underline{010111000} * 2^6$

Next, we need to calculate the sign bit, the exponent bits, and write in the bits of the mantissa.

S = 0_2 [the number is positive]	1 bit
Exp = 6 + 127 = 133 = 10000101_2	8 bits
M = $00000001111\underline{010111000}$	23 bits

Thus, the floating-point number in binary is
0 10000101 00000001111010111000011

= 0100 0010 1000 0000 1111 0101 1100 0011

Show your work. Also, explain what you are doing each step of the way. Your explanation does not have to be a long explanation. Only enough to let your TA know you understand what you are doing. If you do not show and explain your work, you will receive a 0 for the question.

Now convert the binary back to decimal, showing and explaining each step of the process. Again, your explanation does not have to be a long explanation. Only enough to let your TA know you understand what you are doing. If you do not show and explain your work, you will receive a 0 for the question.

Converting backwards is easy, as we already know the steps to take. The binary representation is 0 10000101 00000001111010111000011

Which means the values of s, exp, and m are:

$S = 0_2 = 0$ (positive)	(the first bit)
$Exp = (10000101)_2 = 133 - 127 = 6$	(the next 8 bits)
$M = (0000010111\underline{1010111000)}_2$	(the last 23 bits)

Now we can convert the mantissa back into decimal to find the value of the number:

$M = 2^{-8} + 2^{-9} + 2^{-10} + 2^{-11} + 2^{-13} + 2^{-15} + 2^{-16} + 2^{-17} + 2^{-22} + 2^{-23} = 0.007500052$

Now the decimal value is $(-1)^s * (1 + m) * 2^{exp} = 64.48000336\ldots \sim= 64.48$

Part 3:

Following the instructions in the above video. Convert the following floating-point numbers to binary.

-195.56

First, we need to extract from the number the two binary values for the whole-number portion (left of the decimal separator) and the fraction portion (right of the decimal separator):

195 =	97 * 2	+ 1		0.56 * 2 =	1.12
97 =	48 * 2	+ 1		0.12 * 2 =	0.24
48 =	24 * 2	+ 0		0.24 * 2 =	0.48
24 =	12 * 2	+ 0		0.48 * 2 =	0.96
12 =	6 * 2	+ 0		0.96 * 2 =	1.92
6 =	3 * 2	+ 0		0.92 * 2 =	1.84
3 =	1 * 2	+ 1		0.84 * 2 =	1.68
1 =	0 * 2	+ 1		0.68 * 2 =	1.36
				0.36 * 2 =	0.72
				0.72 * 2 =	1.44
				0.44 * 2 =	0.88
				0.88 * 2 =	1.76
				0.76 * 2 =	1.52
				0.52 * 2 =	1.04
				0.04 * 2 =	0.04

0.08 * 2 = 0.08 ... and so on

195.56 = 11000011.1000111101011100 = 1.1000011100011110101110 0 * 2^7

Next, we need to calculate the sign bit, the exponent bits, and write in the bits of the mantissa.

S = 1_2 [the number is negative]	1 bit
Exp = 7 + 127 = 134 = 10000110_2	8 bits
M = 10000111000111101011100	23 bits

Thus, the floating-point number in binary is
1 10000110 10000111000111101011100

= 1100 0011 0100 0011 1000 1111 0101 1100

Show your work. Also, explain what you are doing each step of the way. Your explanation does not have to be a long explanation. Only enough to let your TA know you understand what you are doing. If you do not show and explain your work, you will receive a 0 for the question.

Now convert the binary back to decimal, showing and explaining each step of the process. Again, your explanation does not have to be a long explanation. Only enough to let your TA know you understand what you are doing. If you do not show and explain your work, you will receive a 0 for the question.

Converting backwards is easy, as we already know the steps to take. The binary representation is 1 10000110 10000111000111101011100

Which means the values of s, exp, and m are:

S = 1_2 = 1 (negative)	(the first bit)
Exp = $(10000110)_2$ = 134 - 127 = 6	(the next 8 bits)
M = $(10000111000111101011100)_2$	(the last 23 bits)

Now we can convert the mantissa back into decimal to find the value of the number:

M = $2^{-1} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-12} + 2^{-13} + 2^{-14} + 2^{-15} + 2^{-17} + 2^{-19} + 2^{-20} + 2^{-21}$ = 0.5278124809

Now the decimal value is $(-1)^s * (1 + m) * 2^{exp}$ = -195.5599976 ~= -195.56

The following link is a nifty tool you can use to check your work. You should understand that sometime online tools like this one will round which could change the last one or two bits on the tool. So, if your answer has a different bit on the end that is perfectly fine. I am not saying this will be the case only letting you know this could happen.

https://evanw.github.io/float-toy/

Submission:

You should submit your document to Canvas. Please make sure your answers are in RED. If you do not, a substantial number of points will be deducted.