Name: Pyung Lee
Email ID: pkl4fr
Date: 2/10/20
File Name: floatingpoint.pdf

Your magic (32 bit) floating point number is 8.8125
This is the number that needs to be converted to (little endian) binary, and expressed in hexadecimal.

Your other magic floating point number is, in hex, 0x00c01ec2
This is the number that needs to be converted to a (32 bit) floating point number.
Note that the hexadecimal printed above is in little-endian format!

1) 8.125

*Converting to Binary
8/2 = 4R0 -> 4/2 = 2R0 -> 2/2 = 1R0 -> 1/2 = 0R1 → 1000
.125 = .50 + .25 + 0.0625 → 1101
Total Binary = 1000.1101

*Scientific Notation
1.**0001101** x 2^3

*Converting Exponent to Binary
3 + 127 = 130
130/2 = 65R0 -> 65/2 = 32R1 -> 32/2 = 16R0 -> 16/2 = 8R0 -> 8/2 = 4R0 -> 4/2 = 2R0 -> 2/2 = 1R0 -> 1/2 = 0R1 → **10000010**

*Putting It All Together
0|100 0001 0|000 1101 0000 0000 0000 0000

*Converting to Hexadecimal (Big Endian)
0x410d0000

*Converting to Little Endian
0x00000d41

2) 0x00c01ec2

*Converting to Big Endian
0xc21ec000

*Converting to Binary
1|100 0010 0|001 1110 1100 0000 0000 0000

*Finding the exponent
1000 0100 = 128 + 4 = 132 – 127 = **5**

*Using Mantissa to Find Value
001 1110 1100 0000 0000 0000 =
(1/8)+(1/16)+(1/32)+(1/64)+(1/256)+(1/512)= .2402 + 1 = 1.2402

1.2402 * 2^(5) = **39.6864**

*Change Sign
-39.6864