32 Bit to Hexadecimal Number

32 Bit # = 4.3984375

1) Sign: Positive so represented by 0
2) Convert the whole number and the decimal to binary
   - Whole # conversion: 4 in binary is expressed as 0100
   - Decimal conversion: 0110 0110

.3984375	.796875	0
.796875	1.59375	1
.59375	.1875	1
.1875	.375	0
.375	.75	0
.75	1.5	1
.5	1	1
0		0

   - Whole number = 0100 . 0110 0110
3) Normalize the mantissa: 1.0001100110 E2
4) Calculate the exponent: 127 + 2(power of the exponent) = 129 = 1000 0001 (binary)
5) Put it all in 32 bit form

   0100 0000 1000 1100 1100 0000 0000 0000

   In Big Endian notation this evaluates to: 408CC000
   In Little Endian notation this evaluates to: 00C08C40

   Solution in Hex = 0x00C08C40

Hexadecimal Number to 32 bit Float

Hexadecimal # = 0x00009fc1
Hex in Big Endian = c19f0000

1) Convert the Hexadecimal number to binary

c	1	9	f	0	0	0	0
1000	**0001**	**1001**	**1111**	**0000**	**0000**	**0000**	**0000**

2) Group the binary into three parts
   - Sign = first bit
     - 1 evaluates to negative
   - Exponent
     - 10000011 evaluates to 131
     - We are trying to determine how much we need to multiply our mantissa by to get our floating point value. In the opposite was as before we will subtract 127 from our number
     - 2^(131-127) = 2^(4) = 16
3) Evaluate the mantissa:
   - 1.001111100000000….. in binary
   - Convert it from binary the same way you would convert a normal binary to decimal
   - SUM(2^(0) + 2^(-3) + 2^(-4) + 2^(-5) + 2^(-6) + 2^(-7)) = 1.2421875
   - The trailing zeros are irrelevant
4) Multiply the sign, the exponent, and the mantissa
5) -19.875

Floating Point Value is -19.875