For userid 'gp3dz':

Your magic (32 bit) floating point number is -19.125
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, 0x00809f40
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!

X = -19.125
Int Y = int(X)
If (Y < 0) {
Y = Y * -1;
}
Y = 19

When its (-) negative the first digit of the 32 bit is (1).

From using power of 2

2^0 = 1
2^1 = 2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32

19 - 2^4 = 3 = 1
3 < 2^3 = = 0
3 < 2^2 = = 0
3 - 2^1 = 1 = 1
1 - 2^0 = 0 = 1

So Y = 19 = 10011

To calculate decimals to binary

Mantissa = K

```
If  ( X < 0 ) {
        X = X * -1;
}
```

K = X - Y
K = 0.125

Binary calculation.

0.125 * 2 = 0.25 = 0
0.25 * 2 = 0.50 = 0
0.50 * 2 = 1.00 = 1
0.00 * 2 = 0.00 = 0
…
…
…

Combine
10011.0010…...

To scientific notation
1.00110010…… * 2^4

To calculate exponent bits
127 + 4 = 131
2^0 = 1
2^1 =2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32
2^6 = 64
2^7 = 128
2^8 = 256

131 - 2^7 = 3
3 < 2^6
3 < 2^5
3 < 2^4
3 < 2^3
3 < 2^2
3 - 2^1 = 1
3 - 2^0 = 0

So… 131 binary is 10000011

Sign bit is 1
Exponent bit is 10000011
Decimal bit is 00110010000000000000000

Combine sign bit, exponent bit, and decimal bit…
1 10000011 00110010000000000000000

1100 0001 1001 1001 0000 0000 0000 0000

Converting to hexadecimal...

1100 = 2^3 + 2^2 = 8 + 4 = 12 = C
0001 = 2^0 = 1
1001 = 2^3 + 2^1 = 8 + 1 = 9
1001 = 2^3 + 2^1 = 8 + 1 = 9
0000 = 0
0000 = 0
0000 = 0
0000 = 0

Hexadecimal (0x) is
c1990000

0xc1990000

Converting hexadecimal to decimals
0x00809f40
0x = hexadecimal
0 = 0000
0 = 0000
8 = 2^3 = 8 = 1000
0 = 0000
9 = 2^3 + 2^0 = 1001
f = (f is 6th alphabet. So 9 + 6 = 15) = 1111
4 = 2^2 = 0100
0 = 0000

0000 0000 1000 0000 1001 1111 0100 0000