

1. 32-bit floating point number: -2.453125 → convert to little endian binary, expressed in hexadecimal:

- First, the sign is negative so we use a 1
- Exponent = 1 because $2.453125 / 2^1 = 1.2265625 \rightarrow 1+127 = 128 = 10000000$
- Mantissa:
 - $1.2265625 - 1 = 0.2265625 = 3/16 + 1/32 + 1/256$
 - This evaluates to 00111010
- Binary representation: 1100 0000 0001 1101 0000 0000 0000 0000
- Binary to hexadecimal: starts with 0x, break into groups of 4
 - 1100 – c
 - 0000 – 0
 - 0001 – 1
 - 1101 – d
 - Each of the 4 groups of 0's are represented by one 0
- **Hexadecimal representation: 0xc01d0000**

2. Hexadecimal: 0x0040203f → convert to 32-bit floating point number:

- First convert to binary:
 - 00 = 0000 0000
 - 40 = 0100 0000
 - 20 = 0010 0000
 - 3 = 0011
 - F = 1111
- Convert 0000 0000 0100 0000 0010 0000 0011 1111, first reverse the bytes:
1111 0011 0000 0010 0000 0100 0000 0000
- This tells us that sign is negative and exponent is 11100110 = 230 → $230-127 = 103$
- Mantissa = 000001000000100000000000 → only significant bits are 2^{-6} and 2^{-13}
- Add up → **32-bit floating point number is $5.8890395 * 10^{-39}$**