

## Project 1 - Example Cases

### CASE 1 (Denormalized):

Input read: -0.009765625

Byte ordering: Little Endian

Floating Point Size: 1 byte

- This is a floating point number:
  - $-0.009765625 = -0.000000101 = -1.01 * 2^{-7}$
- $E = -7$ . Exponent is  $E = \text{exp} - \text{Bias}$ , where  $\text{Bias} = 2^{4-1} - 1 = 7$ . Therefore, the equation is  $-7 = \text{exp} - 7 \rightarrow \text{exp} = 0$ , which indicates this is in denormalized form and the exp is 0.
- The number is then  $-0.101 * 2^{-6}$ .
- Mantissa is 0.101 which indicates that the fraction is 101.
- The number is negative, so the sign bit is 1.
- The number at total is 10000101 which is 0x85 in hexadecimal.
- Since it is only one byte, the printed number will be 85.

### CASE 2 (Rounding)

Input read: 20179.0

Byte ordering: Big Endian

Floating Point Size: 3 bytes

- This is a floating point number:
  - $20179.0 = 100111011010011 = 1.00111011010011 * 2^{14}$
- $E = 14$ . Exponent is  $E = \text{exp} - \text{Bias}$ , where  $\text{Bias} = 2^{8-1} - 1 = 127$ . Therefore, the equation is  $14 = \text{exp} - 127 \rightarrow \text{exp} = 141$  which is 10001101 in binary.
- Mantissa is 1.00111011010011.
  - In the project document, it says "you will only use the first 13 bits of the fraction part (for 3-byte and 4-byte data sizes). You will use "round to nearest even" method for rounding fraction bits to 13 bits."
  - Fraction part is 14 bits in this example and we need to round it to nearest even.
  - The number 1.00111011010011 is at exactly half way since the bits to right of rounding position is "1" (the red bit at the end).
  - Therefore, it will be rounded to 1.00111011010011  $\approx$  1.0011101101010
  - For 3 bytes floating point representation, 15 bits is used for fraction part ( $24-1-8=15$ ). So, the fraction part will be 001110110101000
- The number is positive, so the sign bit is 0.
- The number at total is 010001101001110110101000 which is 0x469DA8.
- The byte ordering is Big Endian, so the floating point number is represented as: 46 9D A8
- In the output file, the printed value will be: 46 9D A8