Lab 2 Worksheet

1. Convert the following decimal numbers to our 8-bit floating-point format with a **3-bit exponent e** and a **4-bit significand g**: **s $e_2e_1e_0$ $g_3g_2g_1g_0$**

Steps:

- Convert integer and fractional parts to binary
- Normalize to the form $1.g_3g_2g_1g_0 \times 2^E$
- Adjust exponent with bias (add $B = 2^{3-1}-1 = 3$)
- Set the sign bit

a.

```
2.25
```

b.

```
-4.75
```

c.

```
1.7
```

2. Convert the following floats from our 8-bit floating-point format with a **3-bit exponent e** and a **4-bit significand** into decimal numbers.

Steps:

- Extract the sign, exponent, and significand
- Normalize the significand: restore implied '1.' and remove trailing zeros.
- De-normalize to make exponent 0
- Convert the integer and fractional parts to decimals
- Set the sign according to sign bit

a.

```
1 101 1100
```

b.

```
1 001 0010
```

c.

```
0 010 0110
```

3. What are the largest (excluding infinity) and smallest positive numbers that can be represented as 32-bit IEEE 754 single precision floats (8-bit exponent, 23-bit significant)?

4. Find two numbers that are represented the same in our 8-bit float format and write them as a float

We are now using our minifloat format instead of the IEEE-based 8-bit format we previously used.

5. Add the following numbers together using our 8-bit float format:

Steps:

- Adjust the mantissa of the number with the smaller exponent by shifting it right until both exponents match
- Add the mantissas together
- Recombine and renormalize the result if necessary

a.

```
0 011 1010 + 0 100 1010          (1.25 + 2.5)
```

b.

```
0 100 0001 + 0 110 1000            (0.25 + 8.0)
```

c.

```
0 100 0111 + 0 100 1001          (1.75 + 2.25)
```

6. Multiply these numbers together using minifloats:

Steps:

- Find the sign of the product, determined by the sign bits of both numbers
- Add the exponents of the two numbers, then subtract the combined exponent bias of 6 to get the exponent value
- Multiply the mantissas together.
- Recombine and renormalize the result if necessary

a.

```
0 011 1010 x 1 100 1010                (1.25 x -2.50)
```

b.

```
1 100 0001 x 1 110 1000               (-0.25 x -8.0)
```

c.

```
0 100 0111 x 0 100 1001                (1.75 x 2.25)
```