Fixed point numbers

- **The fixed point *decimal* number representation**
  - The ***decimal* point:**
    - **Decimal point** = a **point** places in a ***decimal* number representation** to **indicate** the **location of the digit** whose **weight = 1**
  - **Weights** of the **digits** in a **fixed point *decimal* number**:
    - The **digit** that **immediately preceeds** the **decimal point** has **weight = 10^0 = 1**
    - The **weight** of **digits** moving towards **left *increases*** by a **factor of 10**
    - The **weight** of **digits** moving towards **right *decreases*** by a **factor of 10**
  - **Example:**

```
Decimal number:     123.45
                    ^^^ ^^
                    ||| ||
                    ||| |+---  weight = 1/100
                    ||| +----  weight = 1/10
                    ||+------- weight = 1
                    |+-------- weight = 10
                    +--------- weight = 100
```

- **The fixed point *binary* number representation**
  - The **"*binary* decimal" point:**
    - ***Binary* decimal point** = a **point** places in a ***binary* number representation** to **indicate** the **location of the digit** whose **weight = 1**
  - **Weights** of the **digits** in a **fixed point *decimal* number**:
    - The **digit** that **immediately preceeds** the **decimal point** has **weight = 2^0 = 1**
    - The **weight** of **digits** moving towards **left *increases*** by a **factor** of **2**
    - The **weight** of **digits** moving towards **right *decreases*** by a **factor** of **2**
  - **Example:**

```
   Binary number:      101.01
                       ^^^ ^^
                       ||| ||
                       ||| |+---   weight = 1/4
                       ||| +----   weight = 1/2
                       ||+-------- weight = 1
                       |+--------- weight = 2
                       +---------- weight = 4

  The value represented by   101.01  is:

                1*(4) + 0*(2) + 1*(1) + 0*(1/2) + 1*(1/4) =  5 1/4 = 5.25 (decimal)
```

- **Converting: value <==> fixed point *binary* representation**
  - In the next 2 sections, I will show you how to convert:
    - **fixed point *binary* representation** ==> **value** that is represented
    - Given a (fractional) **value** ==> find the **fixed point *binary* representation**

- **Convert: fixed point *binary* representation ==> value represented**
  - **Method**:
    - Compute the **value** represented by a **fixed point *binary* represention** by **adding** the **weighted sum** of the **value** of the **digits** in the representation

  Example:

```
  Given the following fixed point binary representation:

          10111.1011

  The value represented is computed as:

          10111.1011
          ||||| ||||
          ||||| |||+---  1*(1/16)
          ||||| ||+----  1*(1/8)
          ||||| |+-----  0*(1/4)
          ||||| +------  1*(1/2)
          |||||
          ||||+--------  1*(1)
          |||+---------  1*(2)
          ||+----------  1*(4)
          |+-----------  0*(8)
          +------------  1*(16)

    value represented = 16 + 4 + 2 + 1 + 1/2 + 1/8 + 1/16
                      = 23 11/16
                      = 23.6875
```

- **Convert: value ==> fixed point *binary* representation**
  - **Method**

    - ***Split*** the **value** into **2 parts**:

      - An ***integral* part**
      - A ***fractional* part**

    - Find the **binary representation** of the ***integral* part** by **repeatedly *divide* the value** by **2** (to obtain the **powers of 2^n**)

    - Find the **binary representation** of the ***fractional* part** by **repeatedly *multiply* the value** by **2** (to obtain the **powers of 2^{-n}**)

  - **Example:**

    - Given the value **23.6875**

    - ***Split*** the value into **2 parts**:

```
   integral part   = 23
   fractional part = 0.6875
```

    - **Convert** the **integral part 23** by **repeated *division***:

```
             23
        /2 -------- 1 <---- remainder (= digit for the weight 2^0)
             11
        /2 -------- 1 <---- remainder (= digit for the weight 2^1)
              5
        /2 -------- 1 <---- remainder (= digit for the weight 2^2)
              2
        /2 -------- 0 <---- remainder (= digit for the weight 2^3)
              1
        /2 -------- 1 <---- remainder (= digit for the weight 2^4)
              0

   Binary representation for the value 23 = 10111
```

    - **Convert** the **fractional part 0.6875** by **repeated *multiplication***:

```
        0.6875
  x2 ------------   1 <--- Overflow digit (= digit for the weight 2^-1)
        1.375

 remove the overflow digit before continuing: ~~1~~.375 ==> 0.375
```

```
        0.375
```

```
  x2 ------------  0 <--- Overflow digit (= digit for the weight $2^{-2}$)
        0.75

remove the overflow digit before continuing: ~~0~~.75 ==> 0.75
```

```
        0.75
  x2 ------------  1 <--- Overflow digit (= digit for the weight $2^{-3}$)
        1.5

remove the overflow digit before continuing: ~~1~~.5 ==> 0.5
```

```
        0.5
  x2 ------------  1 <--- Overflow digit (= digit for the weight $2^{-4}$)
        1.0

remove the overflow digit before continuing: ~~1~~.0 ==> 0.0

Done  (remainder of the value is 0 !)
```

```
Binary representation for the value 0.6875 = 0.1011
```

- **Summary:**

```
  Binary representation for the value 23 = 10111
  Binary representation for the value 0.6875 = 0.1011
```

Therefore:

```
  Binary representation for the value 23.6875 = 10111.1011
```