Binary Arithmetic

We use the decimal numbering system (base 10) in our everyday lives. This can be seen below:

$$632 = (100 * 6) + (10 * 3) + (1 * 2)$$

This may be seen through the formula

$$\sum_{i=0}^{2} d_i = dn * 10^n + dn_{-1*10^{n-1+\ldots}} d_1 * 10^1 + d_0 * 10^0$$

d_n is the most significant digit and d_0 is the least significant digit.

Generalize - Numbering System with base X

In Base X, a non-negative intereger $d_n d_{n-1} \ldots d_1 d_0$

$$\sum_{i=0}^{2} d_i * Xi = dn * Xn + dn_{-1} X^{n-1+\ldots+} d_1 * X^1 + d_0 * X^0$$

- The same number can have many representations on many bases
- For example, consider the decimal number 23
- Computers use a binary system so 23 would be 10111_2
- Sometimes we use a hexadecimal system to make reading binary representations 'easier'. $23_{10} = 0x17(17_{16})$

`cout` converts the binary storage of the number into the requested format
The C language uses specifiers to convert within `scanf()` this can be seen in C for C++ Users

Base 10	Base 2	Base 8	Base 16
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9

Base 10	Base 2	Base 8	Base 16
10	1010	12	A
11	1011	13	B

Number Representation and Binary Arithmetic

- To convert to binary repeatedly divide the decimal number by 2, until the quotient is 0.
- Collect the remainders as you go
- Write down the remainders from left to right

Index