NUMBER SYSTEMS:

When we type some letters or words, the computer translates them in numbers as computers can understand only numbers. A computer can understand the positional number system where there are only a few symbols called digits and these symbols represent different values depending on the position they occupy in the number.

The value of each digit in a number can be determined using −

- The digit
- The position of the digit in the number
- The base of the number system (where the base is defined as the total number of digits available in the number system)

Decimal Number System

The number system that we use in our day-to-day life is the decimal number system. The decimal numeral system has ten as its base. It is the most widely numerical base. Decimal notation is the writing of numbers in a base-10 numeral system. Positional decimal systems include a zero and use symbols (called digits) for the ten values (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9) to represent any number. These digits are often used with a decimal separator which indicates the start of a fractional part, and with a symbol such as the plus sign(+) or minus sign (−) adjacent to the numeral to indicate its polarity. Each position represents a specific power of the base (10). For example, the decimal number 1234 consists of the digit 4 in the units position, 3 in the tens position, 2 in the hundreds position, and 1 in the thousands position.

Binary Number System

Machine language is binary. This means that the machine language has binary values or two values, the combination of which represents the data. These two states are "on" state represented by 1 and "off" state, represented by "0". Let us start with the more familiar number system, the one where we use the numbers 0 to 1.

The binary numeral system, or base-2 number system represents numeric values using two symbols, 0 and 1. More specifically, the usual base-2 system is a positional notation with a radix of 2. Owing to its straightforward implementation in digital electronic circuitry using logic gates, the binary system is used internally by all modern computers.

Binary system (0,1)

Convert from Decimal to Binary

To convert from decimal to binary you can follow the following steps

1. Divide the number by 2.
2. Get the integer quotient for the next iteration.
3. Get the remainder for the binary digit.
4. Repeat the steps until the quotient is equal to 0.
5. Convert 13_{10} to binary:

Division by 2	Quotient	Remainder
13/2	6	1
6/2	3	0
3/2	1	1
1/2	0	1

Then : it write from down to up

$13_{10} = 1101_2$

Convert 174_{10} to binary:

Division by 2	Quotient	Remainder
174/2	87	0
87/2	43	1
43/2	21	1
21/2	10	1
10/2	5	0
5/2	2	1
2/2	1	0
1/2	0	1

Then $174_{10} = 10101110_2$

Example

$$0.375 \cdot 2 = 0 + 0.75$$
$$0.75 \cdot 2 = 1 + 0.5$$
$$0.5 \cdot 2 = 1 + 0$$

Now, let's just write out the resulting integer part at each step — **0.011**. So, **0.375** in decimal system is represented as **0.011** in binary.

Binary to decimal conversion

The key to understanding why those algorithms work is a base-q expansion of a number. An integer number in any numeric system can be represented in the following form:

$$N = x_n \cdot q^n + \ldots + x_1 \cdot q^1 + x_0 \cdot q^0$$

Where

- **N** is integer
- **x** is the digit (0 through 9 for base-10 system, 0 and 1 for base-2 system)
- **q** is the base value (10 for base-10 system, 2 for base-2 system)

Example

$$12_{10} = 1 \cdot 10^1 + 2 \cdot 10^0$$
$$1100_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

Convert the following

$(1101)_2$---------------------( )$_{10}$

$(1001)_2$---------------------( )$_{10}$

$(1000)_2$---------------------( )$_{10}$

$(1101)_2$---------------------( )$_{10}$

$(1101)_2$---------------------( )$_{10}$

$(15)_{10}$----------------------( )$_2$

$(10)_{10}$----------------------( )$_2$

$(16)_{10}$----------------------( )$_2$

$(66)_{10}$----------------------( )$_2$

$(70)_{10}$----------------------( )$_2$