Week 1 - Number systems

The idea of numbers as “symbols”

Humans have an intuitive understanding of quantities or amounts in an absolute and/or relative sense. E.g. a child already understand the number of items presented to it, or the size of bigger and smaller things, and the capability to contrast that three apples are more than two apples. This understanding is pretty much internal. But humans are social beings - we like to communicate. So how do we communicate this idea of amounts of quantities or the general idea of magnitude of a ? Using symbols!

Every modern idea that we communicate originates from primitive symbols - cave paintings to signify that Tigers are dangerous. Or domestic cats, or dogs which help us in hunting.

We represent ideas or ideas of items in one of 2 ways

1. The way the object looks like - pictures/ paintings.
2. A symbol that we developed or a sound. E.g. ‘A’ in english alphabets is used to describe some phonetics.

If we combine both of them in a uniform way, we come up with a powerful way of communicating ideas with other people.

Numbers like $1, 2, 3, \ldots$ are no different. They are called arabic numbers because of the place of their origin and popular due to colonization of half of the world. But, the idea of quantities predates them. We just chose to draw some symbols and chose how to pronounce them, but the underlying idea is the same - to denote a magnitude.

Place values

Just like we have a limited set of symbols to write out human understandable language, numbers are no different.

Consider the tally system - for each item we count, we draw a vertical line | and when we reach a quantity of 5, we cross the group diagonally. This can get incredibly long and hard to read. Hence, the entire concept of place value systems.

Place value systems are a simpler way of grouping things together. The most natural way we know how to do this is using our fingers to count - in groups of 10. Hence, it is no surprise that we came up with the decimal number system - deci-numeral (10- numerals). To represent the base set of this counting system we just write out symbols from $0 \rightarrow 9$ (10 digits). And when we want to go beyond that, we just group them in bundles of 10. E.g.

$$56 = 50 + 6 = 5 \times 10 + 6$$

Which essentially means

- Count 5 groups of 10 on our fingers or 10 groups of 5
- and add 1 group of 6 to that.

The same can be extended for any quantity. E.g. $5,467,234$.

$$\begin{aligned} 5,467,234 \\ = 5000000 + 400000 + 60000 + 7000 + 200 + 30 + 4 \\ = 5 \times 1000000 + 4 \times 100000 + 6 \times 10000 + 7 \times 1000 + 2 \times 100 + 3 \times 10 + 4 \end{aligned}$$

Which can be read as

- 1 million (in exponent form - 10^6) groups of 5
- 100k (in exponent form - 10^5) groups of 4
- 10k (in exponent form - 10^4) groups of 6
- 1k (in exponent form - 10^3) groups of 7
- 100 (in exponent form - 10^2) groups of 2
- 10 (in exponent form - 10^1) groups of 3
- 1 (in exponent form - 10^0) group of 4

This way of grouping quantities together is what is called the ***place value system***. The number of groups that we have in decimal number system are essentially powers of 10. Reading the number from right to left, we see that each number is counted in some groups of 10. If we consider the right-most position to be 0, then the place value system can be generally written as

$$\begin{aligned} \sum_{i=0}^{k}(n_i \times 10^i) = \\ n_0 \times 10^0 + n_1 \times 10^1 + n_2 \times 10^2 + \ldots + n_k \times 10^k \end{aligned}$$

Where i is the position from $0 \to k$, where k is some quantity and n_i is any number from $0 \to 9$ at position i.

To take an example from above, $5,467,234$ has $k = 6$.

- At position $i = 0$, $n_0 = 4$
- At position $i = 1$, $n_1 = 3$
- At position $i = 2$, $n_2 = 2$
- At position $i = 3$, $n_3 = 7$
- At position $i = 4$, $n_4 = 6$
- At position $i = 5$, $n_5 = 4$
- At position $i = 6$, $n_6 = 5$

So the above can be written as

$$5,467,234 = \sum_{i=0}^{6}(n_i \times 10^i) =$$
$$4 \times 10^0 + 3 \times 10^1 + 2 \times 10^2 + 7 \times 10^3 + 6 \times 10^4 + 4 \times 10^5 + 5 \times 10^6$$

Number systems

Once we realize that place value systems are just a way of counting items based on the group size, we can easily see that we can set that group to anything. Groups of 2, 5, 6 - anything. Some of these are of greater interest to us and so we will focus on that. The group size is often called the ***base*** of a number system.

Also, based on the group size, the number of symbols available to us changes. The higher the base, the more symbols we have at our disposal, and the lower the base, fewer symbols we have to work with.

E.g. for a group of decimal number system AKA base-10 numbers - the numbers we use in our day-to-day lives - we have 10 digits from 0-9 available to us. A typical way of representing them is to write them in subscript format $(number)_{base}$. Decimal numbers like 56 are naturally understood to be $(56)_{10}$.

Arithmetic operations like addition, multiplication are preserved change from number system to number system. What changes is the place value and symbolic representation of a quantity in a number system. The point being that we don't need to abandon arithmetic we learned growing up - just need to adjust our views on the symbolic representation of the same quantity.

> This entire discussion comes from a mature and absolutely gargantuan field of mathematics called as number theory, more specifically modulo arithmetic, if this interests you further and you wish to read.

We will now define some number systems that we come across quite often in computer science and software engineering.

Binary number system

Binary number system is a number system which puts things together in groups of 2. This implies

- The base of the number system is 2. Hence each place value is a power of 2.
- The symbols available to us are only 2 - $0, 1$.

Common notations

1. $(number)_2$
2. `0b<number>`

Some examples

1. $(1001)_2 \equiv (9)_{10}$
2. $(10)_2 \equiv (2)_{10}$
3. `0b101` $\equiv (5)_{10}$
4. .. and so on.

Octal number system

Octal number system is a number system which puts things together in groups of 8. This implies

- The base of the number system is 8. Hence each place value is a power of 8.
- The symbols available to us are only 8 - 0 → 9.

Common notations

1. $(number)_8$
2. `0o<number>`

Some examples

1. $(12)_8 \equiv (10)_{10}$
2. $(5)_8 \equiv (5)_{10}$
3. `0o70` $\equiv (56)_{10}$
4. .. and so on.

Hexadecimal number system

Hexadecimal number system is a number system which puts things together in groups of 16. This implies

- The base of the number system is 16. Hence each place value is a power of 16.
- The symbols available to us are only 16 - 0 → 9 and `A, B, C, D, E, F`. (alphabets can be written in lower case too)

Common notations

1. $(number)_1 6$
2. `0x<number>`

Some examples

1. $(\mathrm{c})_{16} \equiv (12)_{10}$
2. $(4\mathrm{e})_{16} \equiv (78)_{10}$
3. `0x1c8` $\equiv (456)_{10}$
4. .. and so on.

Conversion between number systems

Although the converting from decimal number system to other number systems might seem unnatural at first, it makes sense once you understand the rationale behind it. Converting between 2 different number systems has the effect of saying what is group size in which we are counting the parts in.

If you want to group things together, it can be simply represented by factors of the group size. E.g. 23 can be written equivalently in many ways

1. 2 groups of 10 and the remainder of 3
2. 5 groups of 4 and then a remainder of 3
3. 2 groups of 11 and then a remainder of 1
4. 3 groups of 6 and then a remainder of 5
5. and so on ...

Hence going from our natural way of representing numbers in decimal format to any other number involves repeated division of a number by the base of the number system to which we want to convert to, till the final quotient we get is 0. We then collect all of the remainders resulting from this repeated division process starting from the first to last are arranged from right to left.

Let us take an example of converting the number $(57)_{10}$ to binary.

Here $number = 57$, $base = 2$

$$\begin{aligned}
57 \div 2 &\implies quot = \mathit{28}, rem = \mathbf{1} \\
28 \div 2 &\implies quot = \mathit{14}, rem = \mathbf{0} \\
14 \div 2 &\implies quot = \mathit{7}, rem = \mathbf{0} \\
7 \div 2 &\implies quot = \mathit{3}, rem = \mathbf{1} \\
3 \div 2 &\implies quot = \mathit{1}, rem = \mathbf{1} \\
1 \div 2 &\implies quot = \mathit{0}, rem = \mathbf{1}
\end{aligned}$$

Arranging these remainders from top moving down as right to left in our new base 2 number, up we get `111001`. We can also write this as $(111001)_2$ or `b111001`.

To convert back from the base k system back to decimal number system, we just use our logic from place value system.

In our example above, to convert `b111001` back to decimal number system, we just multiply each digit with their place value, in this case powers of 2. We have 6 digits in this number, so we have 6 positions from $0 \rightarrow 5$

$$\begin{aligned}
\sum_{i=0}^{5}(2^i \times n_i) &= \\
\implies 2^0 \times 1 + 2^1 \times 0 + 2^2 \times 0 + 2^3 \times 1 + 2^4 \times 1 + 2^5 \times 1 \\
\implies 1 \times 1 + 2 \times 0 + 4 \times 0 + 8 \times 1 + 16 \times 1 + 32 \times 1 \\
\implies 1 + 0 + 0 + 8 + 16 + 32 = 57
\end{aligned}$$

We can do the same exercise with converting $(57)_{10}$ to hexadecimal

Here $number = 57$, $base = 16$

$$\begin{aligned}
57 \div 16 &\implies quot = 3, rem = 9 \\
3 \div 16 &\implies quot = 0, rem = 3
\end{aligned}$$

Arranging these remainders from bottom up we get `39`. We can also write this as $(39)_{16}$ or `x39`.

To convert back from hexadecimal system back to decimal we again use our place value system logic. We have 2 digits in this number

$$\begin{aligned}
\sum_{i=0}^{1}(16^i \times n_i) &= \\
\implies 16^0 \times 9 + 16^1 \times 3 \\
\implies 1 \times 9 + 16 \times 3 \\
\implies 9 + 48 = 57
\end{aligned}$$

You can use the same logic to work with the octal number system too. As an exercise try to convert $(67)_10$ to octal number system and use the place value logic to get back the original answer in base-10.

Convenience of converting between Binary and Hexadecimal

Consider the example of $(678)_{10} \equiv$ `(0x2a6)` $\equiv$ `0b1010100110`. If you look closely, and group 4 digits of a binary number starting from the right and prefixing the binary with the appropriate number of zeros such that you can form groups of 4 binary digits uniformly, you can easily lookup the chart below and replace those groups of 4 with their hex equivalent hex digits.

So for `0b1010100110` starting from the right and working our way to the left

1. first group is `0110`
2. second group is `1010`
3. third group has only `10` left. so we prefix this with 2 zeros to make a group of 4 binary digits to get `0010`

In hex

1. `0b0110` = `0x6`
2. `0b1010` = `0xa`
3. `0b0010` = `0x2`

Which gives us the number `0x2a6` which is what we started with. The same works the other way around perfectly too.

Quick reference chart

Here is a quick reference charts of the first 20 numbers and their octal, binary and hexadecimal representations.

Some quick notes

- Prefixing a number with any number zeros as we have no effect on the magnitude of the number itself, i.e. 1 = 01 = 001.
- The above property applies for number in any base
- All numbers below in base 2, 8, 16 have been 0 padded to give them uniform length. You can ignore the leading zeros safely.

Decimal (base 10)	Hexadecimal (base 16)	Binary (base 2)	Octal (base 8)
0	0x00	0b00000000	0o00
1	0x01	0b00000001	0o01
2	0x02	0b00000010	0o02
3	0x03	0b00000011	0o03
4	0x04	0b00000100	0o04
5	0x05	0b00000101	0o05
6	0x06	0b00000110	0o06
7	0x07	0b00000111	0o07
8	0x08	0b00001000	0o10
9	0x09	0b00001001	0o11
10	0x0a	0b00001010	0o12
11	0x0b	0b00001011	0o13
12	0x0c	0b00001100	0o14
13	0x0d	0b00001101	0o15
14	0x0e	0b00001110	0o16
15	0x0f	0b00001111	0o17
16	0x10	0b00010000	0o20
17	0x11	0b00010001	0o21
18	0x12	0b00010010	0o22
19	0x13	0b00010011	0o23
20	0x14	0b00010100	0o24

Helpful resources

You need to understand the representation to work with these numbers later. However, you are not expected to do these calculations every time by hand. You can feed them to online converters and get your desired result. In fact during your later stages you will be encouraged to use these tools and libraries to work quickly. Base converter is a useful online tool in this category.