Number systems

Decimal	Binary	Octal	Hexadecimal
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
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A
27	11011	33	1B
28	11100	34	1C
29	11101	35	1D
30	11110	36	1E
31	11111	37	1F
32	100000	40	20

The first column to the right shows the first 33 *natural numbers* written in the *decimal* or *base-10* number system. You know the system. There are 10 different symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. In going from one integer the next higher one, use them in that order. When there are no more, put a 1 before them: 10, 11, 12, …, 19. Then 20, 21, …, 29. After 99 comes 100, and so forth.

In the *binary* or *base-2* number system, two symbols are used: 0 and 1. They are called *bits*. The same method of writing successive integers is used, using the two bits. You can see this in the second column in the table to the right; each entry contains the binary representation of the decimal number to its left. Thus,

$$16_{10} = 10000_2$$

In this equation, a subscript gives the base in which the integer is written. So, read the above line as “16 in the base-10 system equals 10000 in the base-2 system.”

The octal system is similar but it uses 8 symbols. That’s the third column in the table.

The hexadecimal system is similar but it uses 16 symbols: 0, 1, …, 9, A, B, C, D, E, F. That’s the fourth column in the table.

Use in computers

Digital computers generally use the base-2 system because it’s easy to represent bits 0 and 1 by physical devices. For example, typically,

off means 0, on means 1
not magnetized means 0, magnetized means 1
not charged electrically means 0, charged means 1

Relation between binary, octal, and hexadecimal

Take any base-2 integer, like 10110110101. Put a space before each three bits, starting on the right: 10 110 110 101. Now replace each part by its octal equivalent from the table above: 2665. Therefore,

$$10110110101_2 = 2665_8$$

To write any binary integer in hexadecimal, do the same thing but break the binary integer into 4-bit parts. For example, write 10110110101 as 101 1011 0101 and put each part into hexadecimal:

$$10110110101_2 = 5B5_{16}$$

So any binary integer can easily be written in a more compact representation in octal or hexadecimal. On the IBM7090, in the 1950’s and 60’s, one would get a “dump” of memory if a program crashed: a listing of the contents of every memory location, and each would be written in octal.

Today, Unicode characters are written in hexadecimal, though of course in the computer they are in binary. For example, in Java, you can write the character 'a' as '\u0061' where 0061 is in hexadecimal, and

$$0061_{16} = 0000000001100001_2 = 141_8 = 97_{10}$$

For more information on Unicode and character representations, look at JavaHyperText entry “Unicode”.

Number systems

Powers of 2

The powers of 2 are: 1, 2, 4, 8, 16, 32, … In binary, the decimal integer 2^k is a 1 followed by *k* 0's. For example, $2^5 = 32_{10} = 100000_2$. You can also see that any power of 2 written in octal or hexadecimal is all 0's except for the leftmost digit.

General base-b or radix b system

Above, we discussed the decimal, binary, octal, and hexadecimal number systems. These are called the base-10, base-2, base-8, and base-16 number systems. The base is also called the *radix*, so the base-2 uses radix 2. We mention this only because you may come across the term *radix* and wonder what it means.

In general, for any integer $b > 1$, there is the base-*b* or radix *b* number system. It requires *b* symbols, and it follows the pattern shown for the four systems we have looked at.

In general, an integer > 0 is written in the base-*b* system with no leading 0's in the form:

(1) $d_n d_{n-1} \ldots d_1 d_0$ where each d_i is in $0..b\text{-}1$ and $d_n > 0$.

The decimal integer 426 has the value $4*10^2 + 2*10^1 + 6*10^0$.

In the same way, the binary integer 1101_2 has the value $1*2^3 + 1*2^2 + 0*10^1 + 1*10^0$.

And the value of (1) is given by

(2) $d_n*b^n + d_{n-1}*b^{n-1} + \ldots + d_1*b^1 + d_0*b^0 \quad = \quad \Sigma d_k b^k$ over k in $0..n$.

Transforming an integer to base-b.

Given a base *b* number in an array d, this formula (2) above shows you how to calculate its value.

On the other hand, you can use this formula to go the other way: Given an **int** variable *v*, use the formula to calculate and store its base b representation in an array. For example, suppose you want to calculate the base-2 representation of *v*. The formula is:

$$v = d_n*2^n + d_{n-1}*2^{n-1} + \ldots + d_1*2^1 + d_0*2^0 \quad \text{where each } d_k \text{ is in } 0..1$$

Factor out 2 in all but the last term and simplify the last term —remember, $2^0 = 1$:

$$v = 2*(d_n*2^{n-1} + d_{n-1}*2^{n-2} + \ldots + d_1*2^0) + d_0$$

We see that $d_0 = v \% 2$, and the value of the expression within parentheses is v / 2. Thus, d_0 is easy to calculate. Further the expression within parenthesis has the form of the original expression but with one less term, so this process can be repeated, using a loop, to pick off one bit d_k at each iteration. We leave the formulation of the loop invariant and loop to you.

History

Wikipedia (https://en.wikipedia.org/wiki/Arabic_numerals) tells you that the decimal number system is the most common system for representation of numbers in the world. Further, "It was an ancient Indian numeral system which was re-introduced in the *book On the Calculation with Hindu Numerals* written by the medieval-era Iranian mathematician and engineer al-Khwarizmi, whose name was latinized as Algoritmi. The system later spread to medieval Europe by the High Middle Ages." Of course, the Indians did not use the actual symbols 0, 1, 2, … which were introduced later. They used other symbols.

Positional number systems like the decimal and binary systems would not have been possible without the concept of and a symbol for zero. The concept of zero was found in India in a text known as the *Bakshali manuscript*, estimated to have been written in the third or fourth century A.D. or perhaps earlier. Zero is represented by a small dot.