Systèmes de numération en base 2, 8 et 16

Les systèmes de numération font correspondre, à un nombre N, un certain symbolisme écrit ou oral. Dans un système de **base** b>1, les symboles 0, 1, 2 … b – 1 sont appelés **chiffres**. Tout nombre entier positif peut être représenté par une expression de la forme :

$$N = a_n b^n + a_{n-1} b^{n-1} + \ldots + a_1 b^1 + a_0 = \sum_{i=0}^{n} a_i b^i$$

avec $a_i \in \{0,1,\ldots b-1\}$ et $a_n \neq 0$

On utilise la notation condensée équivalente : $N = (a_n a_{n-1} \ldots a_1 a_0)_b$

Ainsi en base 10, le nombre 132 s'écrit : $(132)_{10} = 1 \times 10^2 + 3 \times 10^1 + 2 \times 10^0$

Rappel : un nombre aussi grand soit-il élevé à la puissance 0 est toujours égal à 1.

Dans les ordinateurs, on utilise des **binary digits** (digits binaires) ou **bits**, l'écriture binaire des nombres ne comportant que les deux symboles 0 et 1.

Le nombre décimal 132 est représenté en binaire (b = 2) par 1111011 soit
$1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

Les nombres binaires sont souvent composés d'un grand nombre de bits. On préfère généralement les exprimer dans les systèmes **octal** (b = 8) et **hexadécimal** (b = 16), car la conversion avec le système binaire est simple.

Décimal	Binaire	Hexadécimal	Octal
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	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

Changements de base

- binaire ➔ décimal
  La conversion se fait simplement en additionnant les puissances de 2 correspondant aux bits de valeur 1 :
  $10101_2 = 2^4 + 2^2 + 2^0 = 16 + 4 + 1 = 21_{10}$

ASTUCE : lorsque le nombre binaire à convertir en décimal ne contient que des 1, plutôt que d'additionner chaque puissance de 2, il est plus simple de décaler d'une position vers la gauche et de soustraire 1.

$111111111111_2 = 1000000000000_2 - 1 = 2^{12} - 1 = 4095_{10}$

- décimal ➔ binaire

  La conversion s'effectue par des divisions entières successives par 2. La condition d'arrêt correspond à un quotient nul. Le nombre binaire est obtenu est lisant les restes du dernier vers le premier

25_{10}	$25 \div 2 =$	12 reste **1** ↑
	$12 \div 2 =$	6 reste **0**
	$6 \div 2 =$	3 reste **0**
	$3 \div 2 =$	1 reste **1**
	$1 \div 2 =$	0 reste **1**

  On obtient : $25_{10} = 11001_2$

- octal (hexadécimal) ➔ décimal

  La conversion se réduit à une addition de puissances de 8 (16).

  $123_8 = 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 = 83_{10}$

  $6C5_{16} = 6 \times 16^2 + 12 \times 16^1 + 5 \times 16^0 = 1733_{10}$

- décimal ➔ octal (hexadécimal)

  La conversion correspond à des divisions entières successives par 8 (16). Le nombre octal (hexadécimal) est obtenu en prenant les différents restes du dernier vers le premier.

- octal (hexadécimal) ➔ binaire

  La conversion consiste à remplacer chaque chiffre octal (hexadécimal) par son équivalent binaire sur 3 (4) bits.

  $17_8 = 001\ 111_2$ car $1_8 = 001_2$ et $7_8 = 111_2$

  $2A_{16} = 0010\ 1010_2$ car $2_{16} = 0010_2$ et $A_{16} = 1010_2$

- binaire ➔ octal (hexadécimal)

  On effectue le remplacement, de droite à gauche, de 3 (4) bits par le chiffre octal (hexadécimal) correspondant. Si le nombre de bits n'est pas un multiple de 3 (4), on complète à gauche avec des zéros.

  101101_2 $101\ 101_2 = 55_8$

  101101_2 $0010\ 1101_2 = 2D_{16}$

Représentation des nombres

Sur *p* positions binaires, on peut représenter 2^p valeurs différentes. L'intervalle des valeurs représentables sur *p* bits est donc $[0 \ldots 2^p\text{-}1]$.

Les représentations les plus courantes sont sur 1 octet, 2 octets (mot) ou 4 octets (double mot) :

1 octet	$[0 \ldots 2^8\text{-}1]$	$= [0 \ldots 255]$
2 octets	$[0 \ldots 2^{16}\text{-}1]$	$= [0 \ldots 65\ 535]$