

# Systemes de nombres

# Rappel

**Dans un système en base X, il faut X symboles différents pour représenter les chiffres de 0 à X-1**

Base 2: 0, 1

Base 5: 0, 1, 2, 3, 4

Base 8: 0, 1, 2, 3, 4, 5, 6, 7

Base 10: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Base 16: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

# Systemes de nombres

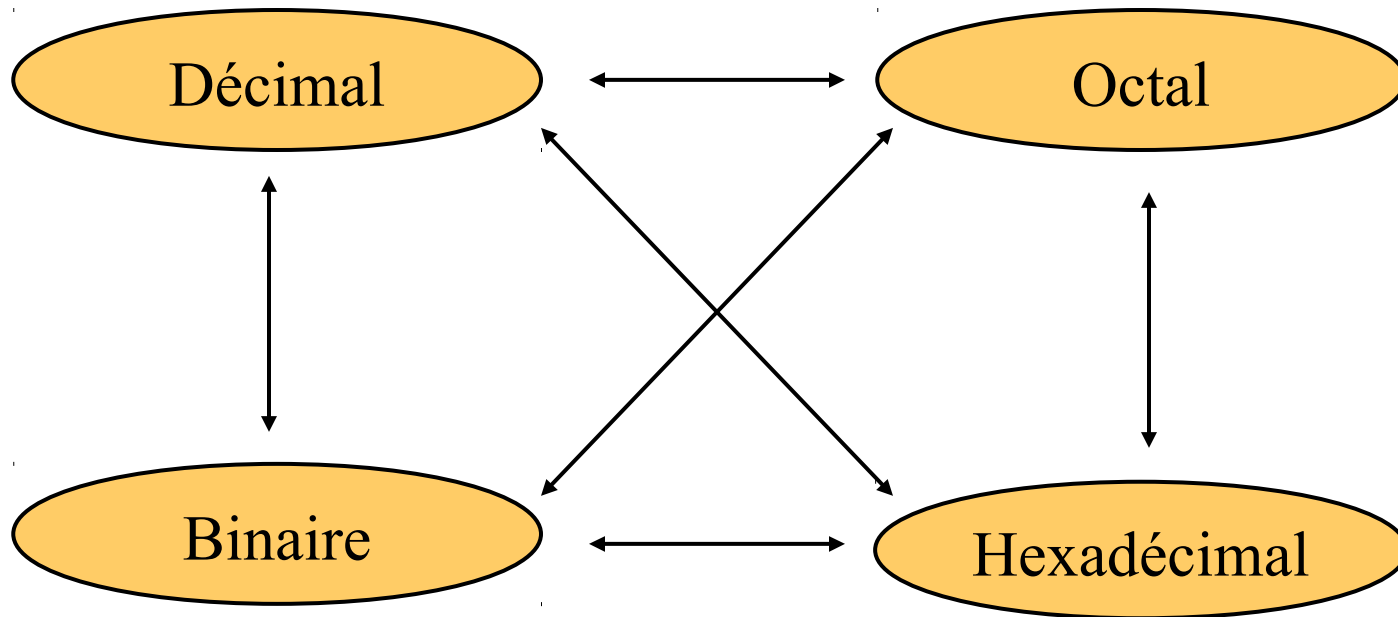
Systeme	Base	Symboles
Decimal	10	0, 1, ... 9
Binaire	2	0, 1
Octal	8	0, 1, ... 7
Hexadecimal	16	0, 1, ... 9, A, B, ... F

# Quantité/Comptage

Décimal	Binaire	Octal	Hexadécimal
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

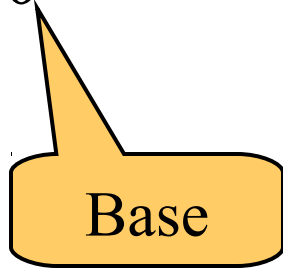
# Conversion d'une base à une autre

- Exemples:



# Exemple

$$25_{10} = 11001_2 = 31_8 = 19_{16}$$



# Rappel, système décimal

Le nombre 125 signifie:

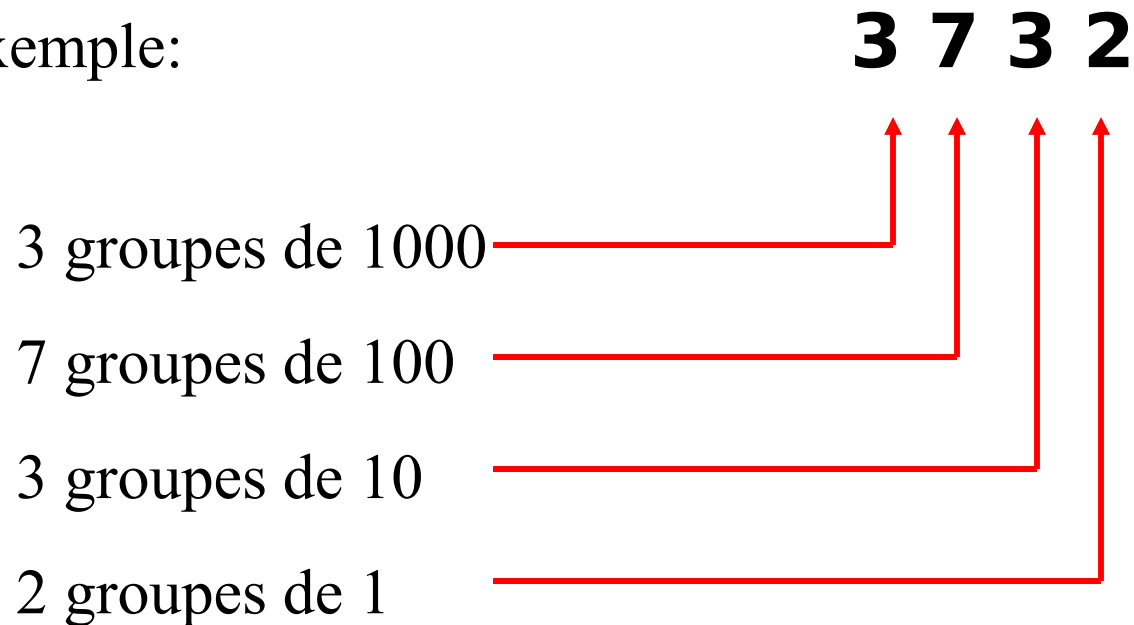
1 groupe de 100 ( $100 = 10^2$ )

2 groupes de 10 ( $10 = 10^1$ )

5 groupes de 1 ( $1 = 10^0$ )

# Placer les valeurs Système décimal

Exemple:





# Représentation d'un nombre N en base X

Représentation d'un nombre N en base X :  $N_x = \sum d_i X^i$

Chiffre de poids fort

Chiffre de poids faible

Poids

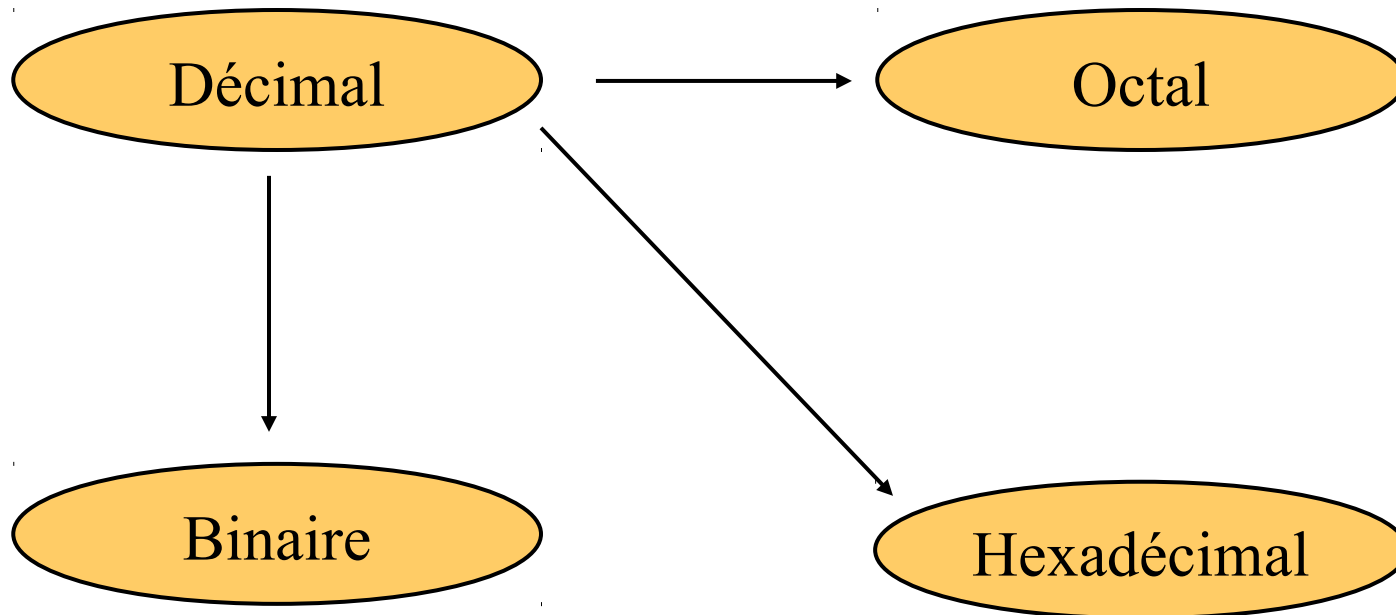
$$\begin{array}{rcl} 125_{10} \Rightarrow & 5 \times 10^0 & = 5 \\ & 2 \times 10^1 & = 20 \\ & 1 \times 10^2 & = \underline{100} \end{array}$$

$$125 = 1 \times 10^2 + 2 \times 10^1 + 5 \times 10^0$$

Base

# Conversion du nombre N exprimé en base 10 vers une base X

- Exemples:



# Conversion du nombre $N$ exprimé en base 10 vers une base $X$

- Conversion d'un nombre **entier**
  - Méthode des divisions successives
  - Méthode des soustractions successives

# Conversion du nombre $N$ exprimé en base 10 vers une base $X$

- Conversion d'un nombre **entier**
  - Méthode des divisions successives
    - $N$  est itérativement divisé par  $X$  jusqu'à obtenir un quotient égal à 0
    - La conversion du nombre  $N$  dans la base  $X$  est obtenue en notant les restes de chacune des divisions effectuées depuis la dernière division jusqu'à la première

# Conversion du nombre N exprimé en base 10 vers une base X

- Conversion d'un nombre **entier**
  - Méthode des divisions successives

$125_{10} = ?_2$

125		2	_____						
<b>1</b>		62		2	_____				
	<b>0</b>	31		2	_____				
		<b>1</b>	15		2	_____			
			<b>1</b>	7		2	_____		
				<b>1</b>	3		2	_____	
					<b>1</b>	1		2	_____
						<b>1</b>	<b>0</b>		

**$125_{10} = 1111101_2$**

# Conversion du nombre $N$ exprimé en base 10 vers une base $X$

- Conversion d'un nombre **entier**
  - Méthode des soustractions successives
    - La plus grande puissance de  $X$  qui est inférieure ou égale à  $N$  est soustraite à  $N$ .
    - Répéter jusqu'à obtenir un résultat égale à 0
    - Le nombre  $N$  exprimé en base  $X$  est obtenu en notant le nombre de fois où une même puissance de  $X$  a été retirée et ce pour chaque puissance depuis la plus grande apparaissant dans l'ordre décroissant des puissances.

# Conversion du nombre N exprimé en base 10 vers une base X

- Conversion d'un nombre **entier**
  - Méthode des soustractions successives

$$235_{10} = ?_8 \quad 8^0 = 1; 8^1 = 8; 8^2 = 64; 8^3 = 512$$

$$235 - 64 = 171; 171 - 64 = 107; 107 - 64 = 43; \Rightarrow 3 \times 64$$

$$43 - 8 = 35; 35 - 8 = 27; 27 - 8 = 19; 19 - 8 = 11; 11 - 8 = 3 \Rightarrow 5 \times 8$$

$$3 - 1 = 2; 2 - 1 = 1; 1 - 1 = 0; \Rightarrow 3 \times 1$$

$$235_{10} = 3 \times 64 + 5 \times 8 + 3 \times 1 = 353_8$$

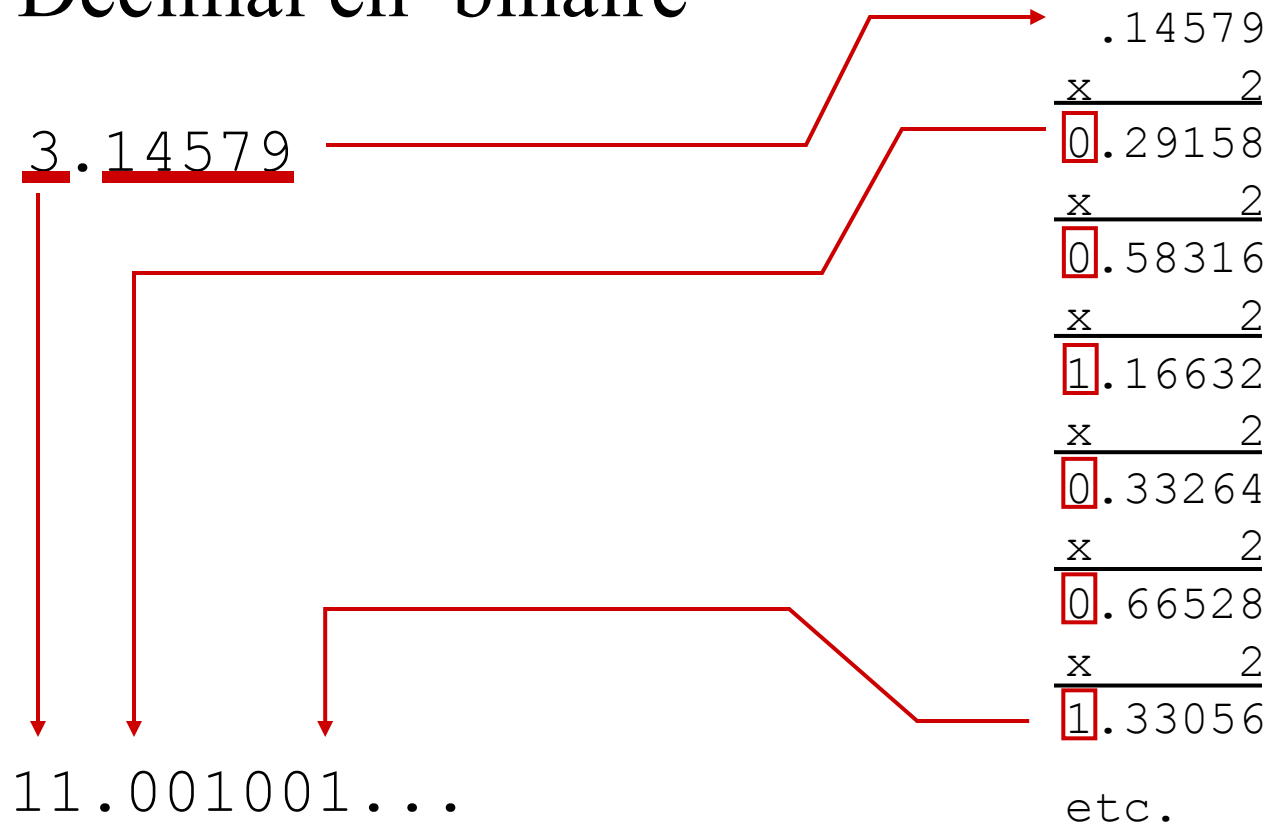
# Conversion du nombre $N$ exprimé en base 10 vers une base $X$

- Conversion d'un nombre **fractionnaire**
  - Nombre  $N$  est fractionnaire
    - Sa partie entière vers une base  $X$ 
      - Méthode des division successives
      - Méthode des soustractions
    - Partie fractionnaire
      - Multiplier cette partie fractionnaire par la base  $X$
      - La multiplication est itérée sur la partie fractionnaire du résultat obtenu
      - Prendre des parties entières de chacun des résultats des multiplications effectuées



# Conversion d'un nombre fractionnaire

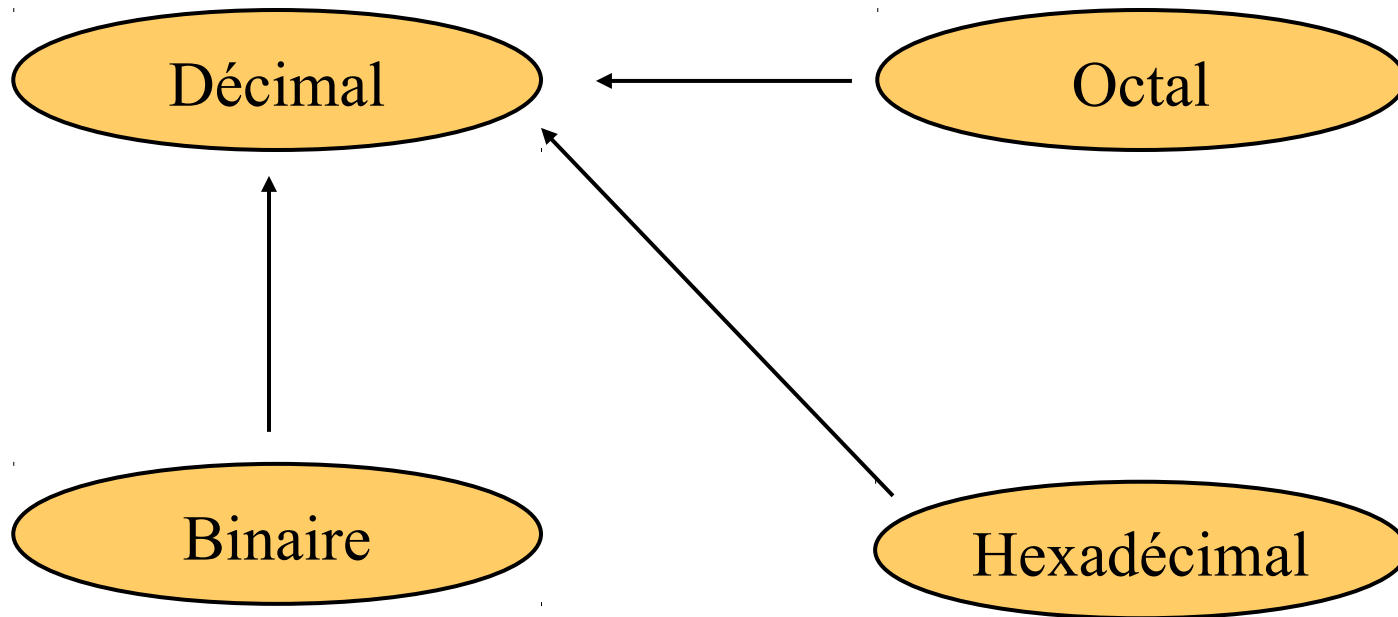
- Décimal en binaire



Le développement s'arrête lorsque la précision voulue est obtenue

# Conversion du nombre N exprimé en base X vers la base 10

- Exemples:



# Conversion du nombre N exprimé en base X vers la base 10

- Technique
  - Multiplier chaque digit par la base  $X^n$ , où  $n$  est le “poids” de ce digit
  - Additionner les résultats

$$N_x = d_n \dots d_0 = d_n \times X^n + d_{n-1} \times X^{n-1} + \dots + d_0 \times X^0$$

# Exemple

Bit "poids 0"

$$\begin{array}{r} 101011_2 \Rightarrow \\ \begin{array}{l} \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \end{array} \end{array} \begin{array}{r} 1 \times 2^0 = 1 \\ 1 \times 2^1 = 2 \\ 0 \times 2^2 = 0 \\ 1 \times 2^3 = 8 \\ 0 \times 2^4 = 0 \\ 1 \times 2^5 = \underline{32} \\ 43_{10} \end{array}$$

# Fractions

- Décimal (rappel)

$$\begin{array}{r} 3.14 \Rightarrow \\ 4 \times 10^{-2} = 0.04 \\ 1 \times 10^{-1} = 0.1 \\ 3 \times 10^0 = 3 \\ \hline 3.14 \end{array}$$

# Fractions

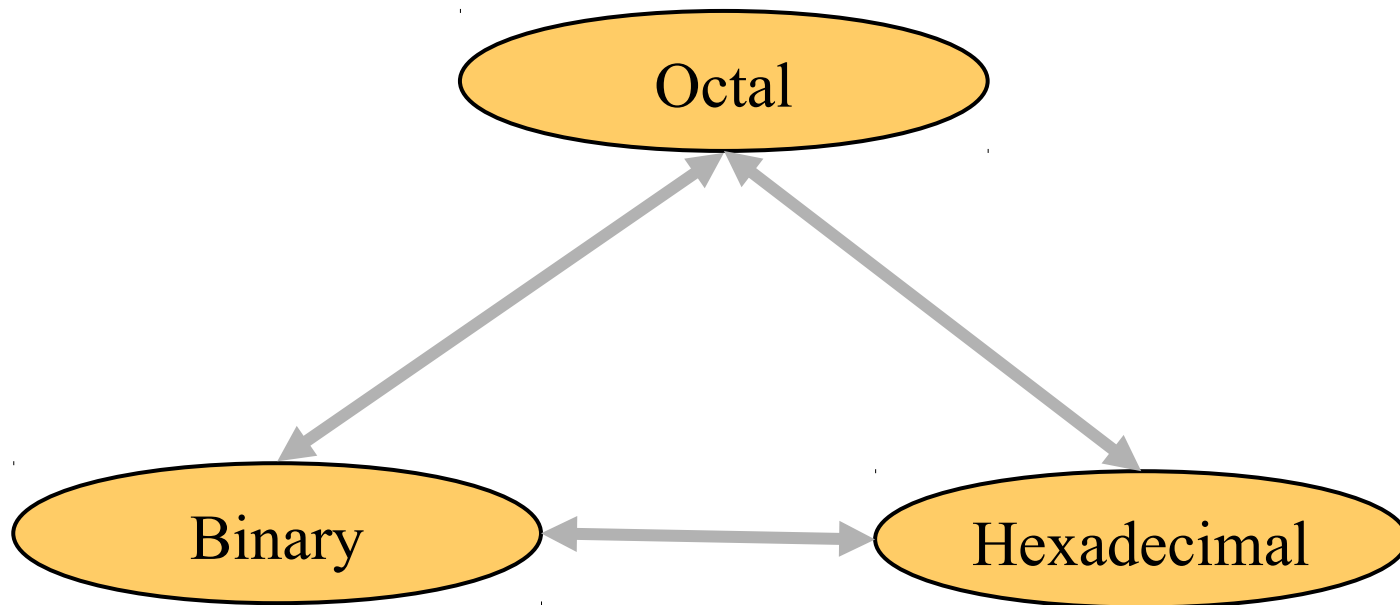
- Binaire vers décimal

$$\begin{array}{r} 10.1011 \Rightarrow \\ 1 \times 2^{-4} = 0.0625 \\ 1 \times 2^{-3} = 0.125 \\ 0 \times 2^{-2} = 0.0 \\ 1 \times 2^{-1} = 0.5 \\ 0 \times 2^0 = 0.0 \\ 1 \times 2^1 = \underline{2.0} \\ 2.6875 \end{array}$$

# Conversion du nombre N exprimé dans la base 8, 16 vers la base 2 (et vice versa)

- Toutes les informations sont représentées dans un ordinateur sous forme d'une chaîne binaire
  - Base de représentation – base 2
  - Chaînes binaires ne sont pas aisément manipulables par l'esprit humain
    - Deux autres bases sont très souvent utilisées
      - La base 8 (système octal)
      - La base 16 (système hexadécimal)

# Conversion du nombre N exprimé dans la base 8, 16 vers la base 2 (et vice versa)



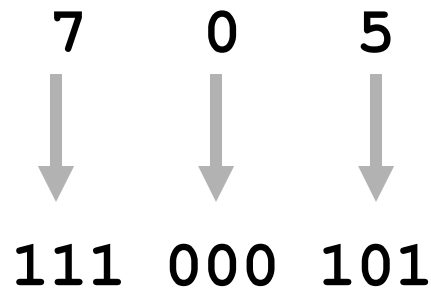


# Conversion du nombre N exprimé dans la base 8 vers la base 2 et vice versa

- Technique
  - Convertir un nombre N exprimé en base 8 vers la base 2 s'effectue en remplaçant chacun des chiffres du nombre par leur équivalent binaire sur 3 bits
  - Convertir un nombre N exprimé en base 2 vers la base 8 s'effectue en découpant la chaîne binaire N en paquet de 3 bits depuis le bit de poids faible jusqu'au bit de poids fort pour la partie entière

# Exemple

$$705_8 = ?_2$$

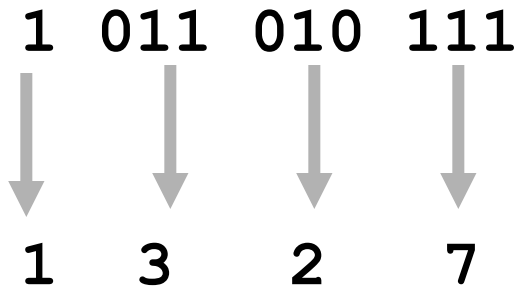


$$705_8 = 111000101_2$$

# Exemple

Digit de poids faible

$$1011010111_2 = ?_8$$



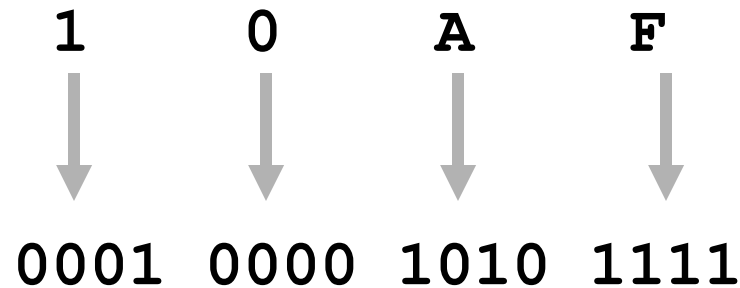
$$1011010111_2 = 1327_8$$

# Conversion du nombre N exprimé dans la base 16 vers la base 2 et vice versa

- Technique
  - Convertir un nombre N exprimé en base 16 vers la base 2 s'effectue en remplaçant chacun des chiffres du nombre par leur équivalent binaire sur 4 bits
  - Convertir un nombre N exprimé en base 2 vers la base 16 s'effectue en découpant la chaîne binaire N en paquet de 4 bits depuis le bit de poids faible jusqu'au bit de poids fort pour la partie entière

# Exemple

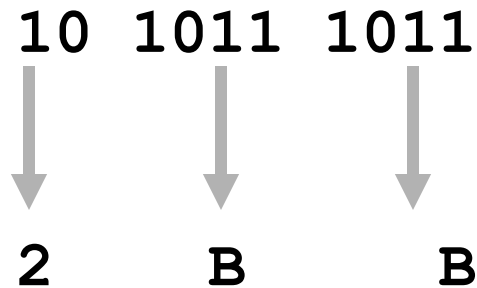
$$10AF_{16} = ?_2$$



$$10AF_{16} = 0001000010101111_2$$

# Exemple

$$1010111011_2 = ?_{16}$$



$$1010111011_2 = 2BB_{16}$$

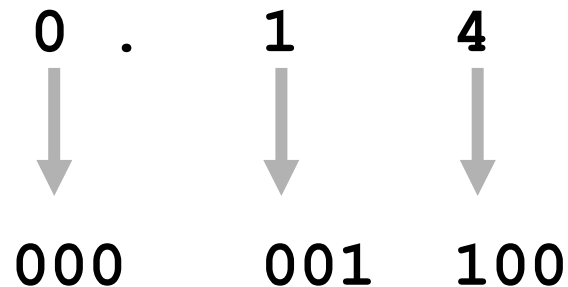
# Fractions

- Technique
  - Convertir un nombre  $N$  exprimé en base 8 (16) vers la base 2 s'effectue en remplaçant chacun des chiffres du nombre par leur équivalent binaire sur 3 (4) bits
  - Convertir un nombre  $N$  exprimé en base 2 vers la base 8 (16) s'effectue en découpant la chaîne binaire  $N$  en paquet de 3 (4) bits depuis le bit de poids **fort** jusqu'au bit de poids **faible** pour la partie fractionnaire

# Fractions

- Octal vers binaire

$$0.14_8 = ?_2$$

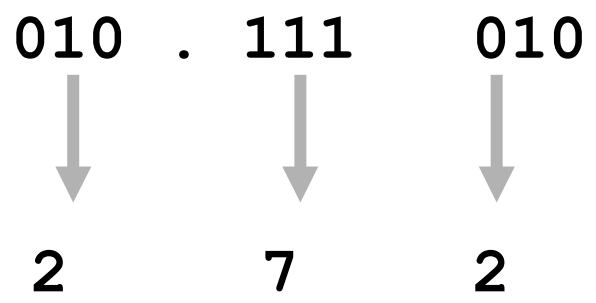
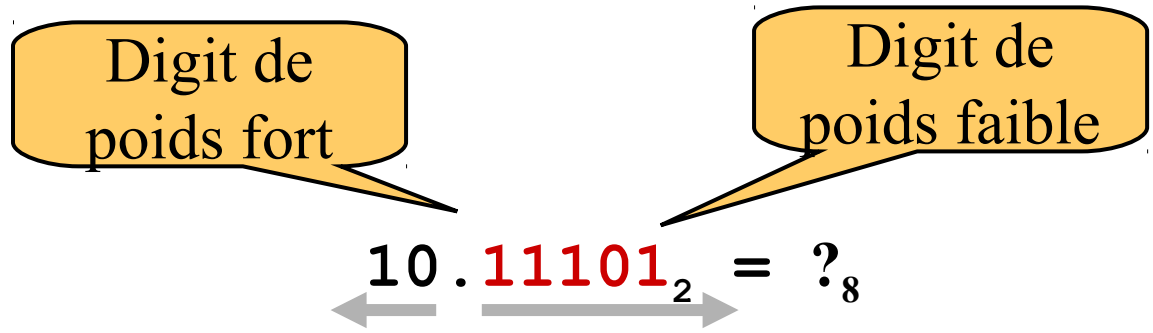


$$0.14_8 = 0.001100_2$$



# Fractions

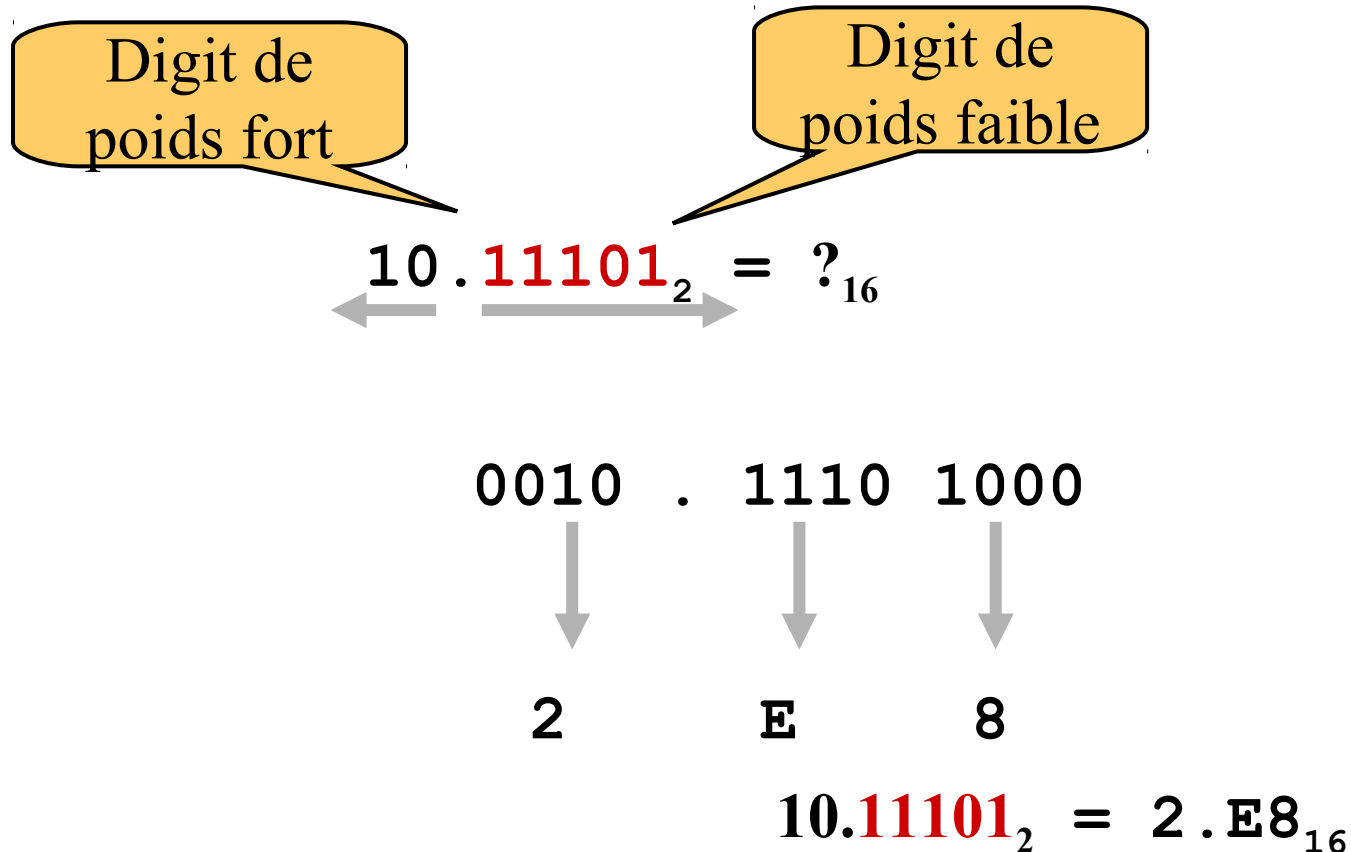
- Binaire vers octal



$$10.11101_2 = 2.72_8$$

# Fractions

- Binaire vers hexadécimal



# Conversion du nombre N exprimé dans la base 8 vers la base 16 et vice versa

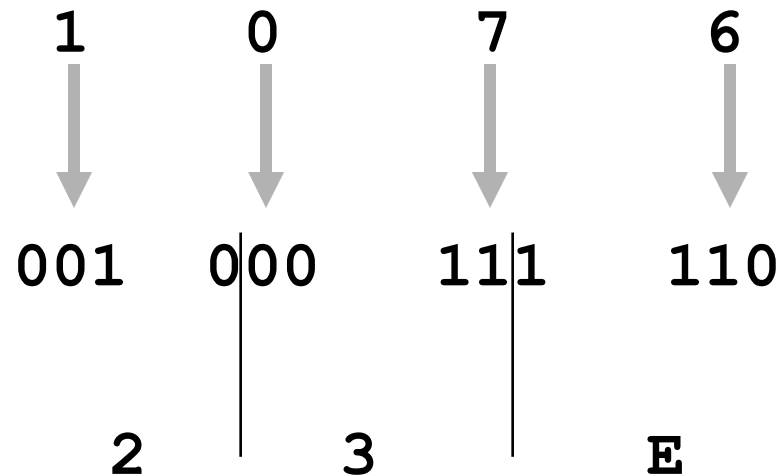
- Technique
  - Utiliser système binaire comme un système intermédiaire

Base 8  $\longrightarrow$  Base 2  $\longrightarrow$  Base 16

Base 16  $\longrightarrow$  Base 2  $\longrightarrow$  Base 8

# Exemple

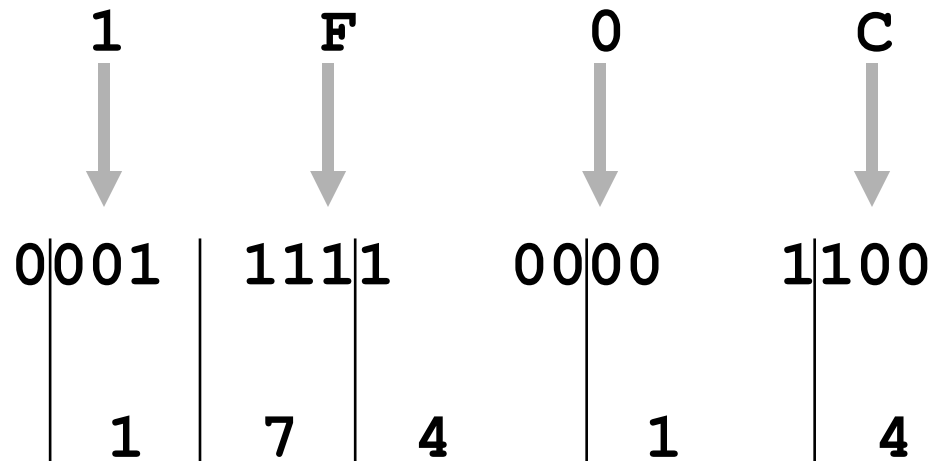
$$1076_8 = ?_{16}$$



$$1076_8 = 23E_{16}$$

# Exemple

$$1F0C_{16} = ?_8$$



$$1F0C_{16} = 17414_8$$

# Mesure de la quantité d'information

- Base 10

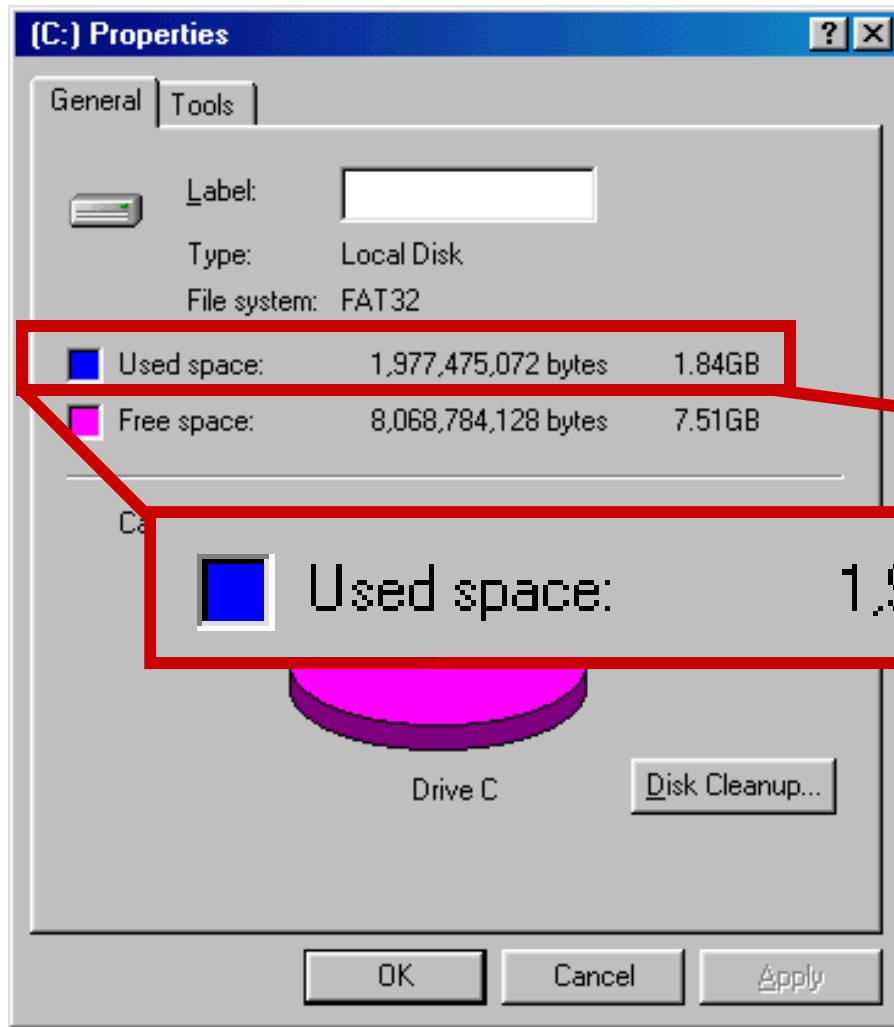
Puissance	Nom	Symbole	Valeur
$10^{-12}$	pico	p	.000000000001
$10^{-9}$	nano	n	.000000001
$10^{-6}$	micro	$\mu$	.000001
$10^{-3}$	milli	m	.001
$10^3$	kilo	k	1000
$10^6$	mega	M	1000000
$10^9$	giga	G	1000000000
$10^{12}$	tera	T	1000000000000

# Mesure de la quantité d'information

- Base 2

Puissance	Nom	Symbole	Valeur
$2^{10}$	kilo	k	1024
$2^{20}$	mega	M	1048576
$2^{30}$	Giga	G	1073741824

# Exemple




$$/ 2^{30} =$$



# Addition binaire

- Deux valeurs de 1 bit

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	10



# Addition binaire

- 2 valeurs de  $n$ -bits
  - Additionner les bits dans chaque position
  - Propager les retenues

$$\begin{array}{r} \phantom{+} \overset{1}{1}0101 \\ + \phantom{+} \overset{1}{1}1001 \\ \hline 101110 \end{array} \qquad \begin{array}{r} \phantom{+} 21 \\ + \phantom{+} 25 \\ \hline 46 \end{array}$$

# Multiplication

- Décimal (rappel)

$$\begin{array}{r} 35 \\ \times 105 \\ \hline 175 \\ 000 \\ 35 \\ \hline 3675 \end{array}$$

# Multiplication

- 2 valeurs de 1-bit

A	B	$A \times B$
0	0	0
0	1	0
1	0	0
1	1	1

# Multiplication

- 2 valeurs de  $n$ -bits
- Comme les valeurs décimales

$$\begin{array}{r} 1110 \\ \times 1011 \\ \hline 1110 \\ 1110 \\ 0000 \\ 1110 \\ \hline 10011010 \end{array}$$