1 – Représentations des entiers naturels en binaire

Nous somme habitués à utiliser l'écriture en base 10 des entiers.
Mais plus généralement, tout entier peut s'écrire dans une autre base, comme la base 2 ou 16, couramment utilisé en informatique car plus adaptées à la mise en mémoire des nombres.

1 – Système de numération positionnel

La notation **positionnelle** est un procédé d'écriture des nombres, dans lequel chaque position d'un chiffre ou symbole est reliée à la position voisine par un multiplicateur.

En base 10 – 237_{10} sera égal à $2x10^2+3x10+7$
En base 8 – 237_8 sera égal à $2x8^2+3x8+7 = 159_{10}$

2 – Les bases privilégiées en informatique

En informatique :

- Base 2 (les bits)
- Base 16 (adresses mémoire, couleurs HTML)
- Base 8 (droit des fichiers UNIX (famille de systèmes d'exploitation))

a – Ecriture en base 16

Pour écrire en base 16, il faut disposer d'un caractère pour chacun des entiers de 0 à 15.
Or, on ne dispose pas d'assez de chiffres pour écrire les 16 chiffres de la base.
On va donc compléter avec les 6 premières lettre de l'alphabet

Base 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Base 16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11

B – Ecriture en base 2

En base 2, tous les nombres sont représentés avec les 2 symboles 0 et 1.

Base 10	0	1	2	3	4	5	6	7	8	9	10
Base 2	0	1	10	11	100	101	110	111	1000	1001	1010

Exemple :

1101_2 = 1x2^3+1x2^2+0x2^1+1x2^0 = 13_{10}

En Python, on utilise des préfixes pour indiquer la base : **0x base 16 / 0b base 2 / 0o base 8**

Exo 1

Parmi ces nombres, lesquels sont potentiellement en base 2 ?

a. 10 b.10011 c. C02KF d.10911 e.102AF

Parmi ces nombres, lesquels sont potentiellement en base 16 ?
a. 10 b.10011 c. C02KF d.10911 e.102AF

Parmis ces écritures, lesquelles sont correctes ?
a. 0x11010 b. 0XAF9 c. 110 d. 0B110

A quel nombre écrit en base 2 correspond le nombre décimal 16 ?
a.10000000 b. 10000 c. 10001

Exo :

Base 2	Base 10
1011	
10101	
110111	
1010111	
11011001	
101010001	
10110110101	
1101010011	
100110100	
1010101010	

3 – Conversion de la base 10 vers la base 2

Pour convertir un nombre de la base 10 vers la base 2, on retranche du nombre la plus grande puissance de 2 possible :

$58_{10} - 1 \times 2^5 = 58_{10} - 32_{10} = 26_{10}$	1	32
$26_{10} - 1 \times 2^4 = 26_{10} - 16_{10} = 10_{10}$	1	16
$10_{10} - 1 \times 2^3 = 10_{10} - 8_{10} = 2_{10}$	1	8
$2_{10} - 0 \times 2^2 = 2_{10}$	0	4
$2_{10} - 1 \times 2^1 = 0_{10}$	1	2
$0_{10} - 0 \times 2^0 = 0_{10}$	0	1

Donc $58_{10} = 111010_2$

Exo

Conversion de base 10 vers base 2

3	
7	
15	
31	
63	
127	

2 – Algorithme de division

Pour convertir un nombre de la base 10 vers la base 2, on effectue des divisions successives de ce nombre par 2.
En lisant les restes de bas en haut, on obtient le résultat.

58_{10} / 2 = 29_{10} reste 0
29_{10} / 2 = 14_{10} reste 1
14_{10} / 2 = 7_{10} reste 0
7_{10} / 2 = 3_{10} reste 1
3_{10} / 2 = 1_{10} reste 1
1_{10} / 2 = 0_{10} reste 1

4 – Autres conversions

De la base 2 vers la base 16, on groupe les bits par paquets de 4, quitte a rajouter des 0 sur la gauche.

Exemple : 10101000011_2 = 0101 0100 0011 = 543_{16}
5 4 3

De la base 16 vers la base 2, on transforme chaque caractère par un groupe de 4 bits.

Exemple : $A3C_{16}$ = 1010 0011 1100 = 101000111100_2
10_{10} 3_{10} 12_{10}

A base 16 = 10 base 2
3 base 16 = 3 base 2
C base 16 = 12 base 2

De la base 16 vers la base 10, on écrit les caractères en base 10 et on utilise la définition avec les puissances de la base de départ.

$A3C_{16} = 10_{10} \times 16\hat{}2 + 3_{10} \times 16\hat{}1 + 12_{10} \times 16\hat{}0 = 10 \times 256 + 3 \times 16 + 12 = 2620_{10}$