LYCÉE de l'ÉLORN
le choix de votre réussite

Sciences de l'Ingénieur

Représentation des données
Numération et codage

I - Numération

1 Les systèmes de base n

1.1 Rappel sur le système décimal

La base 10 utilise 10 symboles différents : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9. Un nombre N (entier positif) exprimé dans ce système de numération est défini par le polynôme :

$N = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \ldots + a_1 \times 10^1 + a_0 \times 10^0$ (où a_n est un chiffre de rang n)

Exemple : $N = (2021)_{10}$
$N = 2 \times 10^3 + 0 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$

Les puissances de 10 sont appelées les **poids** ou les **valeurs de position**. Le poids est égal à la base élevée à la puissance de son **rang**.

	Unités	***Dizaines***	***Centaines***	***Milliers***	***10×Milliers***	***100×Milliers***
Chiffre	a_0	a_1	a_2	a_3	a_4	a_5
Rang	0	1	2	3	4	5
Poids	10^0	10^1	10^2	10^3	10^4	10^5

1.2 Système binaire (binaire pur)

Le système binaire est le système de base 2, il utilise deux symboles différents, le 0 et le 1. Chacun des éléments est appelé bit (contraction de binary digit).

Dans ce système, le poids est une puissance de 2. Pour $N = (110110)_2$ le polynôme s'écrit :

$N = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
$N = (54)_{10}$

Rappel des puissances de 2 :

n	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2^n	0,25	0.5	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536

☞ Convertir en décimal les nombres :

$N = (1011)_2 = (\quad)_{10}$
$N = (00001011)_2 = (\quad)_{10}$
$N = (11110000)_2 = (\quad)_{10}$
$N = (11111110)_2 = (\quad)_{10}$
$N = (11111111)_2 = (\quad)_{10}$
$N = (1111111111)_2 = (\quad)_{10}$

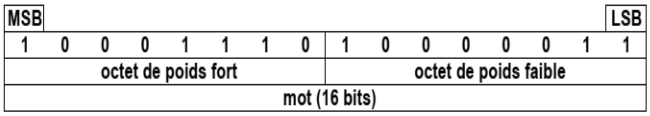
1.3 Format des nombres binaire, définitions :

Octet (byte en anglais) : Nombre binaire formé de 8 bits. *ex : 00000010, 10101111*

Mot (word) : En dehors de l'unité de transfert usuelle (octet), des regroupements plus importants sont couramment utilisés : le mot de 16 bits = 2 octets (**word)**, le mot de 32 bits = 4 octets (**double word**), et le mot de 64 bits = 8 octets (**quad word**).

MSB, LSB : Dans un mot binaire, le bit situé le plus à gauche est le bit le plus significatif. Il sera désigné comme étant le MSB (Most Significant Bit).Celui situé le plus à droite est le bit le moins significatif, LSB (Least Significant Bit).

Remarque : Pour faciliter les manipulations des mots ont peut le diviser en plusieurs octets. Le mot ci-dessous est constitué de 2 octets, celui situé à gauche sera désigné comme étant l'octet de poids fort, et celui situé à droite, l'octet de poids faible.



Quartet : nombre binaire formé de 4 éléments binaires. *Ex : 0001, 1001 ,1111*

Capacités : La capacité en octets des différents constituants tels que les circuits mémoires ou les disques durs est souvent importante. Il devient indispensable d'utiliser **des unités multiples de l'octet.** Historiquement les préfixes « kilo », « méga », « giga » ... représentaient cette capacité naturellement exprimée par une puissance de 2 mais de nouveaux noms ont été créés pour noter ces valeurs. On parle désormais de «kibi», « mébi », « gibi » …

ko (kB) = kilo-octet (kiloByte) = 10^3 octets = 1000 octets
Mo (MB) = Méga-octet (MegaByte) = 10^6 octets = 1000 ko
Go (GB) = Giga-octet (GigaByte) = 10^9 octets = 1000 Mo
To (TB) = Téra-octet (TeraByte) = 10^{12} octets = 1000 Go

kio (kiB) = kibi-octet (kibiByte) = 2^{10} octets = 1024 octets
Mio (MiB) = Mébi-octet (MebiByte) = 2^{20} octets = 1024 kibi-octet
Gio (GiB) = Gibi-octet (GibiByte) = 2^{30} octets = 1024 Mébi-octet
Tio (TiB) = Tébi-octet (TebiByte) = 2^{40} octets = 1024 Gibi-octet

k, M, G, T, … = multiple du système international
b=bit, B=Byte, bi=binary

1.4 Etendue des valeurs en binaire

En utilisant **n** bits, on peut former $\mathbf{2^n}$ nombres différents et le plus grand d'entre eux est égal à $\mathbf{(2^n-1)}$.

Par exemple si n = 8, $N_{max} = (2^8-1) = 255$
On peut former 256 nombres évoluant de 0 $(00000000)_2$ à 255 $(11111111)_2$.

☞ Quelle est l'étendue d'un nombre définie sur 11 bits ?

☞ On souhaite dénombrer un million de valeurs, quelle doit être la grandeur du mot binaire utilisé ?

1.5 Système hexadécimal

Le système hexadécimal est de **base 16** et utilise donc 16 symboles différents. On y trouvera les dix premiers chiffres décimaux : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 et les 6 premières lettres de l'alphabet : A, B, C, D, E, F qui correspondent respectivement aux nombres décimaux de 10 à 15.

Base 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Base 16	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**	**10**	**11**	**12**	**13**
Base 2	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111				

Exemple : $N = (AC53)_{16}$

$N = A \times 16^3 + C \times 16^2 + 5 \times 16^1 + 3 \times 16^0$ *Le poids est une puissance de 16*

$N = 10 \times 16^3 + 12 \times 16^2 + 5 \times 16^1 + 3 \times 16^0$

$N = (44115)_{10}$

☞ *Donnez la valeur en décimale des nombres suivants :*

$N = (1234)_{16} = (\quad)_{10}$ $N = (ABCD)_{16} = (\quad)_{10}$ $N = (FFFF)_{16} = (\quad)_{10}$

2 Changement de base

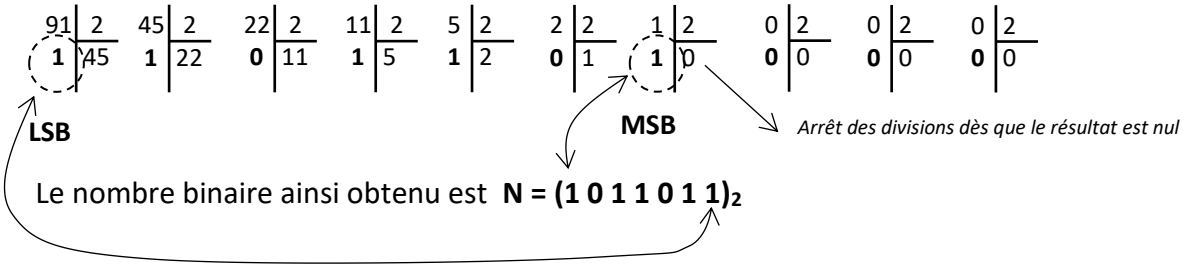
2.1 Conversion d'un nombre décimal en un nombre d'un système d'une autre base :

Un nombre **N** étant donné en base 10, cherchons à l'écrire dans un système de base **b**.

1ère méthode : la division successive

Nous divisons le nombre décimal à convertir par la base b et nous conservons le reste (division entière). Le quotient obtenu est ainsi successivement divisé tant qu'il n'est pas nul. Les restes successifs sont écrits, en commençant par le dernier, de la gauche vers la droite pour former l'expression de N dans le système de base b.

Exemple : Conversion de $N = (91)_{10}$ en un nombre du système binaire (b=2).



☞ Avec cette méthode, convertissez en binaire puis en hexadécimale $(29)_{10}$ $(210)_{10}$ et $(2021)_{10}$

2nd méthode : Approximation successive *(méthode du « compte est bon »)*

On reconstitue le mot binaire en utilisant les poids rencontrés dans la base souhaitée.
☞ Avec cette méthode, convertissez en binaire $(42)_{10}$ et $(203)_{10}$

Poids	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
binaire								
binaire								

2.2 Passerelle entre la notation binaire et hexadécimale

Conversion d'un nombre hexadécimal en un nombre binaire :
Chaque symbole du nombre hexadécimal est remplacé par son équivalent écrit sur 4 bits dans le système binaire.

Exemple : $N = (B\ F\ 8)_{16}$
$N = (1011\ \ 1111\ \ 1000)_2$
B F 8

Conversion d'un nombre binaire en un nombre hexadécimal :
C'est l'inverse de la précédente. Il faut donc regrouper les bits du nombre par quartets en commençant par la droite, puis chaque groupe est remplacé par le symbole hexadécimal correspondant à sa valeur.

Exemple : $N = (100001101111)_2$
$N = (1000\ \ 0110\ \ 1111)_2 = (86F)_{16}$

2.3 Fonction de conversion en Python

Exemple sur la console :

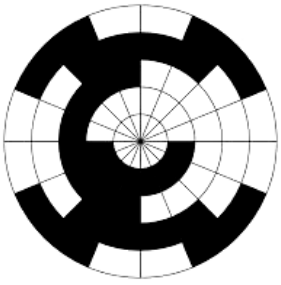
Base10 -> base 2	bin(55) donne 0b110111
Base10 -> base 16	hex(55) donne 0x37
Base2 -> base 10	int('110111',2) donne 55
Base16 -> base 10	int('37',16) donne 55

3 Le code binaire réfléchi (ou code Gray)

Sa propriété réside dans le fait qu'un seul bit change d'état entre deux nombres consécutifs.

Comparaison entre le binaire et le binaire réfléchi

Décimal	Binaire pur	Code Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101



La roue codeuse :
Capteur de position angulaire utilisant le code Gray