1. PARTIE 1 : Représentation des entiers naturels

1.1. Le langage de la machine

1.1.1. Le Binaire

Un ordinateur fonctionne avec du courant électrique car pour agir sur un composant électronique, on doit lui envoyer des signaux électriques. Le microprocesseur est constitué d'un grand nombre de transistors, qui sont comme des minuscules interrupteurs, qui laissent, ou pas, passer du courant.

Dans le langage d'un ordinateur, allumé et éteint sont représentés par 2 états : le courant passe dans le transistor : on utilise le chiffre 1 et le courant ne passe pas on utilise le chiffre 0.

Le langage de l'ordinateur n'est donc constitué que de deux chiffres 0 et 1. Le « mot » le plus élémentaire de ce langage est un mot constitué d'un seul chiffre qui vaut, soit 0, soit 1. C'est ce que l'on appelle un bit (binary digit).

- Avec 1 seul bit, on peut coder 2 valeurs : 0 et 1
- Avec 2 bits on peut coder 4 valeurs : 00, 01, 10 et 11
- Avec 3 bits on peut coder 8 valeurs : 000, 001, 010, 011, 100, 101, 110, 111.
- Avec 4 bits on peut coder 2^4 = 16 valeurs ...

Exercice 1 : Il est possible de choisir le nombre de couleurs affichées à l'écran. En effet les couleurs peuvent être codées sur 4, 8 16 ou 24 bits.

a) Combien de couleurs peut-on coder avec :

- 8 bits ....
- 16 bits ....
- 32 bits ....

b) Calculer le nombre de bits nécessaires pour coder 1000 valeurs différentes : ....

1.1.2. Des ensembles de bits

Les ordinateurs travaillent avec des groupes de 8 bits, appelés octets (ou bytes).

On retiendra que : 1 octet = 1 byte = 8bits

Attention aux multiplicateurs (préfixes)!

Notations :

1 bit : 1b

1 octet : 1o

1 byte : 1B

Nom	Symbole	Valeur
Kilo octet	Ko	10^3 octets
Méga octet	Mo	$10^3 \times 10^3 = 10^6$ octets
Giga octet	Go	$10^3 \times 10^6 = 10^{3+6} = 10^9$ octets
Téra octet	To	$10^3 \times 10^9 = 10^{3+9} = 10^{12}$ octets
Kilo binaire octet (kibioctet)	Kio	2^{10} =1024 octets
Méga binaire octet (mébioctet)	Mio	$2^{10} \times 2^{10} = 2^{20}$ octets
Giga binaire octet (gibioctet)	Gio	$2^{10} \times 2^{20} = 2^{30}$ octets
Téra binaire octet (tibioctet)	Tio	$2^{10} \times 2^{30} = 2^{40}$ octets

Exercice 2 :

a) Convertir 1 Téra octet en Méga octet : ………………………………………………………………

b) Convertir 5 Mio en Kio : ……………………………………………………………………………

c) Une clé USB est vendue avec la mention 32Go, quelle est sa capacité en gibioctets? ………………………

Exercice 3 : Déterminer l'ordre de grandeur des tailles des objets suivants :

Film codé en mp4	Photographie numérique	Fichier texte	Musique codée en mp3	PhotographieRAW
CD ROM	Disque dur	Clé USB	Blue-Ray	Disquette 3.5 pouces

2. Les différentes bases

2.1. La base 10 : le système décimal

Pour que vous compreniez le fonctionnement du binaire, et des systèmes de comptage en général (plus communément appelés bases), nous allons nous appuyer sur le fait que vous sachiez compter...en base 10 (en décimal).

- Il y a 10 chiffres : 0,1,2,3,4,5,6,7,8,9
- Il est interdit d'utiliser autre chose que ces 10 chiffres pour fabriquer un nombre
- Si le rang des unités est plein, il faut passer à celui des dizaines, puis des centaines, milliers etc.

✔ EXEMPLE 1 : 3867 peut être décomposé en puissances de 10 :

$$3867 = 3\times1000+8\times100+6\times10+7\times1 = 3\times10^3+8\times10^2+6\times10^1+7\times10^0$$

Nous avons donc trouvé le chiffre de chaque position pour le nombre 3867.

On dit que notre système de numération est un système de position.

2.2. Autres bases

Partant de ce principe, il est possible de décomposer des nombres décimaux dans d'autres bases.

✔ EXEMPLE 2 : Décomposition de 3867 en base 5

$$3867 = 1\times5^5+1\times5^4+0\times5^3+4\times5^2+3\times5^1+2\times5^0$$

On écrira $3867_{10} = 110432_5$. Comme prévu nous n'avons utilisé que les chiffres 0,1,2,3,4 dans la représentation en base 5.

Exercice 4 : Convertir 3867 en base 3

3. Le binaire

Dans tout ce qui suit, nous n'utiliserons que 2 valeurs : 0 et 1.

Exercice 5 : Ecrire en base 2 les nombres de 0 à 31 sur 5 bits

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

Remarques

- Pour signifier que des valeurs seront écrites en binaires nous utiliserons la notation suivante :

$$1101\ 1011_2$$

- Les bits seront séparés par paquets de 4

  Dans la mesure du possible nous coderont les valeurs binaires avec 4, 8 ou 16 bits

3.1. Passer du binaire au décimal

Pour convertir une valeur binaire en décimal, il suffit de placer les bits dans un tableau des puissances décroissantes de 2. On dit que le bit de gauche est le bit de poids fort et celui de droite, le bit de poids faible :

✔ EXEMPLE 3 : Conversion de $1101\ 1011_2$ en décimal

Rang du bit	7	6	5	4	3	2	1	0
Puissance de 2	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Bit	1	1	0	1	1	0	1	1
Valeur	128	64	0	16	8	0	2	1

Il suffit ensuite d'additionner toutes les valeurs calculées :

$$128+64+16+8+2+1 = 219_{10}$$

Finalement on peut écrire :

$$1101\ 1011_2 = 219_{10}$$

Remarques :

- Avec 8 bits, on peut coder au maximum $2^8 = 256$ valeurs allant de 0 à 255.
- Le bit de poids faible est le seul multiplié par une valeur impaire. C'est donc le bit qui donnera la parité. Si la valeur binaire se termine par 0, le nombre sera pair, sinon il sera impair

Exercice 6 : Convertir $0110\ 0110_2$ en décimal

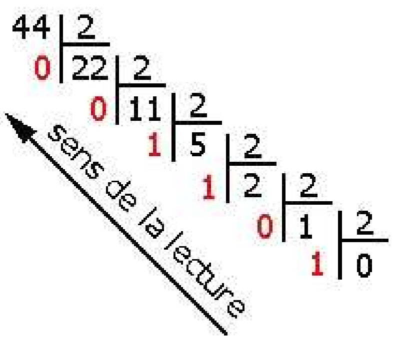
Rang du bit	7	6	5	4	3	2	1	0
Puissance de 2	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Bit								
Valeur								

Donc $0110\ 0110_2$ = ...

3.2. Passer du décimal au binaire

Pour convertir une valeur décimale en binaire, il suffit d'effectuer des divisions successives par 2 jusqu'à ce que le quotient de la division euclidienne soit égal à 0 puis de lire ces valeurs de droite à gauche.

✔ EXEMPLE 4 : Convertir 44_{10} en binaire



On a donc :

$44_{10} = 0010\ 1100$

La valeur 44_{10} se code donc au minimum sur 6 bits. En effet $44 < 2^6 - 1$

Exercice 8 : Convertir 124_{10} et 201_{10} en binaire

3.3. Opérations en binaire

Addition

L'addition se pratique de la même façon que dans le calcul décimal usuel, et repose sur la table d'addition suivante :

+	0	1
0	0	1
1	1	10

MÉTHODE 1 : Poser et effectuer une addition en binaire : $1\ 0101_2 + 1110_2$

Exercice 9 : poser et effectuer les additions suivantes :

$1001\ 1101_2 + 0101\ 1011_2$ $1100\ 1011_2 + 1110\ 0101_2$

Multiplication

La multiplication se pratique de la même façon que dans le calcul décimal usuel, et repose sur la table de multiplication suivante :

×	0	1
0	0	0
1	0	1

MÉTHODE 2 : Poser et effectuer une multiplication en binaire : $1\ 0101_2 \times 101_2$

Exercice 10 : poser et effectuer les multiplications suivantes :

$1001\ 1101_2 \times 11_2$ $1100\ 1011_2 \times 110_2$

4. L'hexadécimal

L'utilisation par l'homme du système binaire est particulièrement délicate (risque d'erreurs). La conversion systématique en décimale est lourde.

Les informaticiens utilisent le système Hexadécimal (base 16). Nous allons voir que le passage de la base 2 à la base 16 est très simple.

> Le système hexadécimal est comme le binaire et le décimal un système de numération de position pondéré. Il nécessite 16 symboles :
>
> {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

Avec en particulier :

A_{16}	B_{16}	C_{16}	D_{16}	E_{16}	F_{16}
10_{10}	11_{10}	12_{10}	13_{10}	14_{10}	15_{10}

Exercice 11 : Écrire en base 16 les nombres de 0 à 31

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

Remarque : Pour signifier que des valeurs seront écrites en hexadécimal nous utiliserons la notation suivante :

$4F_{16}$

4.1. Passer de l'hexadécimal au décimal

Le principe est le même que pour le système binaire. Il suffit de multiplier la valeur hexadécimale par la puissance de 16.

✔ EXEMPLE 5 : Conversion de $AE40_{16}$ en décimal

Rang de la valeur	3	2	1	0
Puissance de 16	16^3	16^2	16^1	16^0
Valeur hexa	A = 10	E = 15	4	0
Valeur	10×16^3	15×16^2	4×16^1	0×16^0

Il suffit ensuite d'additionner toutes les valeurs calculées :

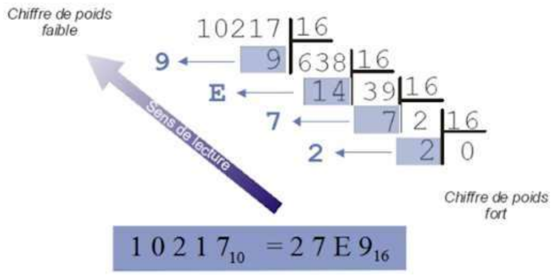
$$AE40_{16} = 10\times16^3 + 15\times16^2 + 4\times16^1 + 0\times16^0 = 44864$$

Exercice 12 : Convertir la valeur F3BC$_{16}$ en décimal

Rang de la valeur	3	2	1	0
Puissance de 16	16^3	16^2	16^1	16^0
Valeur hexa				
Valeur				

4.2. Passer du décimal à l'hexadécimal

Pour passer du décimal à l'hexadécimal, il suffit d'effectuer des divisions successives par 16 jusqu'à ce que le quotient de la division euclidienne soit égal à 0 puis de lire les restes de droite à gauche.



4.3. Passer du binaire à l'hexadécimal

EXEMPLE 6 : Convertir 011 1111 1010 1001$_2$ en hexadécimal.

On regroupe les valeurs binaires par groupe de 4 et on traduit chaque groupe en valeur hexadécimale. Au besoin on rajoute des 0 à gauche.

011 1111 1010 1001$_2$ = 3FA9$_{16}$

4.4. Passer de l'hexadécimal au binaire

La conversion du binaire vers l'hexadécimal (et inversement) se fait directement sans passer par le décimal et très rapidement.

EXEMPLE 7 : Convertir A19D$_{16}$ en binaire

On remplace chaque valeur hexadécimale par son écriture binaire sur 4 bits

A19D$_{16}$ = 1010 0001 1001 1101$_2$

Exercice 13 : Donner l'écriture binaire des nombres suivants :

- 7B$_{16}$ = ...
- C7D3$_{16}$ = ...
- 1234$_{16}$ = ...
- FF$_{16}$ = ...

Exercice 14 :

Donner l'écriture hexadécimale des nombres suivants :

- 145236$_{10}$=...
- 253$_{10}$ = ...
- 1 1010 0111$_2$ = ...
- 110 1011 0101$_2$ = ...

5. Les nombres binaires et hexadécimaux en python

Le langage Python comporte des instructions permettant de passer de la notation binaire ou hexadécimale à la notation décimale, et inversement.

5.1. Binaire

L'instruction **bin(valeur)** convertit des valeurs décimales en binaire.

Ainsi **bin(89)** renvoie la chaine de caractère **'0b1011001'**

Le **0b** au début indique que la chaîne représente un nombre en binaire.

5.2. Hexadécimal

De la même manière, il est possible de convertir du décimal en hexadécimal avec l'instruction **hexa(valeur)**.

Ainsi **hex(60)** renvoie la chaine de caractère **'0x3C'**

Le **0x** au début indique que la chaîne représente un nombre en hexadécimal.

5.3. Ecriture des nombres en binaire ou décimal

En mettant **0b** ou **0x** au début d'un nombre on indique à Python que le nombre qui suit est en base 2 ou en base 16.

Ainsi l'instruction **a=0x4f** fait que la variable `a` prend la valeur 79.

✍ Exercice 15 :

Lancez le logiciel Python afin d'avoir accès à une console. Dans cette console, vous pourrez obtenir immédiatement la valeur des expressions que vous taperez.

Utilisez ce mode immédiat pour compléter le tableau suivant :

Numération décimale	Numération binaire	Numération hexadécimale
12		
		12
	10010110	
		FADE
	121	