

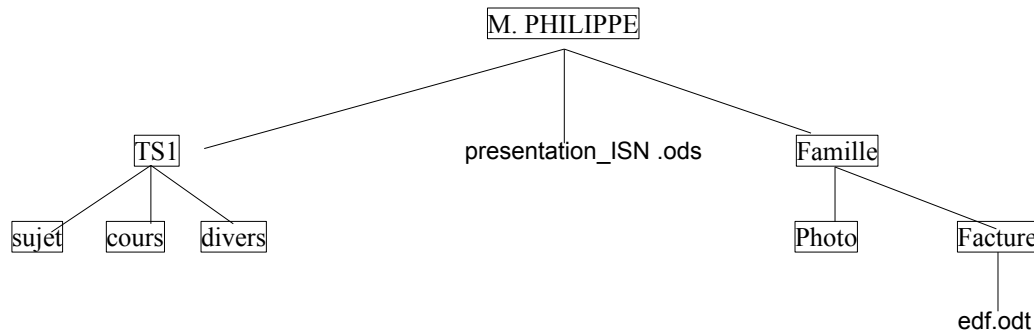
TP1 : Structurer l'information

Objectifs:

- Conversion (octet, Ko, Mo, Go , To)
- organiser une arborescence
- Découvrir un nouveau système de numération

1. Organiser ses fichiers sous forme d'une arborescence

Pour retrouver rapidement des documents stockés sur un ordinateur, il est utile d'organiser son disque dur (ou autre). Cette organisation est une arborescence que l'on peut visualiser sous forme d'un arbre.



Le chemin pour atteindre le fichier edf.odt est alors : **M.PHILIPPE / Famille / Facture / edf.odt**

Afin d'organiser l'année, organiser votre bureau pour classer les fichiers suivants que nous rencontrerons tout au long de l'année :

- les programmes Python brouillons c'est à dire les essais
- les programmes Python finaux c'est à dire qui fonctionnent
- les pages d'information sur la programmation python
- les modalités de l'épreuve d'ISN
- les TP
- les images utilisées durant les cours

2. Comment sont codés ces fichiers ?

Le fonctionnement d'un ordinateur repose sur l'utilisation de transistors . Dans ces transistors, le courant passe (1) ou ne passe pas (0) . Toutes les informations traitées par un ordinateur doivent donc être représentées par des 0 ou des 1 . C'est le codage de l'information en binaire

2.1. Poids d'un fichier

En Anglais ou en Français, on appelle ce 0 ou ce 1 : un BIT (BInaire digiT)

4 bits donne un **nibble**

8 **bits** donne un **octet** (ou **byte** en anglais)

Un **mot** correspond à 2 octets (16 bits)

Un **mot long** à 4 octets (32 bits)

On peut retenir qu'un octet correspond à une lettre ou un symbole

Exemple : 11111111 → Lettre Z

00000001 → Espace

00000111 → ,

Donc un fichier de 150 octets contient 150 symboles tels qu'une lettre, un chiffre ou encore un espace

Comme pour toute unité, le poids d'un fichier peut être décliné en « kilo » , « Méga » , « Giga » , « Tera » , ... Octets.

Exercice 1 :

a) On veut coder 1000 symboles . Combien de bits sont nécessaires ?

b) Comparatif des poids & stockage

Le poids d'un fichier est jaugé en octet .

Pour avoir un ordre de grandeur des échelles de poids voici un liste de poids approximatifs :

Un texte d'une page (lettre)	25 ko
Un morceau de musique mp3 (compression avec perte)	1 Mo/minute
Un morceau de musique wave (compression sans perte)	10 Mo/minute
Un film de 1 h 30 (qualité dvd)	2,5 Go
Un film de 1 h 30 (qualité divx)	700 Mo

Quel est l'ordre de grandeur d'un fichier mp3 de 4 minutes ?

Combien de fichier mp3 de 4 minutes peut-on mettre sur une clé usb de 4 Go ?

Pour info : Les multiples et sous multiples des unités employées en informatique

	Valeur exacte	en pratique dans le système international SI	Exemples d'utilisation
1 kilo-octet(Ko)	$2^{10} = 1024$ octets	10^3 octets	▲ un modem téléphonique débite à 56,6 Kbps, un téléphone GSM à 9.6 Kbps ▲ une page de texte correspond environ 2 Ko.
1 Méga-octet(Mo)	2^{10} Ko = 1 048 576 octets	10^6 octets	▲ un CDROM contient 640 Mo
1 Giga-octet (Go)	2^{10} Mo = 1 073 741 824 octets	10^9 octets	▲ 1 DVD a une capacité de stockage de 4,7 Go 10 mètres de livres dans une bibliothèque tiendraient sur environ 1 Go ▲ un étage rempli complètement de journaux équivaut à 100 Go, c'est à dire au disque dur d'un PC
1 Téra-octet (To)	2^{10} Go = 1 099 511 627 776 octets	10^{12} octets	▲ une grande bibliothèque publique tiendrait sur 1 To environ ▲ l'ordinateur Blue Gene/L d'IBM calcule à la vitesse de 36Tflops (36 Téra opérations par seconde) ▲ YouTube annonce plus de mille milliards de vidéos vues

2.2. Le codage des entiers naturels

En numération décimale, nous n'utilisons que les chiffres de 0 à 9. Cette numération consiste à grouper les objets par paquets de 10.

Par exemple, l'écriture 2359 exprime un nombre formé de 2 milliers, 3 centaines , 5 dizaines et 9 unités c'est à dire $2359 = 2 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 9 \times 10^0$

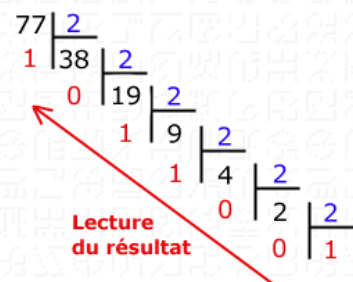
Les ordinateurs, ne fonctionnant qu'avec des 0 ou des 1, regroupent les nombres en paquets de 2 c'est à dire en base deux (ou en binaire)

Ainsi, 1101 en base deux se lit : $1101 \rightarrow 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13$ en base dix

Un algorithme pour trouver cette décomposition consiste à diviser successivement par 2, en division euclidienne, le nombre décimal, puis conserver les restes qui sont égaux à 0 si le nombre est pair ou 1 s'il est impair. On s'arrête dès que le dernier quotient est 1 . L'écriture binaire du décimal est alors illustrée ci-dessous

Exemple : $77 = 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

Ainsi, l'écriture binaire de 77 est 1001101_2



Exercice 1 : En utilisant cet algorithme, décomposer en base 2 les entiers suivants et donner leur écriture binaire :

5 = 10 =
 31 = 45 =
 321 = 73 =

Exercice 2 :

a) Quelle est l'écriture décimale des nombres suivants :

$1011001_2 =$ $1010011110_2 =$
 $1001110_2 =$ $111111_2 =$

b) Pour multiplier un entier naturel exprimé en base 10, il suffit d'ajouter un 0 à sa droite . Quelle est l'opération équivalente pour les entiers naturels exprimés en base 2 ? Exprimer en base 2 les entiers 3 , 6 et 12 pour illustrer la réponse

c) Coder les nombres suivants :

- 25 dans un octet
- 359 dans un mot
- Peut-on implémenter 340 dans un octet ?
- Quel est le plus grand nombre que l'on peut représenter sur un octet ?

Exercice 3 :

Ecrire un algorithme **en langage naturel** qui permet de convertir un nombre en base 2

(A rendre avant la fin de l'heure à l'aide de la feuille annexe)

Exercice 4 : Codage dans une autre base

On peut convertir des nombres dans d'autres bases que la base 2. Le principe de conversion reste le même.

On a vu que la base 2 n'utilise que des 0 et des 1 . La base 8 utilise tous les chiffres de 0 à 7. Mais comment faire en base 16 appelé base hexadécimale? On a besoin de 16 symboles. On dispose des 10 chiffres de 0 à 9 . On utilise ensuite pour 10 , 11 , 12 , 13 , 14 , 15 les lettres de l'alphabet A , B , C , D , E , F .

Ainsi le nombre $5A1_{16}$ correspond à $5 \times 16^2 + 10 \times 16 + 1 = 1441$ en base 10

a) Traduire en décimale le nombre $(B8C)_{16}$

b) A quel nombre décimal correspond FF ?

c) Compléter le tableau suivant en convertissant les nombres dans la base demandée :

Base 10	60			435
Base 2		11001100		
Base 16			A2D	

d) **Pourquoi les informaticiens préfèrent-ils la base 16 ?**

On voit dans le tableau que la base 16 est moins gourmande en longueur d'écriture. Un autre avantage est que le retour de la base 2 vers la base 16 est très simple . Pour cela il suffit de grouper les nombres par quatre bits en partant de la droite. Par exemple, 10011100010000_2 peut se couper en paquets de quatre par : 10 0111 0001 0000 . Chacun de ces paquets en base 2 donne 2 7 1 0 donc en base 16 , 10011100010000_2 correspond à 2710_{16} .

Amusez-vous ainsi à compléter ce tableau

Base 2	11010011001	1001110100110	1000011110111
Base 16			