LYCÉE RÉGIONAL d'Enseignement Général et Technologique Arthur VAROQUAUX	**ISN. TD 07**	**Term S**

Le système binaire…

Ecrire des nombres à l'aide de 0 et de 1 uniquement

1. Principe

Pour comprendre le système binaire, il faut bien comprendre notre système de numération « usuel » : le système décimal.

Le système décimal

Ce système, de base 10, est positionnel.

Il utilise 10 symboles, les chiffres de 0 à 9 mais la valeur d'un symbole dans un nombre dépend de sa position.

Exemple : Pour le nombre 1 231, le « 1 » à droite représente une unité alors que le « 1 » à gauche représente une dizaine de milliers.

Dans le système décimal, tout nombre peut être décomposé en une somme de facteurs de puissances de 10.

Exemple : 1231 = 1 x $\mathbf{10^3}$ + 2 x $\mathbf{10^3}$ + 3 x $\mathbf{10^1}$ + 1 x $\mathbf{10^0}$

Le système binaire

C'est également un système positionnel mais de base 2, utilisant uniquement les deux symboles 0 et 1

Dans le système binaire, tout nombre peut être décomposé en une somme de facteurs de puissances de 2.

Exemple : 1011 = 1 x $\mathbf{2^3}$ + 0 x $\mathbf{2^2}$ + 1 x $\mathbf{2^1}$ + 1 x $\mathbf{2^0}$

Comptons en binaire d'abord 0, puis 1, ensuite 10, et viennent 11, 100, 101, 110, 111, 1000, 1001… soit

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

Exercice 1

Compléter le tableau de conversion suivant :

Décimal	10	11	12	13	14	15	16
Binaire							

Un peu de vocabulaire :

- **bit** : contraction de « *binary digit » littéralement :* « chiffre binaire », c'est un élément (0 ou 1) d'un code binaire.
- **octet** (byte en anglais) est une série de 8 bits. (symbole SI : « o » minuscule)

2. Du binaire au décimal

Pour convertir un nombre binaire dans le système décimal on utilise la décomposition en somme de puissances de 2.

Exemple :

$$110\ 1010_{(2)} = 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$
$$110\ 1010_{(2)} = 1 \times 64 + 1 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1$$
$$110\ 1010_{(2)} = 64 + 32 + 8 + 2$$
$$110\ 1010_{(2)} = 106_{(10)}$$

Ainsi le nombre binaire 110 1010 correspond à 106 dans le système décimal.

Exercice 2

1. Convertir en décimal les nombres binaires suivants :
   $10101_{(2)}$ =
   $111000_{(2)}$ =
   $1001001_{(2)}$ =
   $10001111_{(2)}$ =
2. Quelle est la plus grande valeur décimale possible d'un octet ?

Exercice 3 : Programmation

Ecrire un programme où

1. l'utilisateur entre un octet sous la forme d'une liste de 8 bits
2. une fonction qui détermine si l'entier correspondant est pair ou impair (sans convertir en décimal)
3. une fonction convertit cet octet en entier (base 10).

3. Du décimal au binaire

Pour convertir un nombre décimal en binaire, il suffit de décomposer le nombre décimal en extrayant progressivement les plus grandes puissances de 2 possibles.

Exemple : $37 = 32 + 4 + 1$

$37 = \mathbf{1} \times 2^5 + \mathbf{0} \times 2^4 + \mathbf{0} \times 2^3 + \mathbf{1} \times 2^2 + \mathbf{0} \times 2^1 + \mathbf{1} \times 2^0$

$37_{(10)} = 100101_{(2}$

Pour se faciliter la tâche, on peut établir puis utiliser un tableau dans lequel on indique les valeurs des puissances de 2 utiles.

Exemple : conversion de 172 en binaire

Puissance de 2	$2^8 = 256$	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Reste à convertir	172	172	172-128 = 44	44	44 -32 = 12	12	4	0	0
Nombre binaire	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**0**

$172_{(10)} = 10101100_{(2)}$

Exercice 4

En s'inspirant de l'exemple ci-dessus convertir les nombres suivants dans le système binaire.

a) 18 **b)** 45 **c)** 233 **d)** 666

Exercice 5 (pour les plus rapides)

Généralisons un peu en analysant la procédure de conversion.
On se limite aux nombres n inférieurs ou égaux à 255,c'est-à-dire $n < 2^8$.

Ci-contre un algorithme en langage naturel.

Analyser cet algorithme.
Ecrire ce programme en Python et le tester. (*on finalisera avec une fonction qui prend un entier n en argument et renvoie une liste A de bits*)

n est un entier compris entre 0 et 255
$i \leftarrow 7$
$A \leftarrow [\ \ ]$
Tant que $i \geq 0$:
    Si $n - 2^i \geq 0$ alors :
        Ajouter 1 à la fin de la liste A
        $n \leftarrow n - 2^i$
    Sinon
        Ajouter 0 à la fin de la liste A
    Fin de Si
    $i \leftarrow i - 1$
Fin de Tant que

4. Utilité du binaire

Pour traiter des informations, tous les ordinateurs manipulent des codes binaires. Un ordinateur est constitué de composants électroniques qui ne peuvent gérer que deux états. Les 0 et 1 sont les états logiques qui correspondent physiquement à la présence ou non d'une tension.

Pour un processeur, effectuer une opération sur des nombres ne contenant que des 0 et 1 est beaucoup plus simple et rapide qu'avec des nombres pouvant contenir 10 chiffres différents.

Il est possible de choisir le nombre de couleurs que l'on souhaite voir s'afficher à l'écran. En effet, il peut être affiché, au maximum, soit 16 couleurs, soit 256 couleurs, soit 65 536 couleurs, soit 16 777 216 couleurs, Mais pourquoi ces nombres ??

Tout dépend du type de codage pour chaque pixel.

Application :

a) Si l'on souhaite qu'un pixel ne prenne que 2 couleurs noir ou blanc, combien de bits faut-il utiliser pour coder son état ?
b) Même question si l'on souhaite que le pixel puisse prendre 16 couleurs ?

En réalité chaque pixel est composé de 3 constituants appelés des chromophores (un chromophore rouge, un vert et un bleu).

c) L'intensité de la luminosité de chaque chromophore est codée sur 8 bits. Combien d'états différents peut prendre chaque chromophore ?
d) Sachant que le pixel est composé de 3 chromophores, combien de bits faut-il pour coder un pixel ? combien d'états (de couleurs) différentes peut prendre ce pixel ?