**NUMERATION ET BINAIRE (1)**

*Les ordinateurs codent l’information sous forme binaire. Nous allons voir ici, que ce code binaire correspond à un système de numération à base 2. Ce point est important car il va permettre à l’ordinateur de faire des calculs sur l’information pour la traiter et la transformer.*

**La numération :**

Dans toute numération, il faut distinguer **chiffres** et **nombres**.

* Un nombre est le résultat du comptage d'un ensemble d'objets, d'animaux, de personnes ...
* Pour écrire un nombre, on peut utiliser un ou plusieurs chiffres ( symboles ).

**I. Numération décimale :**

**Dans la vie courante, nous pratiquons la numération décimale …reposant à l’origine sur nos dix doigts : les dix symboles 0,1,2,3,4,5,6,7,8,9 (chiffres) permettent de représenter tous les nombres. (Base 10)**

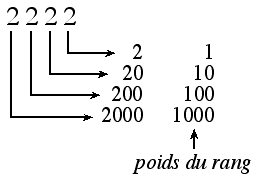
**La position des chiffres est primordiale dans cette représentation (numération de position) : il y a quelques années déjà, vous avez appris ce qu’étaient les unités (colonne de droite), les dizaines, les centaines, etc**

**On peut généraliser cette définition à une numération à base n :**

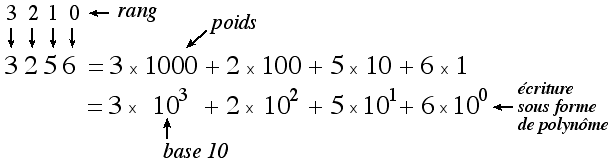
**Si n = 2 alors on a le binaire : tout nombre est écrit avec 0 et 1.**

**Si n = 16 alors on a l’hexadécimal : tout nombre s’écrit grâce à 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, et F**

**Si n = 20 on a le vigésimal (cf. les mayas)**



La place du chiffre dans le nombre a une signification. Il détermine la valeur ou le **poids** du chiffre.

**Exemple**:

Décomposition du nombre 3256 :

On appelle le chiffre dont le poids est le plus élevé le « **Most Significant Bit » ou MSB**.

Ici, c’est le chiffre 3. Ce chiffre est toujours placé le **plus à gauche** du nombre.

On appelle le chiffre dont le poids est le plus faible le « **Least Significant Bit » ou LSB**.

Ici, c’est le chiffre 6. Ce chiffre est toujours placé le **plus à droite** du nombre.

Chaque position a un poids fixe : 100 pour le rang 0 soit un poids de 1, 101 pour le rang 1 soit un poids de 10,   
102 pour le rang 2 soit un poids de 100… C’est un **code pondéré.**

Ecrire une égalité semblable à celle donnée pour 3256 avec les puissances de 10 pour les nombres 2134 et 805.

……………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………..

……………………………………………………………………………………………………………………………..

**II. La base 2 : le binaire**

En base 2, pour représenter les nombres on utilise que 2 chiffres : 0 et 1 appelés bit ( binary digit).

***Exemple : 1001***

Chaque bit correspond à une puissance de 2, le plus à droite correspond à , puis les exposants augmentent de 1 en remontant de la droite vers la gauche.

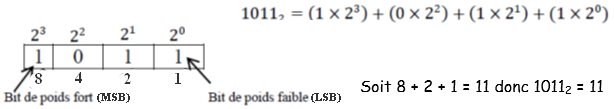
Avec *1* bit, combien de nombres entiers naturels différents peut-on coder ? ...................................  
Avec *2* bits, combien de nombres entiers naturels différents peut-on coder ? ...................................  
Avec *3* bits, combien de nombres entiers naturels différents peut-on coder ? ...................................  
Avec *4* bits, combien de nombres entiers naturels différents peut-on coder ? ...................................

Avec *n* bits, combien de nombres entiers naturels différents peut-on coder ? ...................................

**Correspondance entre binaire et décimal :**

Pour ne pas confondre les écritures, on écrira **1012** pour un nombre en base 2 et **101** pour un nombre en base 10.

**a. Conversion d’un *nombre binaire* en base 10 :**



**b. Conversion d’un *nombre entier naturel* en base 2 :**

**On effectue des divisions euclidiennes par 2, successives, et on s’arrête lorsque le quotient devient nul.**

230 s’écrie donc en binaire 111001102



**On peut aussi décomposer le nombre décimal à l’aide des puissances de 2 : 1, 2 ,4, 8, 16 , 32 , 64 , 128, 256**

Convertissons 230 : 256 est plus grand que 230 donc on commence à 128

230 = 128 + 64 + 32 + 4 + 2 = = 1110 0110

**Exercices** :

1. Convertir en base 10 les nombres suivants : 111111112 ; 100101102

2. Que se passe-t-il si on rajoute un 0 à droite d’un *nombre binaire* ?

4. Convertir en base 2 les nombres suivants 31 ; 109 ; 128 ; 253 ; 1000

**NUMERATION ET BINAIRE (2)**

**III. La base 16 : l‘hexadécimal**

Les nombres binaires étant de plus en plus longs, l’utilisation de la base hexadécimale s’est imposée.

La base hexadécimale consiste à compter sur une base 16, c'est pourquoi au-delà des 10 premiers chiffres on a décidé d'ajouter les 6 premières lettres : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Un chiffre hexadécimal est représenté par 4 chiffres (bits) binaires.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Hexadécimal** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **A** | **B** | **C** | **D** | **E** | **F** |
| Binaire | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

La conversion binaire / hexadécimale est particulièrement simple.

***Conversion binaire / hexadecimal***

Convertissons 01001101 en hexadécimal.

Il suffit de regrouper les bits par quatre (en commençant depuis la droite): 0100 1101

Soit 4D en hexadécimal

|  |  |  |
| --- | --- | --- |
| Binaire | 0100 | 1101 |
| Valeur décimale | 4 | 13 |
| Hexadécimal | 4 | D |

***Conversion hexadecimal / binaire***

* AB(hexa) : on a A=1010 et B=1011 donc : AB(hexa) = 1010 1011(binaire)
* 81(hexa) = 1000 0001(binaire)
* FF(hexa) = 1111 1111(binaire)
* B931(hexa) = 1011 1001 0011 0001(binaire)

***Conversion hexadecimal / decimal***

Dans ce sens, c'est plus simple : prenons un nombre : 4F2C. Il a 4 rangs : chaque rang est une puissance de 16 : pour convertir, on multiplie le premier rang (en partant de la droite) par 160, le second par 161, etc.  
  
Ainsi on obtient :  
4F2C = 4×163 + F×162 + 2×161 + C×160  
4F2C = 4×163 + 15×162 + 2×161 + 12×160  
4F2C = 4×4096 + 15×256 + 2×16 + 12×1  
4F2C hex = 20 268 dec.

***Conversion decimal / hexadecimal***

La conversion d'un nombre de base 10 en base 16 est aussi “facile” qu'avec le binaire.   
Pour le binaire il fallait décomposer en puissances de 2, ici on décompose en puissances de 16.

* 160= 1
* 161= 16
* 162= 256
* 163= 4096
* 164= 65536

Pour l'exemple, je prendrais le nombre 1680. Il faut donc commencer par le décomposer en puissances de 16 :  
1680 = 6×256 + 9×16 + 0×1  
1680 = 6×162 + 9×161 + 0×160.  
  
La conversion en hexadécimal de 1680 est donc 690 (lire “six-neuf-zéro”).