Le terme informatique a été créé pour désigner le traitement automatique de l'information. Or cette information quel que soit sa nature (donnée numérique, texte, instructions …) est codée sous la forme de nombre. Il est donc primordial de savoir comment sont représentés ces nombres dans un ordinateur ou tout autre système numérique.

# Représentation des entiers naturels (positifs)

## Base 10

Quantifier, compter a toujours été essentiel pour les activités humaines. On estime que dès -30000, l'homme manipulait certaines opérations et nombre (les nombres étaient représentés par des crans taillés sur des os d'animaux).En moyen orient, dès -8000, on avait inventé un système de calcul (en base 60) avant l'écriture ! . ( pour plus d'information : <https://lechiffre.info/chapter1/A-Histoire/titre1.html> )

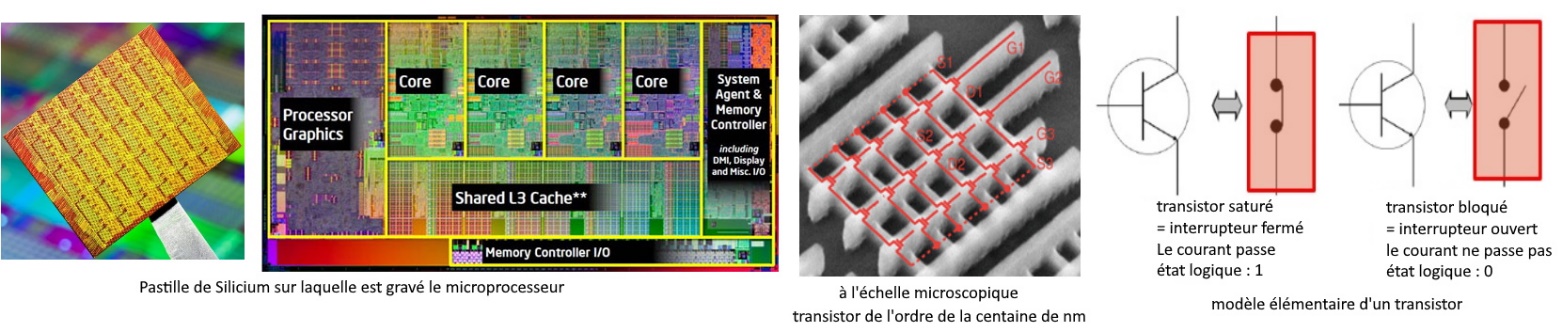
Actuellement, la très grande majorité des civilisations utilisent la base 10 (car on a 10 doigts ! ): on dispose alors de 10 chiffres notés : 0 1 2 3 4 5 6 7 8 9

On compte alors de la manière suivante : 0, 1, 2 ,3 , 4, 6, 7, 8, 9  
Mais à cet instant, on n'a plus de chiffres pour compter d'avantage, on ajoute alors un devant le nombre et on recommence de 0 pour le chiffre le plus à droite : 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 ... jusqu' 99

On a alors des nombres constitués de chiffre (unité, dizaine, centaine...) où chaque chiffre représente une puissance de 10

exemple : 2021=2×103+0×102+2×101+1×100

## Binaire

Coté numération, le problème en informatique, c'est que les ordinateurs n'ont pas de mains, pas de doigts. Donc la base 10 n'est pas forcément pertinente. Les systèmes informatiques fonctionnent à partir d'un composant de base : le transistor. Ces transistors fonctionnent en mode de saturation : transistor saturé ou bloqué.

Conclusion :  
Les systèmes informatiques ne disposent que de 2 états différents possibles : **vrai ou faux (une telle information / variable est appelé booléen)**, qu'on représentera par 2 chiffres 0 et 1. En informatique, le matériel travaille donc en base 2 !

Un chiffre d'un nombre codé en binaire est appelé **un bit**.. Et chacun de ces chiffres représente une puissance de 2 suivant le rang dans le nombre. Un nombre binaire de 8 chiffres est appelé **un octet**.

Pour compter en binaire, on utilise la même méthode qu'en base 10 :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 0 |  |
|  |  |  | 1 | Plus de chiffre dispo, on rajoute un 1 devant |
|  |  | 1 | 0 |  |
|  |  | 1 | 1 | Plus de chiffre dispo, on rajoute un 1 devant |
|  | 1 | 0 | 0 |  |
|  | 1 | 0 | 1 |  |
|  | 1 | 1 | 0 |  |
|  | 1 | 1 | 1 | Plus de chiffre dispo, on rajoute un 1 devant |
| 1 | 0 | 0 | 0 |  |

Chaque bit (chiffre) du nombre binaire représente alors une puissance de 2 (dépendant de son rang dans le nombre)

**Exemple :**  
11001001(2) = 1×20+0×21+0×22+1×23+0×24+0×25+1×26+1×27 =

Rq : Pour distinguer un nombre écrit en base 2, on peut indiquer en indice la base.

Calculer la valeur décimale du nombre suivant : 01011000(2)= 8 + 16 + 64 = 88

Le bit le plus à gauche est appelé bit de poids fort (MSB, most signficant bit en anglais) car bit affecté à la puissance de 2 la plus élevée). Au contraire, le bit le plus à droite est appelé, bit de poids faible (LSB less significant bit).

**Application Python : Convertir un nombre binaire en nombre décimal**

Compléter le fichier python Conversion\_bin\_dec

**Remarques importantes :**

* Nombre maximal : *Nmax=2n−1* (nombre maximal que l'on peut représenter sur n bits )

Exemple : Si on travaille sur un petit microcontrôleur qui travaille avec des registres 8 bits, le nombre entier maximal codable sans 'artifice' (car il existe des systèmes permettant de stocker sur 2 registres une même variable codée sur 14 bits par exemple) serait 11111111(2) soit 28−1 = 256−1 = 255 en décimal.

* Nombre de combinaisons différentes : 2n avec n nombre de bits du registre (ou nombre binaire):

Exemple : Les machines reliées à internet sont identifiées grâce à leur adresse IP, codée sur 32 bits. Combien de machines peuvent être connectées simultanément ?

* Un décalage de bits équivaut à diviser ou multiplier par une puissance de 2

Décaler à gauche les bits d'un nombre binaire d'un rang k revient à le multiplier par 2k et à les diviser par 2k si on décale vers la droite exemple 00001010(2)= 10 et 01010000(2)=10∗23 = 80

**Convertir un nombre décimal en binaire : Méthode par la division euclidienne**

Exemple avec la conversion de 487

**Exercices :**

Convertir les nombres suivants en binaire

345, 2543, 786 et 4724

Écrire une fonction dec\_to\_bin en python pour réaliser cela.

## Hexadécimal

Actuellement, les ordinateurs travaillent sur des registres 64 bits. C’est-à-dire qu'ils manipulent des nombres (des mots) de 64 chiffres binaires. Écrire un nombre ainsi peut vite s'avérer très fastidieux. C'est pourquoi, en informatique, on préfère travailler en base 16 (un chiffre en base 16 représente 4 chiffres en binaire).

Les chiffres en base 16 (en hexadécimal) sont : 0 1 2 3 4 5 6 7 8 9 A B C D E F (A valant 10 ... jusqu'à F=15)

De la même manière que précédemment, dans un nombre codé en hexadécimal, chaque chiffre représente une puissance de 16.

Exemple: BAC(16)=11×162+10×161+12×160 = 2816 + 160 + 12 =

**Exemple d'utilisation :**

Pour coder une couleur, on peut utiliser le système RGB (red, green, blue) où on indique la valeur de chaque composante couleur par un nombre compris entre 0 et 255 ( nombre codé sur 8 bits , que l'on peut aussi représenter sur 2 chiffres hexadécimaux).



À quelle code couleur (R,G,B) en décimal correspond le blanc ?

**Application python :**

En vous inspirant de la fonction bin\_to\_dec(string), donner la fonction renvoyant la valeur décimal d'un nombre codé en hexa.

**Remarque :**

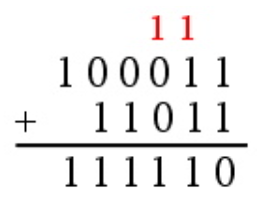
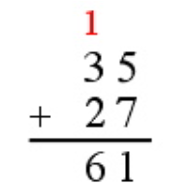
- Pour indiquer qu'un nombre est écrit en base 16, on peut trouver les écritures suivantes : 0x123 (langage C et dérivés), &h123 (BASIC), $123 (en Pascal, et dérivés comme le VHDL en électronique), mais aussi #123 (Common Lisp), 0h123 (Texas Instrument) ou X'123'

## Addition en base n

L'addition dans une base quelconque se fait de la même manière que ce que vous avez appris en primaire. On additionne les chiffres rang par rang, et si l'addition du rang dépasse la valeur de la base, il faut poser une retenue sur le rang suivant.

Exemple en binaire : En sachant qu'en binaire, on a : 1+1=10 ( on pose donc 0, et on met 1 en retenue sur le rang suivant)

Addition décimal Addition binaire



On procède avec la même méthode en hexadécimal.

Application :

Faites l'opération suivante en héxadécimal : A5F(16)+14D(16) = BAD

**Remarque :**

De même, pour la soustraction , on utilise la même méthode qu'en décimal, cf <https://www.youtube.com/watch?v=1jDEmBeMEOA&ab_channel=TopMaths>.

## Passage d'une base à une autre

### Passage de décimal à binaire

Deux méthodes sont utilisables :

* La division successive par 2 :  
  On divise successivement par 2 jusqu'à un résultat de 0, les restes successifs (de bas en haut) forment le nombre binaire. On a déjà utilisé cette méthode.
* Décomposition en puissance de 2 :  
  On décompose le nombre cherché en somme de puissance de 2. (2, 4, 8, 16 ,32 ,64 ,128, 256 ….) Pour cela, on commence par chercher la plus grande puissance de 2 contenu dans le nombre, on calcule alors la différence entre le nombre et la puissance de 2. Et on continue ainsi avec cette différence.

ex : 230 = 128+102 = 128+64+38=...= 128+64+32+4+2 = 27+26+25+22+21 =11100110(2)

### Passage de décimal à hexadécimal

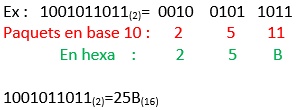
Deux méthodes sont possibles :

* **Division successives par 16** (cf méthode de passage de décimal vers binaire, mais cette fois le reste sera un nombre entre 0 et 15 , correspondant donc aux chiffres 0 à F en héxadécimal)
* Plus simple (pour un calcul à la main), on passe d'abord de décimal à binaire, puis de binaire à héxadécimal (cf paragraphe suivant )

**Application :** Compléter la fonction permettant le passage de décimal à hexadécimal par la méthode de division successives par 16.

### Passage de binaire à hexadécimal (et vice versa)

En notant que 16=24, donc qu'un chiffre en hexadécimale représente 4 bits en binaire, il suffit coder le mot binaire en hexa par paquet de 4 bits, pour chaque paquet donner le nombre décimal représenté puis en déduire le chiffre en hexa.



Pour le passage d'héxadécimal vers binaire, il suffit de suivre la même procédure en sens inverse.

**Application :**Donner le nombre binaire correspondant à : 1F3(16) =

On donne le fichier python pour cette conversion ***conversion\_bin\_hex***.

En utilisant les fonctions déjà écrite, écrire la fonction de conversion hex vers bin.