

Il est possible de stocker différents types d'informations (textes, images, sons...) sur un ordinateur, sous réserve qu'on ait pu les traduire sous forme numérique. Cela suppose qu'on dispose :

- D'un système de *codage* : l'information est transformée en une suite de nombres,
- D'un système de *décodage* : la suite de nombres est transformée en information.

L'ordinateur traite essentiellement des nombres. Pour des raisons techniques, il ne peut se servir que de deux chiffres : le « 0 », le courant ne passe pas et le « 1 », le courant passe. Il lui faut donc écrire les nombres uniquement avec des 0 et des 1. Le système d'écriture des nombres qui n'utilise que le 0 et le 1 est le **système binaire**. Comme dans le système décimal, qui lui utilise 10 chiffres (0 ; 1 ; 2 ; ... ; 8 ; 9), c'est la position d'un chiffre dans l'écriture du nombre qui donne à ce chiffre sa valeur : ainsi dans le système décimal :

$$234\ 056 = 2 \times 100\ 000 + 3 \times 10\ 000 + 4 \times 1\ 000 + 0 \times 100 + 5 \times 10 + 6 \times 1$$

$$= 2 \times 10^5 + 3 \times 10^4 + 4 \times 10^3 + 0 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$$

Quand un nombre a une partie inférieure à 1, on utilise le séparateur constitué par la virgule (ou le point) :

$$20,504 = 2 \times 10 + 5 \times \frac{1}{10} + 0 \times \frac{1}{100} + 4 \times \frac{1}{1000} = 2 \times 10^1 + 5 \times 10^{-1} + 0 \times 10^{-2} + 4 \times 10^{-3}.$$

L'ordinateur n'utilise donc que le 0 et le 1 pour écrire des nombres, mais le principe est le même que dans le système décimal : c'est la position du chiffre dans l'écriture du nombre qui lui donne sa valeur.

Exercice 1

Compléter le tableau ci-dessous en écrivant **dans l'ordre croissant** tous les nombres entiers que l'on peut écrire en n'utilisant **que** le 0 et le 1 et compléter leur valeur en base 10.

Base 2	0	1	10	11											
Base 10	0	1	2	3											

Base 2															
Base 10															

Base 2															
Base 10															

Entourer les nombres dont l'écriture en base 2 est constituée de 1 suivi de zéros. A quel type d'entiers correspondent-ils ?

Quel est le nombre dont l'écriture en base 2 est 100 000 ?

Quel est le nombre dont l'écriture en base 2 est 1 000 000 ?

Quel est le nombre dont l'écriture en base 2 est 10 000 000 000 ?

Exercice 2

Compléter la deuxième ligne du tableau des puissances de 2 ci-dessous :

n	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2^n															
13	-	-	-	-	-	-	-	-	-	-	-	1	1	0	1

Dans la deuxième ligne, comment passe-t-on d'une case à la précédente ?

Écrire comme une somme de nombres contenus dans la deuxième ligne du tableau les nombres suivants et en déduire leur écriture en base 2 : (attention ! chaque nombre de la deuxième ligne ne peut être utilisé qu'une seule fois ! *Exemple : $13 = 8 + 4 + 1$ donc $13 = 1101$ en base 2. On peut aussi écrire $13 = 4 + 4 + 4 + 1$, mais cette écriture utilise 3 fois le 4, ce qui est interdit*).

$24 =$; $57 =$;

$64 =$; $65 =$;

$273 =$; $1858 =$;

Les réponses dans le désordre... 1 000 000 ; 111 001 ; 11 000 ; 100 010 001 ; 1 000 001 ; 11 101 000 010.

Exercice 3

Problème inverse, traduire en nombres décimaux les nombres suivants écrits en base 2 :

$100\ 011 =$; $1\ 000\ 111\ 111 =$; $1\ 010\ 010 =$

A quoi ressemble la mémoire d'un ordinateur ? Si l'ordinateur ne sait traiter que des nombres écrits en base 2, sa mémoire doit être une série de 0 et de 1 ! L'information sur un disque dur doit donc ressembler à ça : 0001110101001011010101111000000000000010100100000000101010011100101010

Comment s'y retrouver ? On regroupe les 0 et 1 par groupe de huit, ce qui donne :

00011101/01001011/010101110/10101011/11100000/00000000/10100100/00000010/10100111/00101010/

Un tel groupe de huit caractères (de huit bits) est appelé un **octet**. C'est l'unité de mémoire de base de l'ordinateur. Le codage des nombres se fait souvent sur 32 ou 64 bits c'est-à-dire sur 4 ou 8 octets.

Quel est le plus grand entier que l'on puisse coder sur 32 bits ?.....
sur 64 bits ?.....
sur n bits ?.....

Pour coder un **entier négatif** en binaire, il faut ajouter une information supplémentaire qui est le signe. On va donc utiliser un bit pour le signe. Il y aura donc un bit de moins pour la partie numérique de l'entier. Par exemple si le premier bit est 0, le nombre est positif, si c'est 1, le nombre est négatif. On rencontre cependant un problème :.....

On va prendre le complément à 2^n : pour représenter un entier négatif en binaire, on inverse tous les bits, on ajoute 1. Par exemple sur 8 bits : $10_{10} = 00001010_2$ et $-10_{10} = 11110110_2$

Coder en binaire : $-1094_{10} =$

A quel nombre décimal correspond le code suivant ? $100000111_2 =$

Avec 8 bits quel est le plus petit entier négatif que l'on puisse coder ? Le plus grand entier positif ? Avec n bits ?.....