Numérique et sciences informatiques
Classe de première

B. Représentation de l'information
2. Ecriture des entiers naturels en binaire

Dans cette séance, nous allons voir comment tout nombre entier naturel peut lui-même être représenté uniquement avec les signes 0 et 1.

Dans la vie courante, nous utilisons dix chiffres distincts (de 0 à 9) pour représenter un nombre entier. On dit que nous utilisons une écriture décimale.
* Comment faire pour représenter le même nombre avec moins de signes, par exemple avec seulement cinq chiffres (de 0 à 4) ?
* Comment faire pour représenter le même nombre en écriture binaire, c'est-à-dire avec seulement deux chiffres : 0 et 1 ?
Pour répondre, il faut nous souvenir comment nous comprenons et utilisons l'écriture décimale.

I Principes de la représentation d'un entier positif en base b

I.1 Vocabulaire et notation

• La représentation des nombres
  * **en décimal** (ou en base 10), c'est l'écriture des nombres avec 10 signes : les chiffres de 0 à 9.
  * **en binaire** (ou en base 2), c'est l'écriture des nombres avec 2 signes : on choisit les chiffres 0 et 1.
  * **en hexadécimal** (ou en base 16), c'est l'écriture des nombres avec 16 signes. On utilise 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

• Compléter la phrase suivante :
* La représentation des nombres en base 5, c'est l'écriture des nombres avec   signes : on choisit les chiffres

Une notation
Soit n un entier supérieur ou égal à 2. Lorsqu'on représente un nombre en base n, on utilise la notion suivante :

$$\underline{78954}_n$$

Lorsque l'on omet cette notation, c'est que l'on utilise la notation décimale (en base 10).

I.2 Principe de représentation d'un nombre dans une base donné

• Commençons par un petit exercice élémentaire : A partir des deux exemples suivant :

$\underline{7594}_{10} = \mathbf{7} \times 10^{\mathbf{3}} + \mathbf{5} \times 10^{\mathbf{2}} + \mathbf{9} \times 10^{\mathbf{1}} + \mathbf{4} \times 10^{\mathbf{0}}$

$\underline{401{,}329}_{10} = \mathbf{4} \times 10^2 + \mathbf{0} \times 10^1 + \mathbf{1} \times 10^0 + \mathbf{3} \times 10^{-1} + \mathbf{2} \times 10^{-2} + \mathbf{9} \times 10^{-3}$

complétez l'égalité suivante :

$\underline{2401}_5 = \quad \times \quad + \quad \times \quad + \quad \times \quad + \quad \times$

$\underline{42{,}103}_3 = \quad \times \quad + \quad \times \quad + \quad \times \quad + \quad \times \quad + \quad \times$

• Essayons maintenant d'écrire une formule complètement générale :

Un nombre décimal D est représenté en écriture décimale par une succession de p chiffres *avant* la virgule et de q chiffres *après* la virgule :

$$\overline{a_{p-1}a_{p-2}\dots a_0, b_1b_2\dots b_q}_{10}$$

où chaque a_i et chaque b_j est un signe appartenant à un ensemble de 10 chiffres correspondant à l'ensemble de nombres $\{0\ ;\ 1\ ;\ 2\ ;\ 3\ ;4\ ;\ 5\ ;\ 6\ ;7\ ;\ 8\ ;\ 9\}$.
Dans ce cas, **en décimal**, on a :

$$D = a_{p-1} \times 10^{p-1} + a_{p-2} \times 10^{p-2} + \cdots + a_0 \times 10^0 + b_1 \times 10^{-1} + b_2 \times 10^{-2} + \cdots + b_q \times 10^{-q}$$

• De même, **en binaire**, si un nombre B est représenté par

$$\overline{a_{p-1}a_{p-2}\dots a_0, b_1b_2\dots b_q}_{2}$$

où chaque a_i et chaque b_j est un signe appartenant à un ensemble de 2 chiffres correspondant à l'ensemble de 2 nombres $\{0\ ;\ 1\}$.
Dans ce cas, on a :

$$B = a_{p-1} \times 2^{p-1} + a_{p-2} \times 2^{p-2} + \cdots + a_0 \times 2^0 + b_1 \times 2^{-1} + b_2 \times 2^{-2} + \cdots + b_q \times 2^{-q}$$

• De même, **en base $\boldsymbol{n}$**, si un nombre N est représenté par

$$\overline{a_pa_{p-1}\dots a_1, b_1b_2\dots b_q}_{n}$$

où chaque a_i et chaque b_j est un signe appartenant à un ensemble de n chiffres correspondant à l'ensemble de n nombres $\{0\ ;\ \dots ;\ \mathrm{n}-1\}$.
Dans ce cas, on a :

$$N = a_{p-1} \times n^{p-1} + a_{p-2} \times n^{p-2} + \cdots + a_0 \times n^0 + b_1 \times n^{-1} + b_2 \times n^{-2} + \cdots + b_q \times n^{-q}$$

I.3 Espace mémoire et nombre de combinaisons possibles

Propriété : Si on est en base b et si l'on dispose de p emplacements différents pour les chiffres, alors on peut écrire b^p *combinaisons différentes*, donc on peut coder b^p nombres différents.
Ainsi, si on veut coder tous les entiers consécutifs à partir de 0 inclus, le plus grand nombre que l'on pourra coder sera $b^p - 1$.

• En particulier, si on code en binaire et si on dispose d'un espace mémoire de p bits, alors *on peut coder* 2^p *nombres différents*.
Si on code tous les entiers consécutifs à partir de 0, le plus grand entier que l'on peut coder est $2^p - 1$.

I.4 Conversions entre le codage binaire et le codage hexadécimal des entiers

Le nombre 16 étant une puissance de 2 (puisque $16 = 2^4$), il existe une correspondance simple entre le codage binaire et le codage hexadécimal des entiers.

C'est pour cette raison que le codage hexadécimal est souvent utilisé en informatique : il permet de coder certains nombres avec moins de chiffres que le codage binaire mais il est toutefois facile de passer de l'écriture en hexadécimal à l'écriture en binaire et *vice versa*.

a) De l'écriture hexadécimale à l'écriture binaire

• Pour passer de l'écriture hexadécimale à l'écriture binaire, il suffit de connaître l'écriture en binaire sur 4 bits des 16 chiffres de l'écriture hexadécimale. Le tableau suivant nous donne cette conversion :

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

• Il suffit alors de remplacer chaque chiffre de l'écriture hexadécimale par son écriture en binaire sur 4 bits pour obtenir son écriture en binaire.
Par exemple, on a :

$$\underline{\mathrm{B83A}}_{16} = \underline{1011\ 1000\ 0011\ 1010}_2$$

b) De l'écriture binaire à l'écriture hexadécimale

• Pour passer de l'écriture binaire à l'écriture hexadécimale, il suffit d'effectuer l'opération inverse : on remplace *chaque paquet de 4 bits* par le chiffre hexadécimal correspondant.
Exemple : Le nombre écrit en binaire sous la forme suivante

$$\underline{1110\ 1001\ 0101\ 1101}_2$$

s 'écrira en hexadécimale sous la forme :

$$\underline{E95D}_{16}$$

II Algorithmes de conversion des nombres entiers

II.1 De l'écriture en base b au nombre N

• b est un entier supérieur ou égal à 2.
On veut déterminer le nombre entier N représenté en base b par la chaine de caractère "$a_{p-1}a_{p-2} \dots a_0$" (il faut bien noter que cette chaine comporte p caractères).

• Rappelons que, pour effectuer la représentation en base b, on dispose de b signes et chacun de ses signes se traduit par un entier compris entre 0 et $b-1$ (par exemple, en base 16 ou hexadécimal, le chiffre « 3 » se traduit par l'entier 3 en décimal et le signe « E » se traduit par l'entier 14).

• Si on suppose que l'on sait lire simultanément tous les caractères de la chaîne $a_{p-1}a_{p-2} \dots a_0$, et que l'on convertit « instantanément » chacun de ces caractères en un entier correspondant, alors on peut directement calculer N par la formule suivante :

$$N = a_{p-1} \times b^{p-1} + a_{p-2} \times b^{p-2} + \cdots + a_0 \times b^0$$

II.2 Du nombre N à l'écriture en base b

• On dispose d'un entier N. On veut convertir ce nombre en une chaine de caractère $a_{p-1}a_{p-2} \dots a_0$ qui représente ce nombre en base b.

• *Un exemple :* on veut déterminer l'écriture *binaire* du nombre 37 :
On a :
* la division de 37 par 2 donne **18** et il reste **1 ;**
* la division de 18 par 2 donne **9** et il reste **0 ;**

* la division de 9 par 2 donne **4** et il reste **1 ;**
* la division de 4 par 2 donne **2** et il reste **0 ;**
* la division de 2 par 2 donne **1** et il reste **0 ;**
* la division de 1 par 2 donne **0** et il reste **1.**

On arrête les divisions par 2 lorsque le quotient obtenu est 0. L'écriture du nombre en écriture décimale est la succession des restes obtenus *en commençant par le dernier et en finissant par le premier*.

On a donc

$$37 = \underline{100101}_2$$

• Algorithme général avec une construction de la chaine « par la gauche » (on écrit la chaine de droite à gauche) :

```
saisir nombre (un entier positif)
saisir base (un entier entre 2 et 10)
 chaine←""
Tant que N ≠ 0 faire
        chiffre ← nombre%base
        nombre ← nombre//base
        convertir chiffre en un caractère
        chaine ← chiffre + chaine
Fin Tant que
Afficher chaine
```

Rappels :

* $\mathrm{nombre}\%\mathrm{base}$ renvoie le *reste* de la division euclidienne de nombre par base ;
* $\mathrm{nombre}//\mathrm{base}$ renvoie le *quotient* de la division euclidienne denombre par base.

Exercices

Exercice 1

1. Trouver « à la main » en suivant l'algorithme décrit au II.2 l'écriture en binaire des entiers 75 et 48
2. Utiliser l'algorithme ci-dessus en pseudo-code pour écrire dans un fichier « codage.py » le codage d'un entier positif dans une base B comprise entre 2 et 10.
3. Ajouter quelques lignes qui permettent de comparer votre résultat avec la fonction **bin(**...**)**.

Exercice 2

1. Tapez le programme ci-dessous dans un fichier «decodageBase5.py » qui permet de décoder un nombre écrit en base 5 et le tester :

```
chaine=input("Entrez le nombre à convertir : ")
N=0
for car in chaine :
   N = 5 * N + int(car)
print(N)
```

2. a. Dans un fichier « decodageBinaire.py » modifier le programme pour qu'il permette la lecture d'un nombre écrit en binaire.
b. Dans un fichier « decodageGeneral.py », écrire une version plus générale du programme qui permette à l'utilisateur de choisir la base de décodage (entre 2 et 10).
3. Écrire une dernière version de ce programme qui permet de convertir des « nombres à virgules » écrit dans une base comprise entre 2 et 10.