Représentation des entiers naturels positifs

Sources : éditions Ellipses " NSI " de Cécile Canu et " NSI " avec Thibault Balabonski,
site web de José Delamare (lycée Blaise Pascal Rouen)

Introduction

En mathématique on utilise des nombres constitués de chiffres. Ces nombres ne sont pas de natures identiques : entier naturel et relatif, réel, rationnel, pi.
En informatique cette distinction est nécessaire car les nombres de natures différentes sont définis par des objets de types différents. Un type entier permet de coder les nombres entiers, un type flottant permet de coder les nombres réels.
Nous allons d'abord apprendre à coder des nombres entiers dans différentes représentations utilisées en informatique, le binaire et l'hexadécimale.

1 - Des représentations différentes pour un même entier.

En représentation décimale, les chiffres sont les signes 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9. D'autres représentations des nombres sont possibles.
La représentation binaire adopte un sous-ensemble des chiffres précédents dans lequel les seuls signes utilisés pour représenter des entiers sont 0 et 1.
La représentation hexadécimale adopte un sur-ensemble des chiffres précédents avec en plus les signes A, B, C, D, E et F.
Enfin, d'autres représentations sont encore possibles, avec d'autres signes comme par exemple les chiffres romains, les sinogrammes, les cunéiformes, etc.

Représentations	Signes utilisés	Un même entier
Binaire	0, 1	1111011
Ternaire	0, 1, 2	11120
Octale	0, 1, 2, 3, 4, 5, 6, 7	173
Décimale	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	123
Hexadécimale	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	7B

2 - Représentation décimale ou écriture en base 10.

Un nombre en base 10 est une séquence de chiffres entre 0 et 9 ; exemple, le nombre 61027.
Pour calculer la valeur de ce nombre on affecte à chaque chiffre un poids qui est une puissance de 10 et qui dépend de sa position dans la séquence. Puis on calcule la somme de ses poids. Voici un exemple plus concret avec le nombre 61027

Case	Dizaines de milliers	Milliers	Centaines	Dizaines	Unités
Séquence	6	1	0	2	7
Position	4	3	2	1	0
Poids	10^4 10000	10^3 1000	10^2 100	10^1 10	10^0 1

La valeur de la séquence est l'entier n calculé de la manière suivante :

$n = 6 \times 10^4 + 1 \times 10^3 + 0 \times 10^2 + 2 \times 10^1 + 7 \times 10^0 = 60\,000 + 1\,000 + 0 + 20 + 7 = 61027$

Remarque.
La position des chiffres dans la séquence commence à 0.
Terminologie.
Dans une séquence comme 61027, le chiffre 6 en position la plus élevée (ici 4) est dit de poids fort et le chiffre 7 en position 0 est dit de poids faible.

3 - Représentation binaire ou écriture en base 2 et conversion.

Un nombre en base 2 est une séquence de deux chiffres, des 0 et des 1, exemple le nombre 11010010. Mais que vaut ce nombre ?

a) Convertir un nombre binaire en nombre décimal.
Pour calculer la valeur de ce nombre on affecte à chaque chiffre un poids qui est une puissance de 2 et qui dépend de sa position dans la séquence. Puis on calcule la somme de ses poids. Voici un exemple plus concret avec le nombre 11010010 (un, un, zéro, un, zéro, zéro, un, zéro).

Séquence	1	1	0	1	0	0	1	0
Position	7	6	5	4	3	2	1	0
Poids	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Calcul	1 x 128	1 x 64	0 x 32	1 x 16	0 x 8	0 x 4	1 x 2	0 x 1
Résultats	128	64	0	16	0	0	2	0
Somme	128 + 64 + 16 + 2 = 210							

Ce nombre 11010010 peut être vu comme un entier composé de 8 chiffres binaires et calculé de la manière suivante :

$M = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
$M = 128 + 64 + 0 + 16 + 0 + 0 + 2 + 0 = 210$
11010010 en base 2 correspond à 210 en base 10.

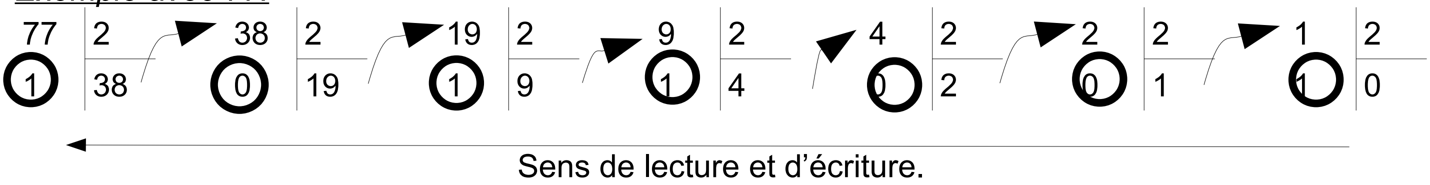
$(11010010)_2 \leftrightarrow (210)_{10}$

Exercices.
Convertir le nombre binaire 01101010 en nombre décimal. **2 + 8 + 32 + 64 = 106**
Convertir 1010101 de la base 2 à la base 10. **1 + 4 + 16 + 64 = 85**

b) Convertir un nombre décimal en nombre binaire.

Méthode.
On divise par 2 le nombre donné, on note le quotient entier et le reste qui est 0 ou 1. Puis on divise le quotient par 2, on note le reste (0 ou 1), etc,. On s'arrête une fois que le quotient devient 0.

Exemple avec 77.



77 base 10 ↔ 1001101 base 2
autres façons d'écrire : $(77)_{10} = (1001101)_2$ ou bien $\overline{77}^{10} = \overline{1001101}^{2}$

Exercices.
Écrire 25 en binaire.
25 : 2 = 12 + 1
12 : 2 = 6 + 0
6 : 2 = 3 + 0
3 : 2 = 1 + 1
1 : 2 = 0 + 1

$(25)_{10}$ < - - > $(11001)_2$

Convertir 135 de la base 10 à la base 2.
135 : 2 = 67 + 1
67 : 2 = 33 + 1
33 : 2 = 16 + 1
16 : 2 = 8 + 0
8 : 2 = 4 + 0
4 : 2 = 2 + 0
2 : 2 = 1 + 0
1 : 2 = 0 + 1

$(135)_{10}$ < – > $(10000111)_2$

Écrire $(255)_{10}$ en base 2.
255 : 2 = 127 + 1
127 : 2 = 63 + 1
63 : 2 = 31 + 1
31 : 2 = 15 + 1
15 : 2 = 7 + 1
7 : 2 = 3 + 1
3 : 2 = 1 + 1
1 : 2 = 0 + 1

$(255)_{10}$ < – > $(11111111)_2$

Il est possible de faire des opérations élémentaires avec des nombre binaires.

c) Addition en base 2
Pour additionner en binaire il faut retenir que : 0 + 0 = 0 ; 0 + 1 = 1 ; 1 + 0 = 1 ; 1 + 1 = 10.

```
exemple       1 0 1 0 1 0 0
            +   1 1 1 0 1 0
            ---------------
            1 0 0 0 1 1 1 0
```

Exercices. Opérations binaires

Additionner les deux nombres binaires suivants 1111 et 1 puis vérifier que le résultat en base 10 est juste. **1111 + 1 = 10000 < – > 15 + 1 = 16**
Réaliser les additions binaires suivantes :
$00001101_2 + 00000010_2$ **= 00001111 < – > 13 + 2 = 15**
$00101001_2 + 00000101_2$ **= 00101110 < – > 41 + 5 = 46**
$11101010_2 + 00010101_2$ **= 11111111 < – > 234 + 21 = 255**

d) Produit en base 2.
La multiplication en base 2 s'effectue comme la multiplication en base 10 (avec un décalage des différents produits).

La table de multiplication s'écrit ainsi : 0 x 0 = 0 ; 0 x 1 = 0 ; 1 x 0 = 0 ; 1 x 1 = 1.
exemple.

```
          1 1 0 1
        x 0 1 1 0
       ----------
          0 0 0 0
        1 1 0 1
      1 1 0 1
  + 0 0 0 0
 ----------------
    1 0 0 1 1 1 0
```

Exercices. Opérations binaires

Calculer le produit de $(1011)_2$ par $(0111)_2$ et vérifier le résultat en décimale.
1011 * 0111 = 1001101 < – > 11 * 7 = 77

Réaliser les multiplications binaires suivantes :

10110 x 111 **= 10011010 < – > 22 * 7 = 154**

11110 x 101 **= 10010110 < – > 30 * 5 = 150**

1001101 x 110101 **= 111111110001 < – > 77 * 53 = 4081**

4 - Représentation hexadécimale ou écriture en base 16 et conversion.

En base seize, on a besoin de 16 chiffres : 0, 1, 2, 3, 4, 5 , 6, 7, 8, 9, puis A (dix), B (onze), C (douze), D (treize), E (quatorze) et F (quinze).
Ce n'est pas une représentation de ce qu'il se passe à l'intérieur de l'ordinateur, mais plutôt de représenter plus simplement des nombres binaires.

a) Convertir un nombre décimal en nombre hexadécimal.

Méthode.
On divise par 16 le nombre donné, on note le quotient entier et le reste qui va de 0 à 15. Puis on divise le quotient par 16, on note le reste (0à 15), etc,. On s'arrête une fois que le quotient devient 0.
Quand le reste vaut 10 on écrit A, quand le reste vaut 11 on écrit B et ainsi jusqu'à F pour 15.

Exemple avec 77.

77	**16**	
- 64	**4**	**16**
13	- 0	**0**
	4	

Le reste 13 va être représenté par D et le reste 4 par 4. Ainsi 77 en base dix représente 4D en base seize.

$(77)_{10} = (4D)_{16}$

Exercices.
Convertir 25 et 125 de la base décimale en base seize.

$(25)_{10} = (10)_{16}$ **$(125)_{10} = (7D)_{16}$**

b) Convertir un nombre hexadécimal en nombre décimal.
Comme pour les nombres binaires on affecte une puissance ici de 16 au chiffre en fonction de sa position dans le nombre.

Exemple.
Convertir 1E2 en décimal.

Séquence	1	E	2
Position	2	1	0
Poids	$16^2 = 256$	$16^1 = 16$	$16^0 = 1$
Calcul	1 x 256	(E) 14 x 16	2 x 1
Résultats	256	224	2
Somme	256 + 224 + 2 = 482		

Exercices.
Convertir 128 de hexadécimal à décimal. Idem avec $F1_{16}$ puis $1A_{16}$.

$(128)_{16} = (296)_{10}$ **$(F1)_{16} = (241)_{10}$** **$(1A)_{16} = (26)_{10}$**

c) Convertir un nombre binaire en nombre hexadécimal.

Méthode.
La conversion se fait en deux étapes :
1. on décompose le nombre binaire en blocs de 4 bits : 100 1101
2. on convertit chaque bloc de 4 bits en caractères hexadécimal grâce à la table suivante :

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

$100_2 = 4_{16}$ $1101_2 = D_{16}$ donc $1001101_2 = 4D_{16}$

Exercices.
Convertir 1010101_2 en base hexadécimale. **55**
Convertir 255_{10} en base 16 puis en base 2. **$(255)_{10} = (FF)_{16} = (11111111)_2$**
Prévoir l'écriture de 256_{10} en base 16 et en base 2. **$(256)_{10} = (100)_{16} = (100000000)_2$**
Représenter le nombre 16_{10} en base 2 et en base 16. **$(16)_{10} = (10000)_2 = (10)_{16}$**

5 - Les unités de mesures.

Il est très courant en informatique de mesurer la capacité d'un disque dur, de la RAM d'un ordinateur ou d'un débit de données internet avec une unité de mesure exprimée comme un multiple d'octets. Ces multiples sont traditionnellement des puissances de 10 et on utilise les préfixes " kilo ", " mega ", etc. pour les nommer.
Voici quelques unités :

Nom	Symbole	Valeur (octets)	Puissance de 10	En octets
Kilooctet	Ko	1 000	10^3 o	10^3
Megaoctet	Mo	1 000 000	10^3 ko	10^6
Gigaoctet	Go	1 000 000 000	10^3 Mo	10^9
Teraoctet	To	1 000 000 000 000	10^3 Go	10^{12}
Pétaoctets	Po	1 000 000 000 000 000	10^3 To	10^{15}
Exaoctets	Eo	1 000 000 000 000 000 000	10^3 Po	10^{18}

Historiquement les multiples utilisés en informatique étaient des puissances de deux. Pour ne pas confondre l'ancienne et la nouvelle notation, on utilise des symboles différents pour représenter ces multiples.

Symbole	Valeur	Valeur en octets	Nombre d'octets
Kio	2^{10} octets	2^{10}	1 024
Mio	2^{10} Kio	2^{20}	1 048 576
Gio	2^{10} Mio	2^{30}	1 073 741 824
Tio	2^{10} Gio	2^{40}	1 099 511 627 776

6 - Longueurs des mots et valeurs correspondantes.

Les mots utilisés sont sur un, deux, quatre ou huit octets, soit 8, 16, 32 ou 64 bits en fonction de la machine.
Quel est le plus grand nombre entier que l'on peut représenter avec 8, 16, 32 et 64 bits ?
À compléter.

Longueur de mots (bits)	Nombre d'octets	Mot le plus long en binaire : de 0 à … (écrire par " paquets " de 4 bits)	En puissance de 2	En valeur base 10
8	**1**	**1111 1111**	**$2^8 - 1$**	**255**
16	**2**	**1111 1111 1111 1111**	**$2^{16} - 1$**	**65 535**
32	**4**	**1111 1111 1111 1111 1111 1111 1111 1111**	**$2^{32} - 1$**	**4 294 967 295**
64	**8**	**1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111**	**$2^{64} - 1$**	**18 446 744 073 709 551 615**

7 - Évaluer le nombre de bits nécessaires à l'écriture en base 2 d'un entier.

Exercices.
Quelle machine (8, 16, 32 ou 64 bits) est nécessaire pour :
- écrire 512 en binaire ? **16 bits 0000 0010 0000 0000**
- écrire 100 000 $_{10}$ en binaire ? **32 bits 0000 0000 0000 0001 1000 0110 1010 0000**
- Et pour faire des calculs sur la population mondiale ? (7 721 000 000)

64 bits 0000 0000 0000 0000 0000 0000 0000 0001 1100 1100 0011 0101 0001 1100 0100 0000

Je veux écrire 135 en binaire sur plusieurs longueur de mots. Sachant que si c'est possible, on ajoute des 0 pour aller jusqu'à n bits, écrire 135 en binaire sur :
- 8 bits : **1000 0111**
- 2 octets : **0000 0000 1000 0111**
- 4 octets : **0000 0000 0000 0000 0000 0000 1000 0111**
- et 4 bits : **pas possible**

Est-il possible d'afficher la somme de $(135)_{10}$ et $(100)_{10}$ en binaire si nous disposons d'un octet pour écrire les nombres entiers ? **Oui car 135 + 100 = 235 < 255 qui tient sur un octet.**

Et la somme de $(135)_{10}$ et $(135)_{10}$? **135 + 135 = 270 > 255 donc pas possible sur un octet**

Je veux calculer le produit $(1110)_2$ * $(110)_2$. Combien d'octets sont nécessaires ? **1 octet**

$(1110)_2 = (14)_{10}$ et $(110)_2 = (6)_{10}$; 14*6 = 84 < 255 donc un octet suffit.

8 - Conversions en python

Exercices.
Écrire une fonction qui prend en argument un nombre en base 10 et retourne le nombre en base 2.