Représentation des entiers

1 Introduction

Dans un ordinateur, toutes les informations (données ou programmes) sont représentées à l'aide de deux chiffres 0 et 1, appelés chiffres binaires ou bits (de l'anglais *binary digits*).

Dans la mémoire d'un ordinateur (RAM, ROM, registres des microprocesseurs, etc.), ces chiffres binaires sont regroupés en octets (c'est-à-dire par « paquets » de 8, qu'on appelle des *bytes* en anglais) puis organisés en mots machine de 2, 4 ou 8 octets pour les machines les plus courantes. Par exemple, une machine dite de 32 bits est un ordinateur qui manipule directement des mots de 4 octets lorsqu'il effectue des opérations.

Ce regroupement de bits en octets ou mots machine permet de représenter d'autres données que des 0 et des 1, comme par exemple des nombres entiers, des (approximations de) nombres réels, des caractères.

2 Représentation des entiers naturels

2.1 Représentation en base 10

Le principe de la numération en base 10 (numération décimale) que nous utilisons quotidiennement est de regrouper ce que l'on compte par paquets de 10. Ainsi, en base 10, les nombres sont représentés en utilisant des « colonnes » qui représentent les puissances successives de 10.

Exemple 1

Le nombre 203 est égal à :

$$2 \times 10^2 + 0 \times 10^2 + 3 \times 10^0.$$

$\times 10^2$	$\times 10^1$	$\times 10^0$
2	0	3

2.2 Représentation en base 2

La numération en base 2 fonctionne de la même façon mais on fait des paquets dès que l'on a 2 éléments.

Remarque 1

En base 2 :

— on n'utilise donc que deux chiffres : 0 et 1 ;

— au lieu des puissances successives de 10, on travaille avec des puissances successives de 2.

Notation 1

Pour indiquer qu'un nombre est écrit en base 2, on le notera de la façon suivante : $\overline{1001}^2$.

Méthode 1 : passer de la base 2 à la base 10

Pour passer de la base 2 à la base 10, on peut utiliser un « tableau » :

$\times 2^3$	$\times 2^2$	$\times 2^1$	$\times 2^0$
1	0	0	1

Le nombre $\overline{1001}^2$ est donc égal à :

$1\times 2^3+0\times 2^2+0\times 2^1+1\times 2^0 = 8+0+0+1 = 9.$

On écrira $\overline{1001}^2 = 9$.

Exercice 1 : passer de la base 2 à la base 10

1. Déterminer la représentation en base 10 de $\overline{1111\,1111}^2$.
2. Déterminer la représentation en base 10 de $\overline{1001\,0110}^2$.
3. Isaac Asimov est né en $\overline{111\,1000\,0000}$. Écrire cette année de naissance en base 10.
4. (a) En écrivant `0b11111111` en Python, on peut vérifier le résultat de la question 1. Le faire.
   (b) De même, vérifier le résultat des questions 2. et 3.

Exercice 2 : ordres de grandeurs

Compléter le tableau suivant :

Avec des mots de :	on peut représenter les entiers naturels compris entre :
8 bits	
16 bits	
32 bits	
64 bits	

Exercice 3 : un peu de Python

1. Écrire le calcul permettant d'obtenir la représentation en base 10 de $\overline{1011}^2$.
2. Compléter la fonction ci-dessous afin qu'elle permette la conversion de la base 2 vers la base 10 :

```
def bin2dec(n):
    """ Permet la conversion de la base $2$ vers la base $10$
        Entrée : une chaine de caractères n qui donne l'écriture
                 d'un nombre en base 2
        Sortie : le nombre en base 10
    """
    longueur = len(n)
    decimal = 0
    for i in range(longueur):
        pass
```

Lien Basthon – Corrigé

Méthode 2 : passer de la base 10 à la base 2

Pour passer de la base 10 à la base 2, on effectue des divisions successives par 2. Par exemple :

$$42 = \overline{101010}^{2}.$$

Retrouver l'explication détaillée ici (lien PeerTube).

Exercice 4 : passer de la base 10 à la base 2

1. Déterminer la représentation en base 2 du nombre 436.
2. Déterminer la représentation en base 2 du nombre 1000.
3. Que fait la fonction `bin` de Python ?

Exercice 5 : un peu de Python

1. Que fait la fonction `str` de Python ?
2. Compléter la fonction suivante afin qu'elle permette la conversion de la base 10 vers la base 2 :

```
def dec2bin(n):
    """ Permet la conversion de la base $10$ vers la base $2$
        Entrée : un entier naturel n
        Sortie : une chaine de caractères qui donne l'écriture
                 de n en base 2
    """
    if n == 0:
        return "0"
    else:
        b = ""
        while n != 0:
            pass
```

Lien Basthon – Corrigé

2.3 Représentation en base 16

Pour écrire en base 16, on a besoin de 16 « chiffres » notés : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (dix), B (onze), C (douze), D (treize), E (quatorze) et F (quinze).

Méthode 3 : passer de la base 16 à la base 10

Pour passer de la base 16 à la base 10, on peut utiliser un « tableau » :

$\times 16^3$	$\times 16^2$	$\times 16^1$	$\times 16^0$
2	9	A	F

Le nombre $\overline{29AF}^{16}$ est donc égal à :

$2\times 16^3+9\times 16^2+10\times 16^1+15\times 16^0 = 10\,671.$

On a donc $\overline{29AF}^{16} = 10\,671$.

Exercice 6 : passer de la base 16 à la base 10

1. Déterminer la représentation en base 10 du nombre $\overline{23A}^{16}$.
2. Déterminer la représentation en base 10 du nombre $\overline{FABE51}^{16}$.
3. Vérifer la première réponse grâce à l'instruction `0h23A`.
4. De la même façon, vérifier la deuxième réponse.

Méthode 4 : passer de la base 10 à la base 16

Pour passer de la base 10 à la base 16, on effectue des divisions successives par 16. Par exemple :

$$315 = \overline{13B}^{16}.$$

Retrouver l'explication détaillée ici (lien PeerTube).

Exercice 7 : passer de la base 10 à la base 16

1. Donner la représentation en base 16 du nombre 324.
2. Donner la représentation en base 16 du nombre 56 026.
3. L'instruction `hex(324)` permet de vérifier la première réponse. Vérifier vos réponses.

2.4 Liens entre la base 2 et la base 16

Méthode 5 : passer de la base 2 à la base 16

Pour passer de la base 2 à la base 16, on groupe les bits par paquets de 4 (de la droite vers la gauche) :

$$\overline{\underbrace{1101}_{\text{D}}\ \underbrace{0110}_{6}}^{2}.$$

On a donc $\overline{1101\,0110}^{2} = \overline{\text{D6}}^{16}$.

Exercice 8 : passer de la base 2 à la base 16

Écrire les nombres $\overline{1011\,1101}^{2}$ et $\overline{1100\,1001\,1010}^{2}$ en base 16.

Méthode 6 : passer de la base 16 à la base 2

Pour passer de la base 16 à la base 2, on procède de la façon inverse :

$$\overline{\underbrace{\text{F}}_{1111}\ \underbrace{5}_{0101}}^{16}.$$

On a donc $\overline{\text{F5}}^{16} = \overline{1111\,0101}^{2}$.

Exercice 9 : passer de la base 16 à la base 2

Écrire les nombres $\overline{\text{23D5}}^{16}$ et $\overline{\text{7CF21}}^{16}$ en base 2.

Exercice 10 : adresse MAC

1. De combien de bits est constituée une adresse MAC ?
2. Consulter ce site et l'essayer avec l'adresse MAC qui vous a servi à enregistrer votre tablette sur le réseau du lycée.

3 Représentation des entiers relatifs

Question : comment représenter les entiers relatifs ? Il s'agit de trouver une représentation des entiers relatifs compatible avec la représentation des entiers naturels vue précédemment.

3.1 Un bit pour le signe

Imaginons qu'on code des nombres entiers avec un octet. Grâce à ces 8 bits, on peut coder les entiers naturels compris entre 0 et 255. Une idée est tentante :

— utiliser le premier bit pour indiquer le signe du nombre : 0 pour un nombre positif et 1 pour un nombre négatif ;

— utiliser les 7 bits restants pour écrire la valeur absolue de l'entier représenté.

Exemple 2

Avec cette approche, 17 s'écrit $\overline{0001\,0001}^2$ et -17 s'écrit $\overline{1001\,0001}^2$ (de la même façon que 17, mais avec le bit de poids fort égal à 1 au lieu de 0).

Exercice 11

En utilisant l'idée précédente, écrire, en utilisant un octet, les nombres 127, -31, -4, 3 et 0.

Remarque 2 : un premier problème

Un premier problème apparaît avec cette idée : le nombre 0 admet deux représentations différentes. Lesquelles ?

Remarque 3 : un second problème

1. Écrire 4 et 3 sur un octet puis effectuer l'addition bit à bit. Qu'obtient-on ?
2. Faire de même avec 4 et -3.
3. Décrire alors le problème rencontré.

3.2 Une autre approche : le complément à 2

Nous allons décrire une autre approche dans le cas où nous disposons d'un octet. Un octet permet de représenter 256 valeurs différentes. Avec cette approche, appelée « complément à 2 », avec un octet, on représente :

— 128 entiers strictement négatifs : -128, -127, ... , -1 ;

— 128 entiers positifs ou nuls : 0, 1, ... , 127.

Méthode 7 : complément à 2

— On représente un entier positif ou nul de la même façon que dans le paragraphe 2 :

$$42 = \overline{0010\,1010}^2.$$

— On représenter un entier strictement négatif en deux étapes. Par exemple, pour représenter -42 avec la méthode du complément à 2 :

1. On commence par inverser tous les bits de l'écriture en base 2 de 42 :

$$\overline{1101\,0101}^2.$$

2. Puis on ajoute 1 au résultat obtenu :

$$\overline{1101\,0110}^2.$$

L'entier relatif -42 s'écrit donc $\overline{1101\,0110}^2$.

Exercice 12

En utilisant la méthode du complément à 2, écrire, sur un octet, les entiers -5, -45 et -68.

Exercice 13

Vérifier que le problème de l'addition bit à bit ne se pose plus pour $4 + (-3)$.