Représentation d'un entier en base b

29 septembre 2012

1 Prérequis.

Les bases de la programmation en langage Python sont supposées avoir été travaillées.

2 Objectifs.

Thème de la représentation de l'information : introduire et travailler l'écriture en base b. Un objectif parallèle : travailler sur de courts algorithmes et programmes python.
Un mini-projet sera évalué pour la synthèse de ce thème : programmation (en langage python) du passage de la base shadock à la base dix et vice versa.

Résumé

Comprendre l'humour geek : « Dans le monde, il y a 10 sortes de personnes : ceux qui comprennent le binaire, et les autres ».

3 0 et 1

La mémoire d'un ordinateur est constituée d'une multitude de petits circuits électroniques. Chacun de ces circuits ne peut prendre que deux états. On associe traditionnellement l'un des états à 0 et l'autre à 1. De ce fait toute information doit être traduite dans un ordinateur uniquement par des 0 et des 1.

bit

bit est l'abréviation de BInary Digit (chiffre binaire).

octet

Un octet (en anglais : byte) est une suite de 8 bits.

mot

L'état d'un circuit, composé de plusieurs circuits mémoire-un-bit, se décrit par une suite finie de 0 et de 1 qu'on appelle mot.

Exercice 1
Si la mémoire d'un ordinateur était constituée de dix circuits à mémoire-un-bit (autrement dit : par des mots de 10 bits), quel serait le nombre d'états possibles de la mémoire de cet ordinateur ? Et avec 1 milliard de circuits ?

Une résolution
Pour l'évolution du nombre de transistors dans un processeur : `http://fr.wikipedia.org/wiki/Transistor`. □

Exercice 2
On veut représenter les 7 couleurs de l'arc en ciel par un mot, les sept mots devant être distincts et de même longueur (en bits). Quelle est la longueur minimale de ces mots ?

Une résolution
Avec un bit, on peut enregistrer deux informations différentes : 0 ou 1. Avec deux bits, on forme 4 mots différents : 00, 01, 10, 11. Avec trois bits, on forme 2^3 mots différents : 000, 001, 010, 011, 100, 101, 110, 111.
Il faut donc au moins une longueur 3. Mais on doit aussi pouvoir coder la nature de l'information transmise pour que le logiciel utilisant cette information sache ici que les mots concernés représentent des couleurs...

4 Exprimer un entier en base b.

On rappelle que l'écriture usuelle pour les entiers est une écriture décimale ou écriture en base dix, ce qui signifie que l'on exprime les entiers sous la forme d'une combinaison linéaire de puissances de dix, les coefficients de la combinaison étant des nombres entiers entre 0 et 9 (ce sont les chiffres de l'entier).
Exemple : $5489 = 5\times10^3+4\times10^2+8\times10^1+9\times10^0$.
Si l'écriture 5489 était l'écriture d'un entier en base seize, 5489 désignerait l'entier $5\times16^3+4\times16^2+8\times16^1+9\times16^0$.
Pour distinguer 5489 base dix et 5489 base seize, on peut utiliser certaines conventions de notations. Par exemple : $5489_{(16)}$ (ou 5489_{seize}) signifiera que l'on utilise la base seize.
Ainsi $5489_{\text{seize}} = 21641_{\text{dix}}$.
Lorsque la base n'est pas précisée, on conviendra qu'il s'agit de la base 10.
D'autres conventions peuvent être utilisées dans les livres. Dans les langages informatiques, des conventions sont également choisies, ces conventions ne sont pas nécessairement les mêmes dans les divers langages.
Par exemple, dans le langage Python, $5489_{(16)}$ sera noté 0x5489 (les chiffres sont précédés du chiffre 0 et de la lettre x minuscule). De même, les chiffres d'un entier écrit en base deux seront précédés de 0b et les chiffres d'un entier écrit en base huit seront précédés de 0o.

A savoir

Soit b un entier au moins égal à 2. Tout entier naturel n peut s'écrire sous la forme d'une somme de termes de la forme $x_i b^i$ où les nombres i sont des entiers naturels et les nombres x_i sont des entiers naturels compris entre 0 et $b-1$. Les entiers entre 0 et $b-1$ constituent les chiffres de la base b.

Ainsi les chiffres des écritures en base 2 sont 0 et 1.
Les chiffres de la base 5 sont 0, 1, 2, 3, 4.
Les chiffres de la base 10 sont 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Exercice 3
En base dix, pour décrire l'entier 4758, on peut écrire : 8 unités, 5 dizaines, 7 centaines et 4 milliers.
En base deux, pour décrire l'entier 1101, on pourra écrire : 1 unité, 0 deuzaine, 1 quatraine, 1 huitaine.
Expliquer les mots choisis.

Exercice 4
Donner en base dix les entiers 101_{deux}, 1110_{deux}.

Exercice 5
1. Quels entiers naturels peut-on écrire en binaire sur un octet ?

2. Quels entiers naturels peut-on écrire en binaire sur un mot de 32 bits ?
3. Quels entiers naturels peut-on écrire en binaire sur un mot de 64 bits ?

Exercice 6
Le tableau ci-dessous présente les premiers entiers en base 4. Compléter.

0	1	2	4
10	11	12	13
20	21		

Une résolution
Tableau.

0	1	2	4
10	11	12	13
20	21	22	23
30	31	32	33
100	101	102	103
110	111	112	113
120	121	122	123

□

Exercice 7
Les chiffres en base 16 sont habituellement notés 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Expliquer pourquoi on utilise de nouveaux symboles pour désigner 10, 11, 12, 13, 14, 15 dans cette base.

Une résolution
L'objectif est de faire comprendre que 10 deviendrait ambigu. $10_{(16)} = 16_{(10)}$ car $10_{(16)} = 1 \times 16 + 0$. □

Définition

Écrire un entier naturel en base b, c'est expliciter les entiers x_i de la propriété précédente.

5 De la base b à la base 10.

Exercice 8
Traduire en base 10 les entiers suivants :
$a_1 = 1A2_{(16)}$, $a_2 = 431_{(5)}$, $a_3 = 10_{(2)}$, $a_4 = 10_{(5)}$, $a_5 = 10_{(b)}$ où $b \in \mathbb{N} - \{0;1\}$, $a_6 = 1011_{(2)}$.

Une résolution
Les réponses sont 418 ; 116 ; 2 ; 5 ; expression de b en base b : 10 ; 11.
On peut les obtenir avec python avec le code suivant :

Python

```
print int('1A2',16)
print int('431',5)
print int('10',2)
print int('10',5)
print int('1011',2)
```

ou encore :

Python

```
print 0x1A2
print int('431',5)
print 0b10
print int('10',5)
print 0b1011
```

□

Exercice 9

1. La fonction ord du langage python renvoie le code ascii (exprimé en décimal) du caractère donné en paramètre. L'instruction ord('a') renvoie par exemple 97, ord('A') renvoie 65.
   Qu'obtient-on avec les instructions ord('0')−ord('0'), ord('1')−ord('0'), ord('2')−ord('0'), ..., ord('9')-ord('0') ?
2. Définir une fonction en langage python :
   – Input : une chaîne de caractères dont les caractères sont les chiffres d'un entier exprimé en base deux.
   – Output : affichage de cet entier en base 10.
   En d'autres termes, on demande d'écrire une fonction ayant le même effet que l'instruction int('1101',2) à l'aide d'une boucle ('1101' n'étant ici évidemment qu'une valeur possible du paramètre parmi tant d'autres).

Une résolution

1. On obtient 0, 1, 2, ..., 9. Ce qui signifie que les chiffres 0, 1, 2, ..., 9 ont des numéros consécutifs dans le code ascii.
2. Il y a un certain nombre de possibilités.
   Première solution :

   Python

   ```
   def f(n):
       s=0
       lg=len(n)
       for j in range(0,lg):
           s+=(ord(n[lg-j-1])-ord('0'))*2**j
       return s

   print f('11')
   ```

   ou (même programmation, variante dans l'utilisation des fonctions de base de python) :

   Python

   ```
   def f2(n):
       s=0
       lg=len(n)
       for j in range(0,lg):
           s+=int(n[lg-j-1])*2**j
       return s
   ```

On pourrait amener les élèves à réfléchir sur le nombre d'opérations avec une écriture telle que : $\overline{abcd}_{(2)} = d + 2 \times (c + 2 \times (b + 2a))$. Ce qui donnerait par exemple :

Python

```
def g(n):
    s=0
    while n!='':
        s*=2
        s+=ord(n[0])-ord('0')
        n=n[1:]
    return s
```

□

Exercice 10

Modifier la fonction précédente pour obtenir :

- Input : un entier b compris entre 2 et 9, une chaîne de caractères dont les caractères sont les chiffres d'un entier n exprimé en base b.
- Output : affichage de l'entier n en base 10.

Python

```
def h(b,n):
    s=0
    while n!='':
        s*=b
        s+=ord(n[0])-ord('0')
        n=n[1:]
    return s

print h(5,'41')
```

□

Savoir Faire

Passer de l'écriture en base b d'un entier à l'écriture en base 10.

6 De la base 10 à la base b

6.1 Recherche du plus grand exposant de b

Exercice 11

Soit n un entier naturel et $b \geqslant 2$ un entier. Soit j le plus petit entier des entiers k vérifiant $n < b^k$.

1. Expliquer pourquoi $b^{j-1} \leqslant n$.
2. Quelle autre définition de j peut-on donner en utilisant l'écriture de n en base b ?

Une résolution

1. j est le plus **petit** tel que ...
2. $b^{j-1} \leqslant n < b^j$. Donc dans l'écriture de n en base b, on aura des coefficients 0 pour b^i avec $i \geqslant j$ et l'écriture de n en base b comporte exactement j chiffres (coefficients de b^0, b^1, b^2, ..., b^{j-1}). □

Fonction logarithme

Vous découvrirez et utiliserez la fonction logarithme népérien dans les autres disciplines scientifiques. Cette fonction est notée ln sur vos calculatrices. On définit la fonction logarithme à base b (où b est un réel strictement positif) par $\log_b(x) = \frac{\ln(x)}{\ln(b)}$ pour tout réel $x > 0$.

A savoir

Le nombre de chiffres dans l'écriture en base b d'un entier naturel n est égal à PartieEntière$(\log_b(n)) + 1$.

Exercice 12

1. Écrire une fonction python (sans utiliser la fonction log) :
   - input : un entier $b \geqslant 2$ et un entier naturel n (tous deux exprimés en base 10).
   - output : le nombre de chiffres de l'écriture de l'entier n en base b.
2. Même question en utilisant cette fois la fonction logarithme du langage python.

Une résolution

Par exemple :

Python

```python
# -*- coding: utf-8 -*-

import math

def nbc(b,n):
    j=0
    while b**j<=n:
        j+=1
    return j

b=5
n=87945
print nbc(b,n)
print int(math.log(n,b))+1
```

Pour la partie entière, on peut aussi utiliser math.floor.

Exercice 13

Soit n un entier naturel donné en base 10. Pour écrire cet entier en base b, on peut procéder ainsi : enlever la plus grande puissance de b possible... et recommencer

1. Écrire en base 5 l'entier $n = 327$.
2. Chercher sur le web la base shadok et écrire en base shadok l'entier $n = 3083$. Vous pouvez par exemple visionner : `http://www.youtube.com/watch?feature=player_embedded&v=M_SigFQfKBw`.

Une résolution

1. $327 = 2 \times 5^3 + 3 * 5^2 + 0 \times 5 + 2 = 2302_{(5)}$.
   Les étapes : 327, on enlève 5^3 une première fois, on obtient 202. On enlève à nouveau 5^3, ce qui donne 77. On ne peut plus enlever 5^3, d'où le chiffre de gauche dans l'écriture en base 5 ...
2. La base shadok est une base 4 : $3083 = 300023_{(4)}$ =MEUGAGAGAZOMEU. Voir le convertisseur : `http://www.dcode.fr/shadoks-ga-bu-zo-meu`. □

Remarque. Quelques cas particuliers en python à tester :

Python

```
print oct(65) # vers la base 8
print hex(65) # vers la base 16
print bin(65) # vers la base 2
```

6.2 La méthode des divisions en cascade

A savoir

Pour tout couple d'entiers naturels $(a;b)$ avec $a \in \mathbb{N}$ et $b \in \mathbb{N} - \{0\}$, il existe un unique couple $(q;r)$ d'entiers tels $a = bq + r$ et $0 \leqslant r \leqslant b-1$. q est appelé quotient de la division euclidienne (on parle aussi de division entière) de a par b et r reste de cette division.

Exemple : $34 = 9 \times 3 + 7$, le reste de la division euclidienne de 34 par 9 est donc 7. Par contre le reste de la division euclidienne de 34 par 3 n'est pas 7 (car le reste dans une division euclidienne par 3 ne peut valoir que 0, 1 ou 2).
On rappelle qu'en python le reste de la division euclidienne de a (de type int) par b (de type int) est obtenu par l'instruction $a\%b$ et le quotient de a par b par $a//b$ ou $\text{int}(a/b)$.

Pour expliquer le principe de l'algorithme des divisions en cascade, traitons un exemple générique en base $b = 5$.
Soit $q_0 = 3412_{(5)} = 3 \times 5^3 + 4 \times 5^2 + 1 \times 5 + 2$.
On peut écrire : $q_0 = \left(3 \times 5^2 + 4 \times 5^1 + 1\right) \times 5 + 2$. On a ainsi écrit q_0 sous la forme $5q_1 + r_0$ où $r_0 = 2$ et $q_1 = 3 \times 5^2 + 4 \times 5^1 + 1$.
Comme r_0 est un chiffre de l'écriture en base 5, il est compris entre 0 et 4 et l'écriture $5q_1 + r_0$ permet alors d'affirmer que r_0 est le reste de la division euclidienne de q_0 par b.
En généralisant ce raisonnement, on constate :

A savoir

Le chiffre des unités (c'est à dire le coefficient de b^0) de l'entier n en base b est égal au reste de la division euclidienne de n par b.

Considérons maintenant l'entier $q_1 = 3 \times 5^2 + 4 \times 5^1 + 1$ (c'est à dire le quotient de la division de q_0 par b).
On peut, sur cet entier q_1, recommencer le même traitement que sur l'entier q_0 : $q_1 = (3 \times 5 + 4) \times 5 + 1$.
Ainsi le chiffre des unités de q_1 en base b est le reste de la division entière de q_1 par b. Et si l'on reporte l'écriture précédente de q_1 dans l'écriture de q_0, on a : $q_0 = ((3 \times 5 + 4) \times 5 + 1) \times 5 + r_0$, on constate que ce chiffre est le coefficient de b^1 dans l'écriture de q_0.
En poursuivant ce raisonnement, on obtient l'algorithme des divisions en cascade :

A savoir

Soit $b \geqslant 2$ un entier et $n \in \mathbb{N}$.

1. On calcule le reste r_0 de la division entière de n par b et le quotient q_1.
2. On calcule le reste r_1 de la division entière de q_1 par b et le quotient q_2.
3. On calcule le reste r_2 de la division entière de q_2 par b et le quotient q_3.
4. ...
5. Et ainsi de suite jusqu'à obtenir un quotient nul q_j.

Alors $n = r_j b^j + r_{j-1} b^{j-1} + r_{j-2} b^{j-2} + r_{j-3} b^{j-3} + \cdots + r_0$. En d'autres termes les restes successifs sont les chiffres de l'écriture de n en base b, ce que l'on écrit parfois sous la forme $n = \overline{r_j r_{j-1} r_{j-2} \ldots r_1 r_0}_{(b)}$.

En d'autres termes, on dispose devant soi de n allumettes : on les regroupe par paquets de cinq, il reste 0 ou 1 ou 2 ou 3 ou 4 allumettes à côté des paquets de cinq : c'est le chiffre de droite (chiffre des unités) dans l'écriture de l'entier n en base cinq. On regroupe ensuite les paquets de cinq (PC) en paquets de cinq (on constitue donc des paquets de cinq paquets PC, appelons PCC ces paquets). Il reste 0 ou 1 ou 2 ou 3 ou 4 paquets PC à côté des PCC : c'est le second chiffre en partant de la droite dans l'écriture de l'entier n en base cinq. Et on continue ainsi : on constitue ensuite des PCCC (paquets de cinq paquets PCC)…

Exercice 14

1. Écrire l'entier $n = 548$ en base 7 à l'aide de l'algorithme des divisions en cascade.
2. Écrire l'entier $n = 548$ en base 5 à l'aide de l'algorithme des divisions en cascade.
3. Si l'on applique l'algorithme des divisions en cascade à l'entier $n = 548$ et $b = 10$, qu'obtient-on ?

Une résolution

$548 = 1412_{(7)}$ et $548 = 4143_{(5)}$. □

Exercice 15

Écrire en python, en utilisant le principe des divisions en cascade :
– input : un entier b au moins égal à 2 (et au plus égal à 9 pour simplifier la tâche) et un entier n écrit en base 10.
– output : une écriture de n en base b.

Une résolution

Par exemple :

Python

```python
# -*- coding: utf-8 -*-

def cascade(b,n):
    """ b entier au moins égal à 2, donné en base 10
    n entier naturel donné en base 10.
    renvoie l'écriture en base b de n par divisions en cascade"""
    e=''
    while n!=0:
        r=n%b
        e=str(r)+e
        n=n//b
    return e

print cascade(5,548)
```

On peut expliquer pourquoi l'algorithme s'arrête nécessairement (suite d'entiers naturels strictement décroissante pour les quotients) et faire remarquer que c'est une preuve de l'existence d'une écriture en base b. □

Savoir Faire

Passer de l'écriture en base 10 d'un entier n à une écriture en base b.

Exercice 16

1. Donner l'écriture en base deux de l'entier 2^5.
2. Donner l'écriture en base deux de l'entier $2^5 - 1$.

Une résolution
$2^5 = 100000_{\text{deux}}$ et $2^5 - 1 = 11111_{\text{deux}}$. □

Exercice 17
Écrire les entiers de 0 à 20 en base deux (sans passer par la base dix).

Une résolution
0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, ...

Exercice 18
Justifier qu'avec des mots de n bits, on peut écrire les entiers de 0 jusqu'à $2^n - 1$.

Une résolution
Par exemple, avec $n = 8$, on écrit les entiers de 00000000 à 11111111, c'est à dire de 0 à $1 + 2 + 2^2 + \cdots + 2^7 = 2^8 - 1$. □

Exercice 19
Si l'on dispose de l'écriture d'un entier n en base dix, il suffit d'ajouter un 0 à sa droite pour avoir l'écriture de l'entier $10n$. On dispose de l'écriture d'un entier n en base b, on ajoute un 0 à la droite de cette écriture. A quelle opération cela correspond-il ?

Une résolution
Si $n = x_k b^k + x_{k-1} b^{k-1} + \cdots + x_1 b + x_0$ alors $bn = x_k b^{k+1} + x_{k-1} b^k + \cdots + x_1 b^2 + x_0 b + 0$. L'ajout d'un 0 à droite de l'écriture revient donc à multiplier par la base b.
En python, une opération de décalage des bits. Tester :

Python
```
1 a=0b101
2 a=a<<1
3 bin(a)
```

Exercice 20
1. Vérifier sur quelques exemples que pour passer d'une écriture en base 16 à une écriture en base 2, il suffit d'écrire chacun des chiffres de l'écriture en base seize en base deux (sur quatre rangs, en collant des 0 à gauche si nécessaire).
2. Cela fonctionne-t-il pour passer par exemple de la base 7 à la base 2 ?