TD10 - Ecriture binaire/héxadécimale d'une entier

MPSI-PCSI - Lycée Thiers

Ecriture d'un nombre dans une base

Rappel : Division euclidienne. $\forall\, n \in \mathbb{N},\ \forall\, m \in \mathbb{N}^*$:

$$\exists!\,(q,r) \in \mathbb{N}^2, \quad \text{tel que } \begin{cases} n = q \times m + r \\ 0 \leqslant r < m \end{cases}$$

Corollaire. *Pour tout entier $m \in \mathbb{N} \setminus \{0,1\}$, pour tout entier $n \in \mathbb{N}$, il existe une unique suite finie $(u_n)_{n\in[[0,N]]}$ d'entiers compris entre 0 et $m-1$, tels que :*

$$n = \sum_{i=0}^{N} u_i \times m^i$$

et si $N > 0$, $u_N \neq 0$. L'écriture en base m de n est $u_N\, u_{N-1} \cdots u_1\, u_0$.

Ecriture d'un nombre en base $m > 1$

Preuve. Existence : par division euclidienne $\exists\,!(q_0, u_0) \in \mathbb{N}^2$ avec $n = q_0 \times m + u_0$ et $0 \leqslant u_0 < m$. Si $q_0 = 0$ la suite u_0 convient puisque $n = u_0 \times m^0$. Si $q_0 \neq 0$ alors par division euclidienne $\exists\,!(q_1, u_1) \in \mathbb{N}^2$ avec $q_0 = q_1 \times m + u_1$ et $0 \leqslant u_1 < m$. Remarquer que $n > q_0 > q_1$ puisque $m > 1$. Ainsi en poursuivant ce procédé on construit une suite finie $(u_0, u_1, \ldots, u_N)$ avec $0 \leqslant u_i < m$, $q_N = 0$ et :

$$n = q_0 \times m + u_0 = (q_1 \times m + u_1) \times m + u_0$$

$$= ((\cdots((0 \times m + u_N) \times m + u_{N-1})\cdots) \times m + u_1) \times m + u_0 = \sum_{i=0}^{N} u_i \times m^i.$$

Unicité. Elle provient de l'unicité du quotient et du reste dans la division euclidienne : si $n = \sum_{i=0}^{N} u_i \times m^i$ alors le procédé précédent produit la suite $(u_0, u_1, \ldots, u_N)$.

Exemple : Ecriture de 72 en base 2 :

$72 = 2 \times 36 + 0 \quad u_0 = 0$
$36 = 2 \times 18 + 0 \quad u_1 = 0$
$18 = 2 \times 9 + 0 \quad u_2 = 0$
$9 = 2 \times 4 + 1 \quad u_3 = 1$
$4 = 2 \times 2 + 0 \quad u_4 = 0$
$2 = 2 \times 1 + 0 \quad u_5 = 0$
$1 = 2 \times 0 + 1 \quad u_6 = 1$

72 s'écrit 1001000_2 en base 2 (ou binaire).

En python :

```
>>> a = 72
>>> while a>0:
...     print(a%2)
...     a = a//2
...
0
0
0
1
0
0
1
```

La "boucle" `while` s'exécute tant que la condition a>0 est satisfaite.

Exemple : Ecriture de 72 en base 16 :

$72 = 16 \times 4 + 8 \quad u_0 = 8$

$4 = 16 \times 0 + 4 \quad u_1 = 4$

72 s'écrit 48_{16} ou 0x48 en base 16 (ou hexadécimal).

```
>>> a = 72
>>> while a>0:
...     print(a%16)
...     a = a//16
...
8
4
```

Pour écrire un nombre en base 16 on utilise les "chiffres" de 0 à 9 ainsi que :

A=10, B=11, C=12, D=13, E=14, F=15.

En 2 chiffres on écrit tous les nombres de 0 à $FF = 15 + 15 * 16 = 16^2 - 1 = 255$.
Autant qu'en binaire avec 8 'bits' (ou chiffre 0,1) car $11111111 = 2^8 - 1 = 255$.
Puisque $2^4 = 16$ on a la conversion binaire/hexadecimal :

$0000_2 = 0$, $0001_2 = 1$, $0010_2 = 2$, $0011_2 = 3$, $0100_2 = 4$, $0101_2 = 5$,
$0110_2 = 6$, $0111_2 = 7$, $1000_2 = 8$, $1001_2 = 9$, $1010_2 = A$, $1011_2 = B$,
$1100_2 = C$, $1101_2 = D$, $1110_2 = E$, $1111_2 = F$.

Ex : $72_{10} = 0100\,1000_2 = 4\,8_{16}$.

Voilà pourquoi un ordinateur fonctionne en 8, 16, 32 ou 64 bits
(multiples de 4... par une puissance de 2)...

Exercice 1. Enoncé

Exercice 1. Ecriture binaire d'un entier naturel.

Tout entier naturel naturel peut s'écrire en binaire. Plus précisément pour tout $n \in \mathbb{N}$, il existe une unique suite finie $(b_k)_{k\in[[0,N]]}$ tel que :

$$n = \sum_{k=0}^{N} b_k \cdot 2^k$$

avec $b_k \in \{0,1\}$ et $n > 0 \implies b_N \neq 0$. Pour $n = 0$ on prend la suite réduite à un seul élément $b_0 = 0$.

1. Ecrire une fonction `bin2dec` qui prend en paramètre une chaine de caractères `0` ou `1`, et renvoie l'entier dont c'est l'écriture binaire.
2. Ecrire une fonction `dec2bin` qui prend en paramètre une entier positif `n` et renvoie une chaine de caractères `0` et `1` représentant l'écriture binaire de `n`. Pour cela on appliquera l'algorithme :

```
k = 0
TANT QUE n > 0 :
    b_k ← n % 2
    n ← n // 2
    k ← k+1
```

Exercice 1. Corrigé

1)

```
def bin2dec(bin): # conversion binaire (str) -> décimal (int)
    n = 0
    k = 0
    for b in bin[::-1]:
        n += int(b) * 2**k
        k += 1
    return n
```

2)

```
def binaire(n): # conversion décimal (int) -> binaire (str)
    if n == 0:
        return '0'
    bin = ''
    while n > 0:
        bin = str(n%2) + bin
        n = n//2
    return bin
```

Exercice 2. Enoncé

Exercice 2. Ecriture hexadécimale d'une entier naturel. Tout entier naturel peut s'écrire en héxadécimal, c'est à dire en base 16. Plus précisément pour tout $n \in \mathbb{N}$, il existe une unique suite finie $(h_k)_{k\in[[0,N]]}$ tel que :

$$n = \sum_{k=0}^{N} h_k \cdot 16^k$$

avec $h_k \in [[0,15]]$ et $n > 0 \implies h_N \neq 0$. Pour $n = 0$ on prend la suite réduite à un seul élément $b_0 = 0$.

1. Ecrire une fonction `dec2hex` qui prend en paramètre une entier positif `n` et renvoie une chaine de caractères `0`,...,`9`, `A`,...`F` représentant l'écriture hexadécimale de `n`. Pour cela on appliquera l'algorithme :

```
k = 0
TANT QUE n > 0 :
    h_k ← n % 16
    n ← n // 16
    k ← k+1
```

et on définira comme variable globale la liste :
`H = ['A', 'B', 'C', 'D', 'E', 'F']`.

Exercice 2. Enoncé

2. Ecrire une fonction `hex2dec` qui prend en paramètre une chaine de caractères `0`,...,`9`, `A`,...`F` et renvoie l'entier ayant pour écriture hexadécimale cette chaine. Pour cela on pourra utiliser la table ASCII.

Exercice 2. Corrigé

1)

```
L = list(range(10)) + ['A','B','C','D','E','F']

def dec2hex(n): # Conversion décimal (int) -> hexadécimal (str)
    if n==0:
        return '0x0'
    hexa = ''
    while n > 0:
        r = n % 16
        hexa = str(L[r]) + hexa
        n = n//16
    return '0x' + hexa
```

Exercice 2. Corrigé

2)

```
def hex2dec(hex):
    n, k = 0, 0
    for h in hex[::-1]:
        if '0' <= h <= '9':
            n += int(h)*16**k
        elif 'A' <= h <= 'F':
            n += (ord(h)-ord('A')) * 16**k
        k += 1
    return n
```

Complément. Enoncé

Exercice 3. Complément hors-programme : Interface graphique pour l'obtention de l'écriture binaire d'un nombre.

Pour créer une interface graphique :
– Dans l'onglet Pyzo, 'Préférences', choisir pour Gui : Tk - Tk Widget Toolkit.
– Relancer Pyzo. On pourra désormais utiliser le module `tkinter` pour créer des interfaces graphiques.
– Dans le fichier contenant la définition de la fonction `dec2bin` (Ex.1.2), saisir le code suivant puis lancer son éxécution :

Complément. Enoncé

```
# Interface graphique
from tkinter import * # Module pour la creation d'interface graphique

def evaluer(event): # action lors de l'appui de la touche 'Entree'
    resultat = "s'ecrit en binaire : "
    resultat += dec2bin(eval(entree.get())) # Appel de la fonction binaire()
    sortie.configure(text = resultat) # Affichage du resultat

def effacer(event): # pour effacer au clic
    entree.delete(0,END) # Effacer du premier au dernier caractere
    sortie.configure(text = '') # Effacer le texte resultat

fenetre = Tk() # cree une fenetre graphique
entree = Entry(fenetre, width = 30) #cree un champ de saisie dans fenetre
entree.insert(0,"Saisir le nombre a convertir ici")
entree.bind("<Return>", evaluer) # Lie l'entree avec la touche 'Entree'
entree.bind("<Button-1>", effacer) # Lie l'entree avec le CLIC gauche
sortie = Label(fenetre) # cree un champ texte dans fenetre
entree.pack() # affiche le champ de saisie
sortie.pack() # affiche le champ texte
fenetre.mainloop() # ouvre la fenetre graphique
```

On pourra ne pas saisir les commentaires.

• Modifier le code pour obtenir une interface graphique qui donne aussi la conversion en héxadécimal.

Complément. Enoncé

```
# Modifications :
def evaluer(event): # action lors de l'appui de la touche 'Entree'
    resultat = "s'ecrit en binaire : "
    resultat += dec2bin(eval(entree.get())) # Appel de la fonction dec2bin
    sortie.configure(text = resultat) # Affichage du resultat
    resultat2 = "s'ecrit en hexadecimal : "
    resultat2 += dec2hex(eval(entree.get())) # Appel de la fonction dec2hex
    sortie2.configure(text = resultat2) # Affichage du resultat

def effacer(event): # pour effacer au clic
    entree.delete(0,END) # Effacer du premier au dernier caractere
    sortie.configure(text = '') # Effacer le texte resultat

fenetre = Tk() # cree une fenetre graphique
entree = Entry(fenetre, width = 30) #cree un champ de saisie dans fenetre
entree.insert(0,"Saisir le nombre a convertir ici")
entree.bind("<Return>", evaluer) # Lie l'entree avec la touche 'Entree'
entree.bind("<Button-1>", effacer) # Lie l'entree avec le CLIC gauche
sortie = Label(fenetre) # cree un champ texte dans fenetre
sortie2 = Label(fenetre) # cree un 2eme champ texte
entree.pack() # affiche le champ de saisie
sortie.pack() # affiche le champ texte
sortie2.pack() # affiche le champ 2eme texte
fenetre.mainloop() # ouvre la fenetre graphique
```