

TD12 - Représentation binaire des entiers relatifs

MPSI - Lycée Thiers

Vérification de la représentation binaire des entiers relatifs.

Enoncé

Enoncé/Corrigé

Exercice 2. Ecriture d'un entier relatif en binaire signé

Enoncé 1-2

Corrigé

Enoncé 3-4

Corrigé

Exercice 1

Exercice 1. Vérification de la représentation binaire des entiers relatifs.

On rappelle qu'avec des opérandes de type entier les opérateurs logiques bit à bit s'obtiennent par :

Opérateur	action
&	ET logique bit à bit
	OU logique bit à bit
^	XOR (OU exclusif) logique bit à bit
~	NON logique bit à bit

On dispose aussi des 2 opérateurs binaires :

Opérateur	action
<<	Décalage à gauche : $a \ll b$ renvoie l'entier obtenu en décalant à gauche de b bits l'écriture binaire de a .
>>	Décalage à droite : $a \gg b$ renvoie l'entier obtenu en décalant à droite de b bits l'écriture binaire de a .

Exercice 1

1. Que renvoie $1 \ll 1$? $1 \ll 2$? $1 \ll 3$? Pour i un entier positif que renvoie $1 \ll i$?
Réponses : 2, 4, 8, 2^i .
2. Pour n un entier et i un entier positif, quelles valeurs peut renvoyer $n \& (1 \ll i)$?
Réponses : 0 ou 2^i selon que le bit de poids i dans la représentation binaire de n soit 0 ou 1.
3. Pour n un entier et i un entier positif, quelles valeurs peut renvoyer $(n \& (1 \ll i)) \gg i$?
Que représente cette valeur pour la représentation binaire de n ?
Réponses : 0 ou 1 selon que le bit de poids i dans la représentation binaire de n soit 0 ou 1. C'est la valeur du bit de poids i dans la représentation binaire de n .

4. Ecrire le script de la fonction suivante :

```
def representation(n,b=64):  
    rep = ""  
    for k in range(b):  
        bit = (n & 1<<k) >> k  
        rep = str(bit) + rep  
    return rep
```

Tester son fonctionnement pour n variant de 0 à 10 (au sein d'une boucle `for`). De même tester son fonctionnement n variant de -10 à -1.

Tester son fonctionnement pour n valant -2^{63} , et pour n valant -2^{64} .

5. Que renvoie pour un entier relatif n l'appel de `representation(n)` ?

Exercice 2. Enoncé 1-2

Exercice 2. Conversion décimal/binaire signé

1. Ecrire une fonction binaire qui prend en paramètre un entier positif et renvoie une chaîne de caractère représentant son écriture binaire.
Par exemple `binaire(17)` renverra `'10001'`
2. Ecrire une fonction `binaireSigne(n,N)` prenant en paramètre un entier relatif `n` et un entier strictement positif `N`. La fonction renverra une chaîne de `N` caractères représentant l'écriture binaire signée sur `N` bits de l'entier `n`. Pour cela :
 - La fonction vérifiera que l'entier `n` peut être représenté sur `N` bits en binaire signé,
 - Si c'est le cas, elle changera `n` en `n` modulo 2^N .
 - appellera la fonction `binaire` avec le paramètre adéquat.
 - Si nécessaire complétera la chaîne renvoyée par autant de zéros que nécessaire pour obtenir une chaîne de `N` caractères.

Exercice 1. Corrigé

1)

```
def binaire(n):
    if n == 0:
        return '0'
    bin = ""
    while n>0:
        bin = str(n%2) + bin
        n = n//2
    return bin
```

2)

```
def binaireSigne(n,N):
    if -2**(N-1) <= n < 2**(N-1):
        n = n%2**N    # ou : if n<0: n += 2**N
        binS = binaire(n)
        binS = (N-len(binS)) * '0' + binS
    return binS
```

Exercice 2. Enoncé 3-4

3. Ecrire une fonction `decimal` qui prend en paramètre une chaîne de caractère représentant l'écriture binaire d'un entier positif et renvoie cet entier positif.
Par exemple `decimal('10001')` renverra 17.
4. Ecrire une fonction `decimalSigne(bin)` prenant en paramètre une chaîne de N caractères qui représente l'écriture en binaire signée sur N bits d'une entier relatif. La fonction renverra l'entier représenté.
Par exemple `decimalSigne('11111111')` renverra -1 .

Exercice 2. Corrigé 3-4

3)

```
def decimal(bin):  
    n = 0  
    for x in bin:  
        n *= 2  
        n += int(x)  
    return n
```

4)

```
def decimalSigne(bin):  
    N = len(bin)  
    n = decimal(bin)  
    if n < 2**(N-1):  
        return n  
    else:  
        return n-2**N
```