

RD01-S

REPRÉSENTATION DES ENTIERS POSITIFS

Propriété 1 : Représentation d'un entier en binaire

Val dec	Val bin	Val dec	Val bin
0	0000	4	0100
1	0001	5	0101
2	0010	6	0110
3	0011	7	0111

- n bits codent 2^n valeurs entre 0 et $2^n - 1$.
- 8 bits codent $2^8 = 256$ valeurs entre 0 et 255.
- le **bit de poids faible** est le bit de droite. C'est le bit de **parité**.
- le **bit de poids fort** est le bit de gauche.
- un **octet** est un groupe de 8 bits.

Méthode 1 : Conversion binaire – décimal : $11011011_2 = 128 + 64 + 16 + 8 + 2 + 1 = 219_{10}$

Rang du bit	7	6	5	4	3	2	1	0
Puissance de 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Bit	1	1	0	1	1	0	1	1
Valeur	128	64	0	16	8	0	2	1

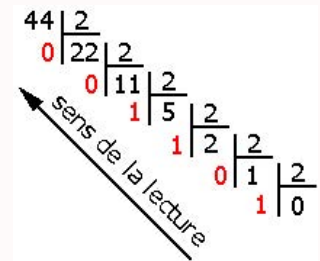
Méthode 2 : Conversion décimal – binaire par division successives par 2

Par exemple pour convertir 44_{10} en binaire :

- On effectue des divisions par 2 jusqu'à ce que le quotient soit nul.
- On lit ensuite les restes de la droite vers la gauche.

On a donc :

$$44_{10} = 00101100_2$$



Méthode 3 : Conversion décimal – binaire par la méthode des soustractions successives

- $222 \geq 128$. On peut donc décomposer $222 = 2^7 + 94$
- $94 \geq 64$. On peut donc décomposer $94 = 2^6 + 30$
- $30 \geq 16$. On peut donc décomposer $30 = 2^4 + 14$
- $14 \geq 8$. On peut donc décomposer $14 = 2^3 + 6$
- $6 \geq 4$. On peut donc décomposer $6 = 2^2 + 2$
- $2 \geq 2$. On peut donc décomposer $2 = 2^1 + 0$



Propriété 2 : Opérations binaires

- **Multiplier** une valeur décimale par 2 revient à décaler tous les bits d'un cran vers la gauche et à ajouter un 0 sur le bit de poids faible.
- **Diviser** une valeur décimale par 2 revient à décaler tous les bits d'un cran vers la droite. Le bit de poids faible est perdu. Cela revient à récupérer le quotient de la division par 2.
- **Multiplier** une valeur décimale par 2^n revient à décaler tous les bits de n crans vers la gauche et à ajouter des 0 sur les bits de poids faible.
- **Diviser** une valeur décimale par 2^n revient à décaler tous les bits de n crans vers la droite. Les bits de poids faible sont perdus. Cela revient à récupérer le quotient de la division par 2^n .

Propriété 3 : Représentation d'un entier en hexadécimal

Val dec	Val hexa
10	A
11	B
12	C
13	D
14	E
15	F

- 1 valeur hexadécimale codent 16 valeurs entre 0 et 15.
- 2 valeurs hexadécimales codent 256 valeurs
- un **octet** est représenté par 2 valeurs hexadécimales comprises entre 00 et FF.

Méthode 4 : Conversion en python**</> Code Python**

```
1 a_bin = 0b1101      # le 0b veut dire que la valeur est écrite en binaire
2 a_dec = int(a_bin)  # conversion en décimal - valeur entière égale à 13
3 a_hex = hex(a_dec)  # conversion en hexadécimal - valeur égale à 0xd
4
5 b_dec = 44
6 b_bin = bin(b_dec)  # conversion en binaire - valeur égale à 0b101100
7 b_hex = hex(b_dec)  # conversion en hexadécimal - valeur égale à 0x2c
8
9 # de manière générale, pour convertir un entier n en base b, on peut utiliser:
10 a_dec = int('1101', 2)      # conversion binaire en décimal égale à 13
11 a_hex = hex(int('1101', 2)) # conversion binaire en hexadécimal égale à 0xd
12 a_bin = bin(int('A3', 16))  # conversion hexadécimal en binaire égale à 0b10100011
```