Arithmétique des ordinateurs et codage

Rappel

Conversions entre bases

Base *b* vers base 10 : il suffit de substituer la valeur b dans l'expression polynomiale par la valeur de la base.

$$(F1C)_{16} = 15 \cdot 16^2 + 1 \cdot 16^1 + 12 \cdot 16^0 = (3868)_{10}$$

Base 10 vers base *b* : division successives du nombre décimal par ***b*** jusqu'à obtenir un quotient nul. Le nombre dans la base ***b*** correspond aux restes des divisions faites dans le sens inverse où ils ont été obtenus.

Division	Quotient	Reste
1836/7	262	2 (a_0)
262/7	37	3 (a_1)
37/7	5	2 (a_2)
5/7	0	5 (a_3)

$\Rightarrow$ $(1836)_{10} = (5232)_7$

1. Effectuez les conversions suivantes :

   a. **$B4F_{(16)} \rightarrow$ Base 10** ***Base b vers Base10***

   $B4F_{(16)} = 11 \times 16^2 + 4 \times 16^1 + 15 \times 16^0 = 2816 + 64 + 15 = 2895$

   b. **$(256)_{(10)} \rightarrow$ Base 2**

   $(256)_{(10)} = (100000000)_{(2)}$

   c. **$(01001101)_{(2)} \rightarrow$ Base 10**

   $0 + 2^6 + 0 + 0 + 2^3 + 2^2 + 0 + 2^0 = 64 + 8 + 4 + 1 = 77$

   d. **Additionner (110011001) par (111011011), afficher le résultat d'addition en binaire et en décimale**

```
   1     1 1    1 1
   1 1 0 0 1 1 0 0 1
   1 1 1 0 1 1 0 1 1
11 0 1 1 1 0 1 0 0
```

   $11\,0\,1\,1\,1\,0\,1\,0\,0 = 2^{10}+2^9+2^8+0+2^6+2^5+2^4+0+2^2+0+ = 256+$

$110011001 = 2^8+2^7+0+0+2^4+2^3+0+0+2^0 = 256+128+16+8+1 =$ 409

$111011011 = 2^8+2^7+2^6+0+2^4+2^3+0+2^1+2^0 = 256+128+64+16+11 =$ <u>475</u>

884

<u>Rappel</u> : Le mode opératoire est le même qu'en base 10

table d'addition

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10 (on pose 0 et on retient 1)

1+1+1 = 11 (on pose 1 et on retient 1)

exemple

```
  1 1      retenues
  1 0 1
+ 1 1 1
------------
1 1 0 0
```

e. 101101$_{(2)}$ → Base 8

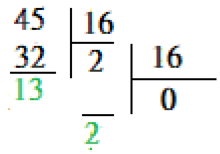
$2^5 + 0 + 2^3 + 2^2 + 0 + 2^0 = 32 + 8 + 4 + 1 = 45$ (en Base 10)

```
45 | 8
40 |____
 5 | 5 | 8
     __|____
     5 | 0
```

$(45)_{(10)} = (5 \quad 5)_{(8)}$

f. 101101$_{(2)}$ → Base 16

$2^5 + 0 + 2^3 + 2^2 + 0 + 2^0 = 32 + 8 + 4 + 1 = 45$ (en Base10)



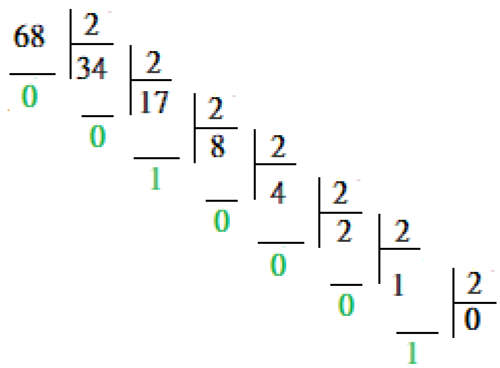
$(45)_{(10)} = (2\ \ D)_{(16)}$

g. **125**$_{(7)}$ **→Base 2** ***Base b vers Base10***

Etape1 : *Conversion de* ***125***$_{(7)}$ *en Base 10*

$1x7^2 + 2x7^1 + 5x7^0 = 49 + 14 + 5 = 68$ (en Base10)

Etape2 : *Conversion de* ***68***$_{(10)}$ *en Base 2*



$(68)_{(10)} = (1000100)_{(2)}$

125$_{(7)}$ **= (1000100)**$_{(2)}$

h. A quoi sert d'utiliser le code hexadécimale !
???

Donner quelques exemple d'utilisation en informatique de ce type de codage.
???

2.? Combien d'entiers positifs peut-on coder en binaire sur un octet ?

Un octet contient 8 bits, on peut donc coder 28 = 256 entiers.

Combien de bits faut-il pour représenter 65 563 entiers différents en binaire ?

2^n >= 65 563

n ln 2 >= ln 65 563

$$n \geq \frac{\ln 65563}{\ln 2} =$$

n >= 17

Autre explication

Avec b bits, on peut coder 2^b entiers différents.
Pour coder n entiers, il nous faut donc m bits tels que
$2^{m-1} < n <= 2m$, c.-à-d.
m-1 < ln n <= m.
On a donc
m = [ln n].
Ainsi pour n=65563, on a
m=ln 65563=17

3. Soit un ordinateur dont les mots mémoire sont composés de 32 bits. Cet ordinateur dispose de 4 Mo de mémoire. Un entier étant codé sur un mot, combien de mots cet ordinateur peut-il mémoriser simultanément ?

4 Mo / 32 bits = 1024

On a donc **1024 mots binaires** que l'on peut enregistrer sur 4 Mo

Rappels de calcul

32 bits = 4 octets

4 Mo = 4 x 1024 = 4096 octets (car 1 Mo = 1024 octets)

Quelle est la plus grande valeur entière (décimale) que cet ordinateur peut mémoriser, cette valeur étant représentée par son codage binaire pur ?

???

4. Coder en binaire sur un octet les entiers 9 et 5

puis effectuer la multiplication binaire des entiers ainsi codés.