Notes de cours – Programmation 1

Les NOMBRES

En décimal (base 10)

La base 10 signifie que nos nombres sont formée de chiffre de 0 à 9. Si je prends une autre base, par exemple la base 6, les nombres seraient composés de chiffre de 0 à 5. Il serait impossible que 1543 soit en base 4, car un nombre en base 4 doit être composé des chiffres de 0 à 3, par contre 1332_4 est tout à fait possible.

Prenons 1654 en base 10 par exemple :

Poids correspondants au chiffre	10^3	10^2	10^1	10^0
Chiffre	1	6	5	4

Donc $1654_{10} = 1x10^3+6x10^2+5x10^1+4x10^0$

Prenons 1001_2 en binaire (base 2)

Poids correspondants au chiffre	2^3	2^2	2^1	2^0
Chiffre	1	0	0	1

Ce nombre (1001_2) en binaire correspond à 9_{10} en décimal car : $1x2^3+1x2^0=9_{10}$

Les ALGORITHMES

1. <u>Passer de la base X à la base 10 (où X correspond à n'importe quel nombre ou chiffre)</u>

Il suffit de multiplier le chiffre avec le poids qu'il contient et d'additionner le résultat pour tous les chiffres composants ce nombre.

Exemple 1 : 1223_4 en base 10?

Poids correspondants au chiffre	4^3	4^2	4^1	4^0
Chiffre	1	2	2	3

$1x4^3+2x4^2+2x4^1+3x4^0 = 107_{10}$

Exemple 2 : 110011_2 en base 10

Poids correspondants au chiffre	2^5	2^4	2^3	2^2	2^1	2^0
Chiffre	1	1	0	0	1	1

$1x2^5+1x2^4+1x2^1+1x2^0 = 51_{10}$

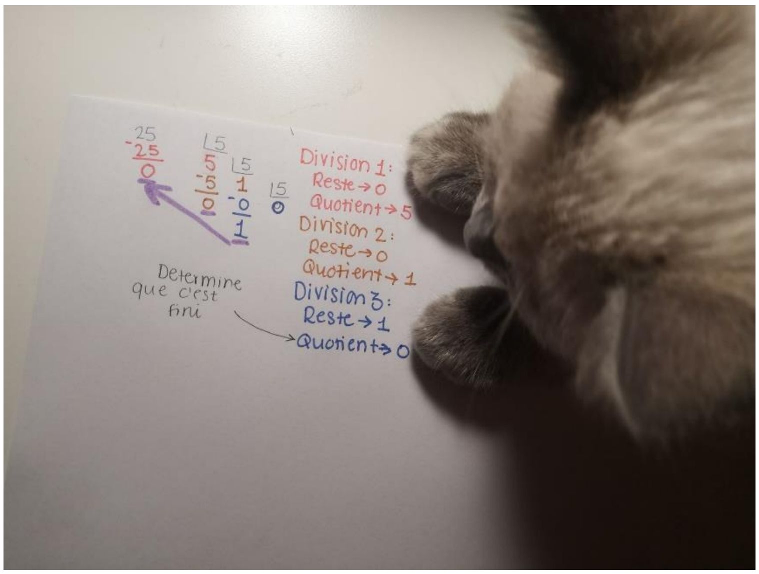
2. <u>Passer de la base 10 à la base X.</u>

Méthode 1

Divisions multiples

Exemple : 25_{10} en base 5

1ère étape : diviser le nombre voulu par la base désirée (dans ce cas, diviser 25 par 5)



2e étape : Rediviser le quotient de la division par la même base (diviser 5 par 5

3e étape : Étape 2 mais avec le quotient de la division de l'étape précédente. (diviser 1 par 5)

Répété les étapes jusqu'à ce que le quotient de la division soit <u>0</u>.

ne étape : le nombre initial (25) en base désirée (5) est les restes des divisions successives du plus récent au plus ancien (100_5).

Donc $\mathbf{25_{10} = 100_5}$

Méthode 2

Exemple : 25_{10} en base 2

$2^0=1$, $2^1=2$, $2^2 = 4$, $2^3 = 8$, $2^4=16$, $2^5=32$

25_{10} – 16 (le plus grand 2^n qui entre dans 25, dans ce cas n=4) = 9

9 – 8 (le plus grand 2^n qui entre dans 9, dans ce cas n=3) = 1

1 – 1 (le plus grand 2^n qui entre dans 1, dans ce cas n=0) = 0 STOP

2^4	2^3	2^2	2^1	2^0
1	1	0	0	1

Donc $\mathbf{25_{10} = 11001_2}$

REPRÉSENTATION DES CHIFFRES ET DES NOMBRES UTILISANT PLUSIEURS NOTATIONS

1. Représentation binaire non signée sur n bits

NB : le nombre de bits signifie le nombre de poids disponibles pour représenter le chiffre.

Si n=nb de bits, le « range » des chiffres que nous pouvons représenter est de $\mathbf{2^n-1}$.

Ex : avec 4 bits on peut représenter les nombres de 0 à 2^4-1, 0 à **15.**

Min : 0000 → 0 Max : 1111 → 15

Avec cette méthode, nous pouvons facilement représenter les **<u>nombres entiers positifs</u>** seulement.

2. Encodage binaire signé en complément à 2

Complément : représentation d'un nombre négatif avec son complément (ils se complètent) positif.

Le nombre de bits nous permet de seulement représenter un certain « range » de nombre. Par exemple, avec 2 bits en base 10 nous pouvons seulement représenter les nombres de 0 à 99.

Méthode 1 en mode décimal

« Range » des nombres que je peux représenter avec 2 bits en base 10 (ex) :

0 ⟷ 99

Il faut « couper » le domaine de valeur en 2, la section de droite de la ligne médiane représente les nombre de 0 à 49 qui demeure leur valeur exacte. La section à gauche de la médiane contenant les chiffres de 50 à 99 représentent les nombres négatifs où 99 équivaut à -1 et 50 équivaut à -49.

50	99 | 0	49
-50	-1 0	49

NB : il devient donc impossible de représenter des nombres plus grands que 50 avec seulement 2 bits.

Ex 1 : S'il y est demandé de représenter -39 à l'aide de 2 bits :

(Maximum possible) 99 – 39 = 60 + 1 (le zéro)= 61. Donc -39 est représenté par 61 en notation complément sur 2 chiffres décimaux. (Pouvons faire 100-39)

Ex 2 : Trouver le complément de 61 : 99-61=38+1 = 39

Je sais que 61 représente un nb négatif car il est plus grand que la moitié de mon range.

Méthode 2 : en mode binaire

Exemple : n=6 en mode binaire. Représente -16_{10} sur 6 bits.

1) Convertir 16_{10} (valeur absolue) en base 2

$$16_{10} = 2^4 = 10000$$

2) Écrire sur le nombre de bits alloué (ajouter des zéros au besoin)

010000 (6 bits)

3) **Si le nombre binaire débute par zero : valeur positive, si le nombre binaire débute par 1 : valeur négative.** Si jamais ça ne correspond pas (comme dans notre cas), utiliser les méthodes d'encodage.

complément à 1 ou « flip and add »

010000 — Changer les 1 pour des 0 et vice-versa

Complément à 1 ou restreint — 101111 — Ajouter 1.

Complément à 2 — **110000**

complément à 2 ou soustraction binaire

Emprunt			~~2~~ 1	2					
1 suivi de 0 x n	-	~~1~~	0	0	0	0	0	0	0
Nombre binaire			0	1	0	0	0	0	0
Complément à 2		x	**1**	**1**	**0**	**0**	**0**	**0**	**0**

Dans les deux cas : -16_{10} → complément à deux sur 6 bits → 110000_2

Méthode 3

La position la plus à gauche est -2^{n-1}.

Ex :

-2^3	2^2	2^1	2^0
0	1	0	1

= 4+1=5

-2^3	2^2	2^1	2^0
1	0	0	0

= -8

Le plus petit nombre possible avec 4 bits : 1000 → -8

Le plus grand nombre possible avec 4 bits : 0111 → 7

Donc le range est de -8 à 7