3 – Codage des nombres à virgule

Tout comme les nombre entiers relatifs, les nombres à virgule peuvent aussi être représentés en base 2.

1 – Conversion de la base 2 à la base 10

A gauche de la virgule, sont représenté les puissances de 2 positives, et à droite, les puissances de 2 négative.

Ex : 110,1011

1x2^2 + 1x2^1 + 0x2^0 + 1x2^-1 + 0x2^-2 + 1x2^-3 + 1x2^-4

2^2 + 2^1 + 2^-1 + 2^-3 + 2^-4

6,6875

2 – Conversion de la base 10 vers la base 2

On commence par convertire la partie entière, puis la partie décimale.

Pour convertir la partie décimale, on va procéder par des multiplications par 2 successives. Après chaques multiplication, le résultat est reporté sans sa partie entière. On continue jusqu'a obtenir 1.

Ex : 6,6875

Partie entière : 110
Partie décimale :

0,6875 x 2 =	1,375
0,375 x 2 =	0,75
0,75 x 2 =	1,5
0,5 x 2 =	1

On lit ensuite la partie entière de chaques résultats, de haut en bas, et l'on obtient la représentation binaire.
Ici : 1011

Notre 6,6875 donne bien 110,1011

3 – Ecritures infinies

Certains nombre ont une écriture décimale infinie périodique.
Par exemple 7/11 = 0,63636363...
Certains ont même une écriture binaire infinie périodique, alors que leur écriture décimal est finie.

Ex :	0,2 x 2 = 0,4 0,4 x 2 = 0,8 0,8 x 2 = 1,6 0,6 x 2 = 1,2 0,2 x 2 = 0,4 ...	Ici, la ligne 0,2 x 2 se répète. Les chiffres 0011 vont aussi se répèter indéfiniment. L'écriture de 0,2 est donc infinie et périodique. 0,2 base 10 = 0,001100110011... base 2

Exo :

0,1 x 2 = 0,3 x 2 = 0,4 x 2 = 0,5 x 2 = 0,6 x 2 = ... jusqu'a 0,9

4 – Codage des nombres à virgule

1 – Virgule fixe et virgule flottante

Il existe 2 codages à virgule en machine :

le codage en virgule fixe et le codage en virgule flottante (norme IEEE-754).

Le codage à virgule fixe est utilisée sur les processeurs à faible coût (les microcontrôleurs)
Le codage à virgule flottatnte est urilisé partout ailleurs (ordinateurs, smartphones,...)

L'idée du codage à virgule fixe est de retenir un nombre fixe de chiffre aprés la virgule.

Dans le cas du codage à virgule flottante, l'idée est de retenir un **nombre significatif de chiffres**.
Beaucoup de chiffre après la virgule pour les petits nombres
Beaucoup de chiffre avant la virgule pour les grands nombres

2 – Les nombres à virgule sont approchés

Quel que soit le codage choisit, le problème est le même :
Si le nombre a une écriture infine en base 2, il ne peut pas être représenté dans un ordinateur qui ne stocke qu'un nombre fini de chiffres et utilise la base 2.
Le nombre manipulé en machine n'et alors qu'une valeur approchée du nombre réel.

C'est le cas avec tous les langages qui utilisent les nombres à virgule flottante (càd quasiement tous les langages qui ne sont pas spécialisés dans le calcul exact).

Ex :

0,5 – 0,2 – 0,2 – 0,1 ne donne généralement pas 0
en Python : -2.7755575615628914e-17

Ajouter un petit nombre à un grand, donne des résultats surprenants

Ex :

9 007 199 254 740 992.0 + 1.0 == 9 007 199 254 740 992.0
En python : True

L'addition avec des flottants n'est plus commutative

Ex :

9 007 199 254 740 992.0 + 1.0 + 1.0 = 9 007 199 254 740 992.0
1.0 + 1.0 + 9 007 199 254 740 992.0 = 9 007 199 254 740 994.0