1) Présentation des nombres flottants

L'erreur est "humaine" ou pas !

Le 25 février 1991, pendant la guerre du Golfe, une batterie de missiles Patriot américains à Dharan, en Arabie Saoudite, n'a pas réussi à intercepter un missile Scud irakien en approche. Le Scud a frappé une caserne de l'armée américaine et a tué 28 soldats.

Un rapport du bureau de la comptabilité générale, GAO/ IMTEC-92-26, intitulé Patriot Missile Defense : Software Problem Led to System Failure at Dhahran, Saudi Arabia (Un problème de logiciel a conduit à une défaillance du système à Dhahran, en Arabie Saoudite) a fait état de la cause de cette défaillance.

Il s'avère que la cause était un calcul inexact du temps écoulé depuis le démarrage en raison d'erreurs d'arithmétique informatique. Plus précisément, le temps en dixièmes de seconde mesuré par l'horloge interne du système a été multiplié par 10 pour produire le temps en secondes. Ce calcul a été effectué à l'aide d'un registre de 24 bits. En particulier, la valeur 1/10, qui a une expansion binaire non terminale, a été coupée à 24 bits.

La petite erreur de découpage, lorsqu'elle est multipliée par le grand nombre donnant le temps en dixièmes de seconde, conduit à une erreur significative.

En effet, la batterie du Patriot était en service depuis environ 100 heures, et un calcul facile montre que l'erreur de temps résultante due à l'erreur de hachage amplifiée était d'environ **0,34 seconde**. (Le nombre binaire de 1/10 est de 0,0001100110011001100110011001100.... Mais, le registre 24 bits du Patriot a stocké à la place 0.00011001100110011001100 introduisant une erreur de $(0.0000000000000000000000011001100)_2$ soit environ 0.000000095 en décimal. En multipliant par le nombre de dixièmes de seconde en 100 heures, on obtient 0,000000095 × 100 × 3600 × 10 = 0,34 s). Un Scud se déplace à environ 1 676 mètres par seconde, et parcourt donc plus d'un demi-kilomètre dans ce laps de temps. Cette distance est suffisante pour que le Scud entrant se trouve en dehors de la "porte du champ de tir" que le Patriot a suivie.

Pour en avoir le coeur net !!!

Que font 0,7 x 0,7 sur python ?

```
>>> 0.7*0.7
0.48999999999999994
>>>
```

Virgule fixe et virgule flottante

Il existe deux façons de coder les nombres à virgule:

- Le codage en **virgule fixe** qui consiste à garder *un nombre fixe de chiffes après la virgule*.

...	8	4	2	1	0.5	0.25	0.125	...
...	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	...
...	0	1	1	0,	1	0	1	...

Exemple : $110,101_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-2} = 4 + 2 + 0,5 + 0,125 = 6,625$

Considérons le nombre 3,6875 . Il se décompose en une partie entière (3) et une partie décimale (0,6875).

partie entière *: 3=11_2*

partie décimale **:** la conversion de se fait en plusieurs étapes.

$0,6875 \times 2 = \mathbf{1},375$
$0,375 \times 2 = \mathbf{0},75$
$0,75 \times 2 = \mathbf{1},5$
$0,5 \times 2 = \mathbf{1}$

Donc 3,6875 s'écrit en binaire **$11,1011_2$**

Exercices :

Donner l'écriture binaire de 20,875.

Donner l'écriture binaire de 0,2.

Conclusion :

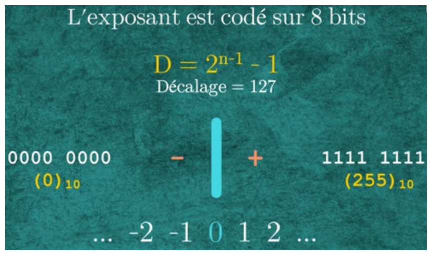
Principe du codage en virgule flottante

Le **codage en virgule flottante** est une méthode de représentation des nombres réels dans un ordinateur, permettant de gérer une large plage de valeurs tout en conservant une précision raisonnable. Il est particulièrement utilisé pour des calculs scientifiques, graphiques et financiers.

La norme IEEE 754 définit le format le plus utilisé, notamment avec des représentations sur **32 bits** (simple précision) et **64 bits** (double précision).

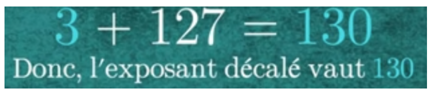
Le signe (S) : 1 bit

- 0 pour un nombre **positif**
- 1 pour un nombre **négatif**



L'exposant (E) : 8 bits (pour IEEE 754 simple précision)

- Il sert à déplacer la virgule et permet de représenter de très grands ou de très petits nombres.



- Il est stocké en "exposant biaisé" (biais= 127 pour simple précision).

La mantisse (M) (ou fraction) : 23 bits

- Stocke les chiffres significatifs du nombre en base 2.
- La normalisation impose que le premier bit de la mantisse soit toujours 1 (il est donc implicite).

Comment faire des tests d'égalité sur les flottants ?

Si `a` et `b` sont deux flottants,

```
a = 0.1
b = 0.3 - 0.2
if a == b :
    print("a et b sont égaux")
else :
    print("a et b sont différents")
```

Que renvoie ce script ?

Si *vraiment* un test d'égalité est nécessaire, on ne va pas tester l'égalité entre `a` et `b` mais leur **proximité**, grâce à la valeur absolue de leur différence.

La fonction `abs(a-b)` renvoie un nombre positif égal à la distance entre `a` et `b`. Il faut alors décider d'un écart minimal `e` en dessous duquel on considèrera que `a` et `b` sont égaux.

Le script :

```
a = 0.1
b = 0.3 - 0.2
e = 10**(-12)
if abs(a - b) < e :
    print("a et b sont égaux")
else :
    print("a et b sont différents")
```

Que renvoie ce script ?

pourquoi ?

Exercice d'application :

On considère la fonction $f(x) = x^3 -6x + 2$
L'équation $f(x) = 1$ admet une solution unique dans l'intervalle [0 ; 1] .
Trouver une valeur approchée de cette solution à 10^{-5} près. On prendra `e=0,001`.

Exercice 1 : Conversion en Virgule Fixe

On utilise un format virgule fixe 8.8 (8 bits pour l'entier, 8 bits pour la fraction).

1. Représenter **5,75** en binaire avec ce format.
2. Représenter **-3,5** en complément à 2 avec ce format.
3. Quelle est la valeur en décimal du nombre binaire 00011010.10100000 dans ce format ?

Exercice 2 : Addition en Virgule Fixe

On utilise un format virgule fixe 6.2 (6 bits pour l'entier, 2 bits pour la fraction).

1. Représenter les nombres **3,25** et **2,75** en binaire dans ce format.
2. Effectuer leur addition en binaire directement.
3. Donner le résultat final en décimal.

Exercice 3 : Conversion en Virgule Flottante IEEE 754

On travaille avec la **simple précision (32 bits)**.

1. Convertir le nombre **-19,375** en binaire.
2. Normaliser l'écriture sous la forme 1.xxxxx×2E1.xxxxx \times 2^E.
3. Déterminer les trois champs (signe, exposant biaisé, mantisse).
4. Donner la représentation complète en **IEEE 754 (32 bits)**.

Exercice 4 : Comparaison Virgule Fixe vs Virgule Flottante

1. Quelle est la principale limitation de la virgule fixe par rapport à la virgule flottante ?
2. Pourquoi la virgule flottante est plus adaptée aux très grands et très petits nombres ?
3. Donnez un exemple où la virgule fixe serait plus efficace que la virgule flottante.