TD : Représentation des flottants et norme IEEE754

Comment représenter des flottants de grande taille dans la mémoire d'un ordinateur ? Pour cela, il faut ruser un peu. Étudions cela…

Sur 32 bits, une idée est de prendre 1 bit pour écrire le signe, de réserver 8 bits pour coder la partie entière (non signée) et le reste pour coder la partie fractionnaire (après la virgule)

Signe Partie entière Partie fractionnaire

1) De combien de bits sera constitué la partie fractionnaire ?
2) L'idée la plus naïve est de coder la partie entière comme nous l'avons fait avec les entiers.

   a. Quelle est la partie entière la plus grande que l'on peut écrire sur les 8 bits réservés ? Cela vous semble-t-il suffisant ?

   b. De la même manière, quelle serait la précision maximale que vous pourriez atteindre avec cette représentation ? Cela vous semble-t-il suffisant ?

On se retrouve avec des nombres bien trop petits et/ou une précision assez limitée… Essayons d'améliorer cela !

3) Écrivez les nombres 3457064,21 et 0,0321 sous forme de notation scientifique.

L'expression que vous avez trouvé est aussi appelée une notation mantisse-exposant. D'une manière générale, en base 10, elle s'écrit : $\boxed{\pm M \times 10^e \text{ avec } M \in [1{,}10[\text{ et } e \in \mathbb{Z}.}$

4) **a.** À votre avis, que devient la notation mantisse-exposant en base 2 ?

   b. À l'aide de puissance de 2, écrivez les nombres 110_2, 1001_2 puis $0{,}0101_2$ sous forme mantisse-exposant. *Pour simplifier, on écrira l'exposant de la puissance de 2 en base 10.*

   c. Que pouvez-vous en déduire sur la valeur de la mantisse M ? Est-il nécessaire de la stocker entièrement en base 2 ?

   d. Proposez un procédé de stockage basé sur la notation mantisse-exposant. On s'aidera du tableau ci-dessous :

   e. Avec votre procédé de stockage, proposez la notation binaire sur 32 bits de -66,825.

Là, on se dit qu'on est plutôt beau gosse. On a même réussi à économiser un bit dans la mantisse pour aller à une précision de 2^{-23} (comme avant en gros…).

5) Quel problème rencontre-t-on si on essaie de donner la notation binaire sur 32 bits de 0,125 ?

6) Sur 8 bits, avec le décalage, quel exposant écrire afin de coder 2^0 ? 2^1 ? 2^{128} ?

L'ensemble de ces spécifications est appelée norme IEEE-754

7) Soit le nombre "-10,125" en base 10. Représentons-le en simple précision :
   - **a.** Écrire en base 2 le nombre $10{,}125_{10}$.
   - **b.** Écrire ce nombre sous forme mantisse-exposant puis décaler l'exposant de 127 unités.
   - **c.** Écrire l'exposant en base 2.
   - **d.** Conclure sur la représentation du nombre les 32 bits
   - **e.** Écrire le nombre obtenu en hexadécimal

C'est beau la norme IEEE754 !

8) Représenter le nombre "0,1" sous la norme IEEE754.

9) Soit le nombre flottant au format simple précision :
   00111101110011001100110011001100.
   Trouvez la représentation <u>en base 10</u> de ce nombre.

10) Déterminez la représentation au format simple précision d'un tiers (1/3) en binaire et en hexadécimal.

<u>Pour aller plus loin :</u>

❖ En notation mantisse-exposant, il est impossible de représenter le nombre 0. Pour pallier à ce problème, **l'exposant 0 est réservé.** Lorsque l'exposant vaut 0, nous pouvons avoir deux valeurs : +0 ou -0 (en fonction du signe).

❖ De même, **l'exposant 255 est réservé** pour des **valeurs spéciales** telles $+\infty$, $-\infty$ ou NaN.
Lorsque, en simple précision, le nombre cherché est en-dehors de $[-1{,}7 \times 10^{38}; 1{,}7 \times 10^{38}]$, on atteint un infini : on met l'exposant à 255 et la mantisse à 0 pour signaler cela.
Lorsque l'on réalise une opération interdite (division par 0, $\sqrt{-1}$ …), on renvoie une valeur spéciale appelée **NaN** (**N**ot **a** **N**umber) : on met l'exposant à 255 et la mantisse différente de 0 pour signaler ce cas.