

Codage de l'information

Représentation des flottants

Exercice 3-1 Combien de décimales pour explorer l'univers ?

Petit exercice pour l'auto-étude

- La sonde Voyager 1, lancée le 5 septembre 1977, est actuellement (2 février 2022) à 21h 35mn 27s - lumière de la Terre. C'est l'objet créé par l'humanité qui en est le plus lointain.
- La vitesse de la lumière dans le vide est de 299 792 458 m/s
- La valeur de Π utilisée par Python est : 3.141592653589793
- On veut calculer la circonférence du cercle de centre la Terre et de rayon la distance de la Terre à Voyager 1.
- De quelle longueur se trompe-t-on si on utilise une valeur de Π erronée sur sa dernière décimale : 3.141592653589794 ?

Exercice 3-2 *Nombres à virgule flottante selon la norme IEEE 754*

Dans la norme IEEE-754, les nombres flottants peuvent être représentés en base $b = 2$ sous la forme

$$x = (-1)^s \times 2^e \times m$$

où m est un réel compris entre 1 inclus et 2 exclu, appelé *mantisse*, et e est un entier relatif, appelé *exposant*.

Il existe quatre codages :

1. sur 16 bits pour une demie précision ;
2. sur 32 bits pour une simple précision ;
3. sur 64 bits pour une double précision ;
4. sur 128 bits pour une quadruple précision.

Dans cet exercice, on va considérer la version sur 16 bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	E					m - 1									

- les dix bits de poids faible forment la représentation binaire de la partie fractionnaire de la mantisse, c'est-à-dire de $m - 1$, ainsi le bit de poids faible de la mantisse représente la valeur $\frac{1}{2^{10}}$ alors que le bit de poids fort de la mantisse représente la valeur $\frac{1}{2}$;
- les cinq bits suivants stockent E duquel on obtient l'exposant e en calculant $e = E - 15$;
- et le bit de poids fort indique le signe du nombre (0 = positif, 1 = négatif).

Question 1 On représente un nombre réel par $\overline{\text{B4D2}}_{16}$.

Quelles sont les valeurs de s , E et $m - 1$? En déduire la valeur du réel ainsi représenté.

Question 2 Représentez le nombre 5,015625 dans la norme IEEE 754 sur 16 bits.

Question 3 Même question avec 4,8125.

Question 4 Faites la somme et le produit des deux nombres précédents, en utilisant leur codage IEEE754.

Question 5

Pourquoi l'écriture binaire de $\overline{0,1}_{10}$ est-elle infinie ?

Question 6 Quelle sera la période maximale du développement binaire de $\overline{0,1}_{10}$?

Question 7 Calculez la suite infinie des chiffres binaires du développement de $\overline{0,1}_{10}$ par la méthode vue en cours.

Question 8 Donnez la représentation complète de $\overline{0,1}_{10}$ dans la norme IEEE-754 en demi-précision.

Question 9 Quel est le nombre exact codé par cette représentation ?

Question 10

Soit un flottant codé dans la norme IEEE . Donnez un algorithme simpliste qui le multiplie par 2.

Question 11 Dans quels cas cet algorithme ne fonctionnera pas ?

Question 12 Comment vois-t-on qu'un nombre flottant en norme IEEE754 est plus grand que 1 ?

Question 13 Comment vois-t-on qu'un nombre flottant en norme IEEE754 est compris entre 1 et 2 ?

Question 14 Décrivez un algorithme simple qui, étant donné un nombre normalisé N au format IEEE754 compris entre 1 et 2, construit le nombre (toujours sous la norme IEEE754) égal à $N - 1$

Question 15

- Utilisez le programme python vu en cours pour retrouver le développement binaire de $\overline{0,1}_{10}$.
- Comprenez-vous pourquoi dans l'exemple du cours, la variable n finit toujours par valoir zéro après un nombre fini d'itérations ?
- A quoi correspond ce nombre d'itérations ?

Exercice 3-3

Question 1 Quel est le plus petit réel positif non nul sur 16 bits ? le plus grand ? le deuxième plus grand ?

Exercice 3-4 *Premier entier non représenté par un flottant*

En Python, les entiers sont de précision illimitée, ce qui signifie qu'ils peuvent être aussi grands que la capacité de la machine le permet. En revanche les flottants sont de précision limitée : ils sont stockés sur 64 bits en suivant la norme IEEE-754.

Nous cherchons à trouver le plus petit entier naturel qui ne peut pas être représenté comme un flottant, en suivant la norme IEEE-754. Autrement dit, nous cherchons le plus petit entier i tel que l'expression Python suivante soit **vraie** : `i != floor(float(i))`

Question 1 Dans la représentation IEEE-754 sur 64 bits, il existe une seule valeur de l'exposant e pour laquelle tous les flottants positifs ayant cette valeur d'exposant sont uniquement les entiers de 2^e à $2^{e+1} - 1$. Quelle est cette valeur de e ?

Question 2 Quel est le plus petit entier qui ne peut pas être représenté par un flottant dans la norme IEEE-754 sur 64 bits ?