Fonctionnement des Ordinateurs

TP4 - Nombres flottants

B. QUOITIN
Faculté des Sciences
Université de Mons

Résumé

L'objectif de cette séance d'exercices est de renforcer votre compréhension de la représentation des nombres flottants et du standard IEEE 754.

Table des matières

1 Motivation

1.1 Erreur suite à une addition

Ecrivez un programme en Python 3 qui stocke dans les variables x et y, les nombres $0,1$ et $0,2$ respectivement. Affichez les valeurs de ces variables à la console. Effectuez ensuite l'addition $x + y$ et affichez la somme à la console. Que constatez-vous ? Comment expliquez-vous ce phénomène ?

Q1) Programme en Python

1.2 Swamping lors de l'addition

Ecrivez un programme en langage C ou Java qui effectue l'addition des nombres $x = 2$ et $y = 33554432$ (2^{25}) placés dans des variables de type `float` (IEEE 754, en simple précision). Que constatez-vous ?

Q2) Programme en C ou Java

1.3 Somme de fractions

Ecrivez un programme en C ou Java qui ajoute 700 fois la fraction $\frac{1}{7}$ à une variable x valant initialement 0. Utilisez d'abord le type `float` et ensuite le type `double`. Que constatez-vous ?

Q3) Programme en C ou Java

Pouvez-vous écrire un programme donnant un comportement similaire en Python ?

2 Virgule flottante

2.1 Représentation IEEE754

Le standard IEEE 754 définit des représentations de nombres flottants. Cette représentation est composée d'un *bit de signe*, d'un *exposant* représenté sur E bits et d'une *mantisse* représentée sur M bits. Donnez les paramètres des représentations en simple et double précisions. La variable B est le biais.

Paramètres IEEE 754

Précision	**Taille (bits)**	E	M	B
simple	32	8	23	127
double	64	11	52	1023

Donnez les formules permettant d'interpréter un mot binaire de 32 bits (simple précision) selon ce format. Attention, par convention certaines valeurs spéciales sont encodées et il est important de distinguer les représentations normalisée et dénormalisée.

Q4) Interprétation nombre IEEE 754 simple précision

Donnez les nombres représentés en IEEE 754 simple précision avec les mots binaires suivants.

Q5) `0 00000000 00000000000000000000000`

Q6) `0 10000000 00000000000000000000000`

Q7) `0 10000000 11000000000000000000000`

Q8) `0 11111111 01111111111111111111111`

Q9) `1 00000000 10000000000000000000000`

2.2 Limites de la représentation IEEE754 (format réduit)

Donnez les plus grand et plus petit nombres positifs non nuls normalisés représentables avec les tailles de mantisse M, d'exposant E et les biais B suivants.

Q10) $M = 4$, $E = 3$, $B = 4$

Q11) $M = 8$, $E = 4$, $B = 8$

Question identique pour les nombres dénormalisés.

Q12) $M = 4$, $E = 3$, $B = 4$

Q13) $M = 8$, $E = 4$, $B = 8$

2.3 Normalisation d'un nombre

Ecrivez un algorithme qui prend en argument un nombre réel x non nul et qui donne comme résultat la valeur x_{norm} normalisée de x, c'est-à-dire telle que $x_{\text{norm}} \in [1, 2[$. Arrangez-vous également pour que votre algorithme fournisse l'exposant e de 2 tel que $x = x_{\text{norm}} \times 2^e$, avec $e \in \mathbb{Z}$.

Q14) Programme de Normalisation

2.4 Conversion vers IEEE754 (format réduit)

Soit une taille de mantisse $M = 4$, une taille d'exposant $E = 3$ et un biais $B = 4$. Donnez la représentation des nombres suivants.

Q15) $1,5$

Q16) $0,625$

Q17) $3,2$

Q18) $64,17$

Q19) $0,03125$

Q20) $-0,015625$

2.5 Biais adéquat

Soit une taille d'exposant E, donner la valeur du biais permettant d'équilibrer au mieux le nombre d'exposants positifs et négatifs.

Q21) $E = 3$

Q22) $E = 8$

2.6 Erreur absolue, erreur relative et epsilon machine

Soit une taille de mantisse $M = 4$, une taille d'exposant $E = 3$ et un biais $B = 4$. Fournissez la formule donnant l'epsilon machine, ϵ_M, la borne supérieure sur l'erreur relative. Donnez sa valeur pour $M = 4$.

Q23) **epsilon machine (ϵ_M)**

Calculez l'erreur d'approximation causée lors de la représentation des nombres suivants. Fournissez l'erreur absolue (Δ_x) et l'erreur relative (ϵ_x). Comparez cette dernière à l'epsilon machine. Note : les nombres ci-dessous sont identiques à ceux de la Section 2.4.

Q24) $1,5$

Q25) $0,625$

Q26) $3,2$

Q27) $64,17$

Q28) $0,03125$

Q29) $0,015625$

2.7 Arrondis

En utilisant la méthode *round-to-nearest-even* (*arrondi au plus proche pair*), arrondissez les nombres dont la représentation binaire est donnée ci-dessous, de façon à ne garder que la partie entière.

Q30) `101.000000`

Q31) `1.010101`

Q32) `11.110000`

Q33) `0001.100001`

Q34) `0.1`

Q35) `0.01111111111111111111111111111111111111`

Q36) `11111.100000`

Q37) `100.100000`

2.8 Addition

Effectuez l'addition des nombres x et y dont les représentations sont données ci-dessous. Indiquez les différentes étapes de l'opération. Le format utilisé est similaire à IEEE 754 avec les tailles de mantisse $M = 6$, d'exposant $E = 4$ et un biais $B = 8$. On ne considère pas le bit de signe. Afin de vérifier vos additions, donnez également les valeurs de x, de y et du résultat $\widehat{x + y}$ de l'addition en décimal. Attention, $\widehat{x + y}$ peut-être différent de $x + y$ en raison d'arrondis ou de dépassements de capacité.

Attention, ne pas oublier le bit caché en représentation normalisée. Dans les solutions ci-dessous, il sera indiqué entre parenthèses avant la mantisse. Par exemple, pour une mantisse valant 0, encodée sur 6 bits, on affichera `(1)000000`.

Q38) $x =$`1000 000000`; $y =$`1000 000000`

Q39) $x =$`1000 000000`; $y =$`0110 000000`

Q40) $x =$`1000 100000`; $y =$`1000 110000`

Q41) $x =$`1000 000000`; $y =$`0001 000000`

Q42) $x =$`1110 000000`; $y =$`1110 000000`

Q43) $x =$`0000 100000`; $y =$`0000 110000`

Q44) $x =$`1010 110000`; $y =$`1000 000100`

Q45) $x =$`1010 110000`; $y =$`1000 000110`

2.9 Soustraction

En utilisant le même système qu'à la Section 2.8, effectuez la soustraction des nombres dont les représentations sont données ci-dessous.

Q46) $x =$`1011 101010`; $y =$`1011 101010`

Q47) $x =$`1001 010000;` $y =$`1000 100000`