T.D. 3
Systèmes de numération flottante

Exercice 1

Convertissez les nombres ci-dessous dans le format IEEE754 **simple précision**. Vous exprimerez le résultat final sous **forme binaire**.

1. 142
2. -64,25
3. 30,625
4. 0,3125

Exercice 2 (travail à la maison)

Convertissez les nombres ci-dessous dans le format IEEE754 **double précision**. Vous exprimerez le résultat final sous **forme binaire**.

1. -256
2. 964
3. 8,19140625
4. 0,2890625

Exercice 3

Donnez la représentation associée aux mots binaires codés au format IEEE754 **simple précision** présents ci-dessous. Si une représentation est un nombre, vous l'exprimerez en base 10 sous la forme $k \times 2^n$ où k et n sont des entiers relatifs.

1. $1011\ 1101\ 0010\ 0000\ 0000\ 0000\ 0000\ 0000_2$
2. $0100\ 1111\ 0001\ 0000\ 0000\ 0000\ 0000\ 0000_2$
3. $1100\ 1100\ 0101\ 0011\ 0000\ 0000\ 0000\ 0000_2$
4. $1111\ 1111\ 1000\ 0000\ 0000\ 0000\ 0000\ 0000_2$
5. $0000\ 0000\ 0011\ 0000\ 0000\ 0000\ 0000\ 0000_2$

Exercice 4 (travail à la maison)

Donnez la représentation associée aux mots binaires codés au format IEEE754 **double précision** présents ci-dessous. Si une représentation est un nombre, vous l'exprimerez en base 10 sous la forme $k \times 2^n$ où k et n sont des entiers relatifs.

1. $417C\ C800\ 0000\ 0000_{16}$
2. $C808\ 0000\ 0000\ 0000_{16}$
3. $000A\ 0000\ 0000\ 0000_{16}$
4. $8000\ 0000\ 0000\ 0000_{16}$
5. $FFF0\ 0001\ 0000\ 0000_{16}$

Exercice 5

Pour chaque question, vous traiterez le cas des codages **simples et doubles précisions** du format à **mantisse normalisée**.

1. Déterminez, en valeur absolue, le plus petit et le plus grand nombre flottant.
2. Quel est le plus petit nombre strictement positif qui, ajouté à 1, donne un résultat différent de 1 ?

Exercice 6

Soit le programme suivant écrit en langage C :

```c
#include <stdio.h>          // Requis pour appeler la fonction printf().

int main()
{
    float f1, f2, f3, r;    // IEEE754 simple precision pour f1, f2, f3 et r.

    f1 = 1E25;              // 10^25 -> f1
    f2 = 16;                // 16    -> f2

    f3 = f1 + f2;           // f1 + f2 -> f3
    r = f3 - f1;            // f3 - f1 -> r

    printf("r = %f\n", r);  // Affiche la valeur de r dans le terminal.

    return 0;               // Termine le programme.
}
```

Indication : $10^{25} \approx 2^{83}$

1. Quelle valeur de *r* sera affichée à la fin de l'exécution du programme ? Expliquez votre raisonnement.
2. Dans le programme, on a `f1=10`^{25}. Supposons maintenant que `f1=10`^{n} avec *n* entier positif. Jusqu'à quelle valeur de *n* un résultat correct apparaîtra-t-il sur *r* ?
3. Même question si les variables *f1*, *f2*, *f3* et *r* sont déclarées avec le mot clé `double` (à la place de `float`). Notez que le mot clé `double` signifie « IEEE754 double précision ».