T.D. 1
Systèmes de numération flottante

Exercice 1

Donnez la représentation flottante, en **simple précision**, des nombres suivants :

1. 128
2. –32,75
3. 18,125
4. 0,0625

Exercice 2

Donnez la représentation flottante, en **double précision**, des nombres suivants :

1. 1
2. –64
3. 12,06640625
4. 0,2734375

Exercice 3

Donnez la représentation décimale des nombres codés en **simple précision** suivants :

1. $1011\ 1101\ 0100\ 0000\ 0000\ 0000\ 0000\ 0000_2$
2. $0101\ 0101\ 0110\ 0000\ 0000\ 0000\ 0000\ 0000_2$
3. $1100\ 0001\ 1111\ 0000\ 0000\ 0000\ 0000\ 0000_2$
4. $1111\ 1111\ 1000\ 0000\ 0000\ 0000\ 0000\ 0000_2$
5. $0000\ 0000\ 0100\ 0000\ 0000\ 0000\ 0000\ 0000_2$

Exercice 4

Donnez la représentation décimale des nombres codés en **double précision** suivants :

1. $403D\ 4800\ 0000\ 0000_{16}$
2. $C040\ 0000\ 0000\ 0000_{16}$
3. $BFC0\ 0000\ 0000\ 0000_{16}$
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>

void main()
{
    float f1, f2, f3, r;

    f1 = 1E25;
    f2 = 16;

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

    printf("r = %f\n", r);
}
```

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

1. Quelle est la valeur de *r* qui est affichée à la fin de l'exécution de la fonction **`main()`** ? 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 en double précision.