
TD n° 5 - Examen 2021

Les exercices suivants sont tirés d'un examen de l'an dernier.

Rappels sur la norme IEEE 754 en simple précision

- 1 bit de signe (S)
- 8 bits d'exposant (E)
- 23 bits de mantisse (M)

Si le champ de l'exposant ne contient que des zéros (représentation dénormalisée), la valeur représentée est

$$(-1)^S \times 0, M \times 2^{-126}$$

Sinon (représentation normalisée), la valeur représentée est

$$(-1)^S \times 1, M \times 2^{E-127}$$

Exercice 1.

Conversions

Effectuez la conversion des valeurs indiquées de leurs bases d'origine vers les bases demandées. Détaillez vos démarches.

- 1 $510_{(7)} = ???_{(10)}$
- 2 $510_{(7)} = ???_{(6)}$
- 3 $260_{(10)} = ???_{(2)}$
- 4 $260_{(10)} = ???_{(16)}$
- 5 $1031_{(4)} = ???_{(10)}$
- 6 $1031_{(4)} = ???_{(16)}$

Exercice 2.

Opérations binaires

Soient les valeurs suivantes :

- $A = 110111_{(2)}$
- $B = 1101_{(2)}$
- $C = 101_{(2)}$

Posez les opérations binaires suivantes (indiquez les calculs binaires) :

- 1 $A + B$
- 2 $A - B$
- 3 $A \times C$
- 4 A/C

Exercice 3.

Utilisation des variables binaires

Dans le langage *JAVA*, les variables de type `int` (entiers signés) sont représentées sur 32 bits en utilisant le complément à deux si besoin. Les variables `float` correspondent à la norme IEE 754 en simple précision.

- `java.lang.Float.floatToIntBits(float a)` est une méthode qui retourne un `int` contenant la représentation 32 bits de son argument.

Par exemple, si $a = 2^{-146}$ (représentation dénormalisée), alors cette fonction retourne 8 (ce qui correspond à 0...001000).

Autre exemple : si $a = 1.0$, alors la valeur `int` retournée est 1065353216, correspondant aux bits

00111111100000000000000000000000

- La méthode `java.lang.Integer.FloatToHexString(int b)` renvoie une représentation de son argument b sous forme de chaîne (`String`) décrivant un entier en base 16.
Par exemple, si $b = -3$ (en binaire : 111111111111111111111111111101), alors la fonction retourne la chaîne « *FFFFFFFD* »
- Soient $p1$ et $p2$ deux variables de type `int`. $p1 \mid p2$ est une opération bit à bit qui retourne un `int` qui contient les bits résultats par OU bit à bit. L'opération $p1 \& p2$ correspond à l'opération bit à bit ET.
Par exemple :

```
int i = 1 ;// 00...0001 en binaire
int j= 2 ; // 00...0010 en binaire
i | j // représente 3 : 00... 0011, le résultat OU bit à bit
```

Dupont écrit un petit programme JAVA pour tester certaines variables :

```
import java.lang.*;
final public class Archi {
    public static void main(String[] args) {
        int i = 5;
        int j = 33;
        int k = i | j;           // OU bit à bit
        int m = i & j;           // ET bit à bit
        int n = (-i) & j;        // ET bit à bit
        System.out.println(k);
        System.out.println(m);
        System.out.println(n);
        float f = 3;
        int r = Float.floatToIntBits(f); // chaîne de 32 bits
        int p = 1024 * 1024 * 8;         // ici, p = 2^23
        r = r / p;                        // division entière
        System.out.println(r);
        float g = 56;
        int s = Float.floatToIntBits(g); // chaîne de 32 bits
        String str = Integer.toHexString(s); // chaîne hexadécimale
        System.out.println("Le nombre hexadécimal est : "+str);
    }
}
```

- 1 Représentez (sur 32 bits) les variables i , j , f et g .
- 2 Donnez (calculez) les valeurs imprimées (k , m , n , r , str) par ce programme. Détaillez vos calculs.

Exercice 4.

Conception d'un circuit indicateur

On aimerait construire un circuit qui indique si un chiffre décimal codé en binaire sur 4 bits est divisible par 3 ou non. Le circuit a 4 entrées (les quatre bits $s_3s_2s_1s_0$) et deux sorties D et E . La première (D) est égale à 1 si le chiffre est divisible par 3, et vaut 0 sinon (les chiffres 0, 3, 6 et 9 sont divisibles par 3). La deuxième sortie (E) est égale à 1 si les bits en entrée ne correspondent pas à un chiffre décimal entre 0 et 9 (elle indique une erreur). Quand les 4 entrées représentent un chiffre décimal, elle est à 0.

- 1 Proposez un circuit réalisé par des portes logiques NON, ET, OU et XOR.

Indication : À partir de la table de vérité (par exemple), établissez des expressions logiques pour les sorties D et E , puis proposez les circuits correspondants.

Exercice 5.

Questions de cours

- 1 Dans quels cas une addition de deux variables de type `int` peut causer un débordement (*overflow*) ? Comment détecter le débordement ? Quelles peuvent être les conséquences de ce débordement au niveau des valeurs obtenues ?
- 2 Quelle est la plus petite valeur négative représentable en simple précision dans la norme IEEE 754 ?