Exercices

Exercice ITC4.1 : Codage/décodage d'un entier positif [*]

1. Donner la valeur de entiers positifs suivants codés sur un octet : $b10011001$, $b00100001$, 11111111, 01010101.
2. Coder sur un octet les entiers positifs suivants : 57, 12, 45, 140.

Exercice ITC4.2 : Codage/décodage d'un entier relatif [*]

1. Donner la valeur de entiers relatifs suivants codés sur un octet : $b10011001$, $b00100001$, 11111111, 01010101.
2. Coder sur un octet les entiers relatifs suivants : 57, -12, 45, -100, 140.

Exercice ITC4.3 : Décodage de réels [*]

1. Donnez la valeur du réel de mantisse $m = b0001110001$ et d'exposant $e = b01111$
2. Donnez la valeur du réel de mantisse $m = b0011000001$ et d'exposant $e = b10100$
3. Donnez la valeur du réel de mantisse $m = b1010101010$ et d'exposant $e = b11000$

Exercice ITC4.4 : À propos du codage des entiers et des réels [*]

Exercice moins important, pour réfléchir.
Un entier signé int64 est codé sur 64 bits. Un flottant float64 a une mantisse de 52 bits et un exposant de 11 bits, les deux étant signés.

1. Quel est le plus grand entier positif qu'on peut coder dans un int64 ?
2. Quel est le plus grand réel positif qu'on peut coder dans un float64 ?
3. Quel est le plus petit réel strictement positif qu'on peut coder dans un float64 ?

Corrections des exercices

Corrigé de l'exercice ITC4.1 : Codage/décodage d'un entier positif [*]

1. Pour le premier : $b10011001 = 2^7 + 2^4 + 2^3 + 2^0 = 128 + 16 + 8 + 1 = 153$.
   Pour les autres, sans détail : 33, 255 et 85.
2. Pour le premier : $57 = 2 * 28 + 1$, puis $28 = 2 * 14 + 0$, puis $14 = 2 * 7 + 0$, puis $7 = 2 * 3 + 1$, puis $3 = 2 * 1 + 1$, puis $1 = 2 * 0 + 1$ donc $57 = b111001$ complété sur 1 octet en $b00111001$.
   Pour les autres, sans détail : $b00001010$, 00101101 et 10001100.

Corrigé de l'exercice ITC4.2 : Codage/décodage d'un entier relatif [*]

1. Quand le nombre commence par 0, c'est positif et la méthode est la même que dans l'exercice précédent ; si le nombre commence par 1, on calcule sa valeur absolue en faisant NOT puis +1. Par exemple, pour le premier : $NOT(b10011001) + 1 = b01100110 + 1 = b01100111 = 64 + 32 + 4 + 2 + 1 = 103$ donc le nombre vaut -103.
   Pour les autres, sans détail : 33, -1 et 85.
2. Pour le premier : ça ne change pas : $b00111001$.
   Pour le second, on code 12 en faisant $12 = 2*6+0$ puis $6 = 2*3+0$ puis $3 = 2*1+1$ puis $1 = 2*0+1$ donc $12 = b1100$ complété sur un octet en $b00001100$. On applique alors $NOT(b00001100) + 1 = b11110011 + 1 = b11110100$
   Pour les autres, sans détail : 00101101, $b10011100$; le dernier est un piège, car sur un octet on ne code des entiers qu'entre -128 et $+127$.

Corrigé de l'exercice ITC4.3 : Décodage de réels [*]

1. On calcule $m = 113$ et $e = 15$ donc ce réel vaut $113 \times 2^{15} = 3702784$
2. On calcule $m = 193$ et $e = -12$ donc ce réel vaut $193 * 2^{-12} = 0.047119140625$
3. On calcule $m = -342$ et $e = -8$ donc ce réel vaut $-342 \times 2^{-8} = -1.3359375$

Corrigé de l'exercice ITC4.4 : À propos du codage des entiers et des réels [*]

1. Pour un entier positif, le premier bit doit être un 0. Le plus grand entier positif est donc celui contenant un "0" suivi de 63 "1", qui vaut $\sum_{i=0}^{62} 2^i = 2^{63} - 1 = 9.2233720368548 \ldots .10^{+18}$ (je n'ai pas réussi à avoir la valeur exacte)
2. De même, la valeur positive maximale de la mantisse vaut $2^{51} - 1 = 2.2517998136852.10^{15}$ et celle de l'exposant $2^{10} - 1 = 1023$ ce qui donne comme plus grand réel $2.2517998136852.10^{15} \times 2^{1023} \approx 20.240225330731.10^{322}$
3. Pour un réel positif minimal, il faut $m = 1$ et $e = -1024$ (plus petite valeur codable) d'où $1.10^{-1024} = 5.562684646268e - 309$