Chapitre 4 : La chaîne d'information des systèmes

4.3 : Numération et codage des données

TD – 4 h

Travail demandé

1. Si la base n'est pas indiquée, à quelle(s) ***base(s) usuelle(s)*** (2, 8, 10 ou 16) peuvent appartenir les nombres :
a) 321 **b)** 1010 **c)** 3CA

2. Lister et compter ***tous les entiers*** inclus : **a)** entre $(287)_{16}$ et $(2A0)_{16}$ **b)** entre $(10)_2$ et $(1010)_2$

3. Quel est le ***poids*** des digits : **a)** de rang 9 et 10 en base 2 ? **b)** de rang 2 et 3 en base 16 ?

4. Selon que le type est signé ou non, déterminer les ***valeurs extrêmes*** codables d'un entier x sur :
a) 16 bits (entier « court », en langage C, **`short`**) ; **b)** sur 32 bits (entier « long », en langage C, **`long`**).
Pourquoi n'est-il pas nécessaire de disposer d'un codage d'entiers sur 64 bits ?

5. Convertir en ***base 2*** par la méthode des divisions, puis par développement en une somme de puissances de la base, les entiers : **a)** 34 **b)** 125

6. Idem que l'exercice 5 en ***base 16*** pour : **a)** 125 ; **b)** 517 ; **c)** 4012.
Comparer la commodité des deux méthodes de conversion sans utilisation d'une machine.

7. Convertir en ***base 10*** les entiers : **a)** $(10\ 0101)_2$ **b)** $(111\ 1011)_2$ **c)** $(1FD)_{16}$ **d)** $(FAC2)_{16}$
Vérifier les résultats en les comparants à ceux des exercices 5 & 6.

8. Convertir en ***bases 16 ou 2*** les entiers : **a)** $(10\ 0101)_2$ **b)** $(111\ 1011)_2$ **c)** $(1FD)_{16}$ **d)** $(AC8)_{16}$

9. Effectuer les ***additions*** suivantes directement dans la base de numération :
a) $(1101 + 1110)_2$ $(10\ 0101 + 111\ 1011)_2$ $(1111\ 1111 + 1)_2$
b) $(7E + 7C)_{16}$ $(1FF + FAC)_{16}$ $(FF + 1)_{16}$
Vérifier les résultats en base 10.

10. Effectuer en ***base 2*** les ***opérations*** : **a)** $(1101)_2 \times 2$ **b)** $(11011)_2 \div 2$
En déduire un algorithme général pour multiplier et diviser par b en base b.

11. Décoder en ***base 10*** les ***entiers signés*** codés sur ***8 bits*** : **a)** $(1110\ 0010)_{2\pm}$ **b)** $(1001\ 1100)_{2\pm}$

12. On montre qu'un entier x négatif sur n ***bits*** est codé $\boxed{(x)_{2\pm} = \overline{(|x|)}_2 + 1}$ où $\overline{(|x|)}_2$ est le code binaire naturel de la valeur absolue de x, ***complémenté*** bit par bit à 2 (chaque **`0`** devient **`1`** et inversement).
Avec cette formule, déterminer le ***code sur 8 bits*** des entiers : **a)** −28 **b)** −125

13. Convertir en **BCD** les entiers : **a)** 128 **b)** 517
À partir de leur code binaire, donner leur représentation hexadécimale. Que constate-t-on ?

14. Déterminer la ***date*** codée en **BCD** au format ***aaaa-mm-jj*** (i.e. « *année – mois – jour* ») ci-dessous :
`0001 1001 1000 0111 - 0000 0110 - 0010 0101`

15. **a)** Décoder la ***chaîne de caractère*** codée en **ASCII 7 bits** : **`53 74 6F 70 20 21 00`**.
b) Quel est le rôle du dernier caractère ?
c) En ASCII, quelle relation vérifient les codes respectifs de la majuscule et de la minuscule d'une lettre ?

16. * Soit $x = (d_k)_2$ un entier codé sur n bits en binaire naturel, et $(g_k)_G$ son ***code Gray***. On montre que :
pour $k = 0$ à $n - 1$, $\boxed{g_k = d_k \oplus d_{k+1}}$ et $\boxed{d_k = g_k \oplus d_{k+1}}$ avec $\boxed{a \oplus b = (a + b) \bmod 2}$ et $d_n = g_n = 0$.
a) Pour $n = 4$, déterminer dans l'ordre inverse des rangs des bits le code Gray des entiers $x = 5$ et $y = 10$.
b) Décoder les valeurs des entiers u et v codés par $u = (0100)_G$ et $v = (1011)_G$.
c) Vérifier les résultats en construisant le tableau des codes binaire naturel et Gray sur 4 bits.