UNIVERSITÉ de Marne-la-Vallée

Travaux Pratiques d'algorithmique n°1

Cours d'Informatique de Deuxième Année

—DEUG Sciences S3-MI—

Codage

L'objet de cette séance est l'étude et l'implémentation de différents codages pour les nombres entiers et réels.
Toutes les fonctions programmées devront être commentées.

▸ **Exercice 1.** *Écrire une fonction de prototype* `int Binaire (unsigned n, char *b)` *qui écrit dans* `b` *la décomposition binaire de l'entier* `n` *et retourne VRAI s'il est possible d'écrire* `n` *en base 2 sur 32 bits, FAUX sinon. Par exemple :*

pour `n` $= 21$*, sa décomposition binaire sera* 10101.

On donne ci-dessous un algorithme qui permet d'obtenir la décomposition binaire d'un nombre décimal :

On effectue la division entière de `n` *par 2. On trouve un quotient* q_0 *et un reste* r_0*. On recommence avec* q_0*. On construit ainsi une suite de restes* $(r_0, r_1, r_2, \ldots)$*. On arrête le processus dès que l'on trouve un quotient* q_k *égal à 0. La décomposition de* `n` *en base 2 est alors* $(r_k, r_{k-1}, \ldots, r_2, r_1, r_0)$*. Par exemple pour* $n = 10$ *:*

$$\begin{array}{ccccccc} 10 & = & 5 \times 2 & + & 0 & \longrightarrow & r_0 = 0 \\ 5 & = & 2 \times 2 & + & 1 & \longrightarrow & r_1 = 1 \\ 2 & = & 1 \times 2 & + & 0 & \longrightarrow & r_2 = 0 \\ 1 & = & 0 \times 2 & + & 1 & \longrightarrow & r_3 = 1 \end{array}$$

La représentation de 10 *en base* 2 *est donc* 1010.

▸ **Exercice 2.** *Écrire une fonction de prototype* `int Decimal (char *b)` *qui renvoie le nombre décimal dont la décomposition binaire est donnée par* `b`.

On donne ci-dessous un algorithme qui permet de passer de la décomposition binaire `b` *d'un nombre* `n` *à son écriture décimale :*

Soit une variable $d = 1$*. On commence par selectionner le chiffre* c_0 *situé le plus à droite dans le nombre* `b`*. On pose alors* `n`$= c_0 \times d$*. Puis on selectionne le chiffre* c_1 *situé à gauche de* c_0 *dans* `b`*. À chaque décalage vers la droite, on multiplie* `d` *par 2. Alors* `n`$=$`n`$+c_1 \times d$*. On* *continue ces décalages et le calcul de* `n` *associé jusqu'au dernier chiffre de* `b` *situé le plus à gauche dans* `b`*. Par exemple pour* `b=11010` *:*

$$\begin{array}{lcrcr} n & = & & & 0 \times 1 \\ n & = & 0 & + & 1 \times 2 \\ n & = & 2 & + & 0 \times 4 \\ n & = & 2 & + & 1 \times 8 \\ n & = & 10 & + & 1 \times 16 \end{array}$$

▸ **Exercice 3.**

- *Écrire une fonction de prototype* `int ComplementA2 (int n, char *c)` *qui écrit dans* `c` *la représentation en complément à 2 de l'entier* `n` *et retourne VRAI si c'est possible, FAUX sinon.*
- *Écrire une fonction de prototype* `int Compl2Decimal (char *c)` *qui renvoie l'entier dont la représentation en complément à 2 est donnée par* `c`*.*

▸ **Exercice 4.** *Écrire une fonction de prototype* `void AffDecomp2 (int n)` *qui affiche la représentation en machine d'un entier* `n`*. On utilisera pour cela les opérateurs bit à bit. Comparer cette représentation avec celle en complément à deux.*

▸ **Exercice 5.** *Rappelons tout d'abord qu'un réel se décompose dans le standard IEEE (simple précision, i.e. 32 bits) sous la forme suivante :*

$$X = (-1)^S \times 2^{E-127} \times (1 + M)$$

où S est le bit de signe ;
E est l'exposant (codé sur 8 bits) ;
M est la mantisse (codée sur 23 bits).

Le but de cet exercice est de représenter en écriture binaire un réel appartenant à $[0, 1[$. Ce nombre binaire sera stockée dans un tableau de 23 "bits".

- *Rappeler sur le papier le détail du calcul pour 0,8 ;*
- *écrire la fonction de prototype* `float Binaire2Reel(int *b)` *qui retourne le réel dont on connaît l'écriture binaire* `b` *passée en argument ;*
- *écrire la fonction de prototype* `int Reel2Binaire(float reel, int *b)` *qui code le nombre réel donné en argument, en écriture binaire. On rappelle que ce nombre doit nécessairement appartenir à $[0, 1[$.*

▸ **Exercice 6.** ***(IEEE)*** *La représentation IEEE d'un réel est donnée par : un "bit" pour le signe, un tableau de 8 "bits" pour l'exposant et enfin un tableau de 23 "bits" pour la mantisse (voir exercice précédent).*

Écrire la fonction qui renvoie le réel associé à une représentation IEEE. Écrire également la fonction qui code dans le standard IEEE un nombre réel quelconque.