

## T.P. 3

# 1 Valeur maximale d'un type de données

**Exercice 1.** L'objectif de cet exercice est d'écrire un programme qui affiche la valeur maximale d'un des types de données suivants :

- short int, unsigned short int
- long int, unsigned long int

Pour cela, écrivez un programme qui part d'un entier *var* de valeur 1 de type donné parmi ceux présentés précédemment, vous multipliez cet entier par 2 jusqu'à atteindre un dépassement de capacité, c'est à dire jusqu'à revenir à 0. Si pour cela, le nombre a été multiplié *n* fois (*n* représentant en réalité le nombre de bits), la valeur maximale est comme vu en TD par exemple pour un entier relatif codé sur *n + 1* bits égale à  $(2^n - 1)$ .

Dans votre programme vous devez afficher :

- la taille en bit de votre variable *var* (utiliser la fonction *sizeof()*)
- l'estimation de la valeur maximale que vous devriez atteindre selon la formule du TD
- puis les différentes valeurs intermédiaires atteintes au cours des multiplications *var \* 2*

Exemple d'affichage :

```
Ma variable 'var' est de type unsigned short int.
---> sa taille est de 2 octets, soit donc 16 bits
---> la valeur maximale estimee est : 2^16 -1 = 65535
***** Debut de la boucle (tant que la capacite n'est pas atteinte ) ***
valeur courante de ma variable 'var' : 1
valeur du compteur des bits : 1
-->sur les 1 bits atteints, la valeur maximale atteinte est= : 1
*****
valeur courante de ma variable 'var' : 2
valeur du compteur des bits : 2
-->sur les 2 bits atteints, la valeur maximale atteinte est= : 3
*****
.....
valeur courante de ma variable 'var' : 32768
valeur du compteur des bits : 16
sur les 16 bits atteints, la valeur maximale atteinte est= : 65535
***** Fin de la boucle (la capacite est pas atteinte ) *****
La valeur maximale pour 'var' est donc 65535 qui est egale a la valeur estimee.
```

# 2 Conversion de rationnels

**Exercice 2.** A l'image des exercices du TD1, convertissez un nombre décimal (c-à-d avec un nombre fini de chiffres après la virgule) saisi par l'utilisateur, de la base décimale à la base binaire et inversement.