

# TD 1 : boucles et tableaux

Programmation en C (LC4)

Semaine du 1<sup>er</sup> février 2008

**Exercice 1** Écrivez une fonction `void somme(int n, int tab[])` qui prend en argument un tableau d'entiers `tab` et sa taille `n` et qui affiche la somme des nombres qu'il contient. Résolvez ce problème :

- en utilisant `while`,
- en utilisant `do - while`,
- en utilisant `for`.

**Exercice 2** Écrivez une fonction `int inverse(int nb)` qui vérifie que `nb` est strictement compris entre 0 et 10000 et, le cas échéant, retourne `nb` à rebours. Par exemple, si `nb` vaut 1234, la fonction retourne 4321 (pensez à utiliser la fonction `modulo`).

**Exercice 3** Écrivez une fonction `int factorielle_boucle(int n)` qui calcule  $n!$  en utilisant une boucle. Écrivez ensuite une fonction `int factorielle_recursive(int n)` qui fait la même chose récursivement.

**Exercice 4** Écrivez une fonction `void seuil(int n, int tab[], int min)` qui remplace les nombres plus petits que `min` dans le tableau `tab` par `min`.

**Exercice 5** Écrivez une fonction `void tasse(int n, int tab[])` qui efface toutes les occurrences du chiffre 0 dans le tableau `tab` et tasse les éléments restants (on remplira les cases vides en fin de tableau par des zéros).

**Exercice 6** Écrivez une fonction `void inverse_tab(int n, int tab[])` qui range les éléments du tableau `tab` dans l'ordre inverse sans utiliser de tableau supplémentaire.

**Exercice 7** Écrivez une fonction `void fusion(int n, int tabA[], int m, int tabB[], int fus[])` qui prend en argument les tableaux `tabA` (de taille  $n$ ) et `tabB` (de taille  $m$ ) triés par ordre croissant et remplit le tableau `fus` (de taille  $n + m$ ) avec les éléments de `tabA` et `tabB` triés par ordre croissant.