



R. Malgouyres, R. Zrouer et F. Feschet  
Initiation à l'algorithmique et à la programmation en C,  
Cours avec 129 exercices corrigés,  
DUNOD, Collection Sciences Sup, 2011, 2<sup>e</sup> édition

## Algorithmique et programmation en C

### TP n° 6

## Passage par adresse et allocation dynamique

#### Objectifs :

Le but du TP est d'allouer dynamiquement des tableaux et de passer des variables par adresse dans des fonctions.

On considère le format de fichiers d'entiers suivant : Le premier nombre est le nombre d'entiers. Les nombres suivants représentent les entiers séparés par des espaces ou des passages à la ligne.

N  
n\_1 n\_2 ... n\_N

**Exercice 1** Écrire une fonction qui

- ouvre un fichier dont le nom est passé en paramètre (type `char*`) au format défini plus haut ;
- qui alloue dynamiquement un tableau d'entiers de la bonne taille ;
- qui lit les entiers dans le fichier et les met dans le tableau ;
- Qui retourne le tableau
- Qui effectue un passage par adresse du nombre d'éléments du tableau.

**Exercice 2** Écrire une fonction qui calcule la somme, le produit et la moyenne des entiers contenus dans un fichier.

**Exercice 3** Écrire une fonction qui prend en paramètre le nom de deux fichiers, l'un représentant des nombres  $n_1, \dots, n_N$ , l'autre des coefficients de pondération  $p_1, \dots, p_N$ . La fonction doit renvoyer deux valeurs. On fera donc un passage par adresse. La fonction doit donner la moyenne pondérée des  $n_i$  avec les coefficients de pondération  $p_i$ . La fonction doit renvoyer un code d'erreur au `main` en cas d'erreur d'ouverture d'un des deux fichiers ou si les deux fichiers n'ont pas le même nombre d'éléments.

**Exercice 4** Considérons l'ordre suivant entre les fichiers :

1. Si deux fichiers n'ont pas le même nombre d'éléments, le plus petit des deux fichiers est celui qui a le moins d'éléments :
2. Si deux fichiers  $f_1$  et  $f_2$  ont le même nombre d'éléments et leurs premiers éléments sont différents, le plus petit des deux fichiers est celui qui a le plus petit premier élément.
3. Si deux fichiers ont le même nombre d'éléments et ont le même premier élément, on compare les deuxièmes éléments, etc.. jusqu'à trouver un élément différent dans les deux fichiers. Le plus petit des deux fichiers est celui dont ce premier élément qui n'est pas le même dans les deux fichiers est le plus petit.

Écrire une fonction qui prend en paramètre deux noms de fichier et qui retourne  $-1$  si le premier est plus petit,  $0$  si les deux fichiers sont identiques, et  $+1$  si le deuxième fichier est plus petit. Le fonction doit renvoyer un code d'erreur dans une variable passée par adresse.