



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

# Devoir à la maison n° 3

## Chaînes de caractères, allocation et fichiers texte

### Objectifs :

Le but du problème est de pratiquer l'allocation dynamique de mémoire, le passage par adresses, les chaînes de caractères, et les fichiers texte.

Dans la base de données d'un commerce, chaque produit est représenté par une structure contenant :

- Le nom du produit (que l'on suppose d'au plus 100 caractères et pouvant contenir des espaces) ;
- Le code du produit (une chaîne de 5 caractères sans espaces) ;
- Le fournisseur (une chaîne d'au plus 20 caractères avec espaces) ;
- Le prix unitaire du produit ;
- Les quantités disponibles.

L'ensemble des produits de l'entreprise est stocké dans un fichier texte. On propose le format de fichier suivant :

- La première ligne du fichier contient le nombre de produits ;
- La suite du fichier contient les produits, chaque produit étant codé sur 5 lignes, chaque ligne contenant la donnée correspondant à un champ de la structure produit.

La figure 1 contient un exemple de fichier.

**Exercice 1** On se propose de charger en mémoire la base de données sous forme de tableau de structures. Dans ce tableau, chaque produit est représenté par une structure. Pour cela, on fera une fonction de chargement qui retourne le tableau de structures alloué à la taille lue dans la première ligne du fichier dont le nom est passé en paramètre. Pour chaque produit de la base de données, on lira une à une les données.

**Remarque 1.** On prendra garde que la fonction `fscanf`, utilisée avec le format `%s`, s'arrête au premier espace rencontré. La fonction `fgets`, par contre, lit toute une ligne dans le fichier, (mais elle met un `\n` à la fin de la chaîne).

**Remarque 2.** Lorsqu'on utilise la fonction `fscanf` avec les formes `%f` ou `%d` dans une ligne, le pointeur de fichiers ne passe pas le `\n` de la fin de la ligne. Il faut faire un `fgetc` ou un `fgetc` pour manger le retour chariot avant le prochain `fgets`.

**Exercice 2** Faire une fonction qui prend en paramètre un indice  $i$  et qui affiche les données du produit se trouvant en position  $i$  dans le tableau précédemment chargé en mémoire. La fonction renverra un code d'erreur en cas d'indice trop grand ou négatif.

**Exercice 3** En utilisant la fonction faite à l'exercice 2, faire une fonction d'affichage de toute la base de données.

**Exercice 4** Faire une fonction qui prend en paramètre un code (une chaîne de 5 caractères), qui recherche le produit ayant ce code dans le tableau des produits en mémoire, et qui retourne l'indice de ce produit dans le tableau.

**Remarque 3.** On pourra utiliser la fonction `strcmp`

**Exercice 5** En utilisant les fonctions des exercices 2 et 4, faire une fonction qui saisit un code au clavier et affiche les données du produit correspondant.

**Exercice 6** Faire une fonction qui affiche tous les produits fournis par un fournisseur saisi au clavier.

**Exercice 7** Faire une fonction de saisie d'un nouveau produit au clavier. On ajoutera le produit au tableau de structures en mémoire. Pour cela, si l'on a suivi l'énoncé et fait une allocation dynamique lors du chargement, il faut ré-allouer le tableau pour en modifier la taille. On procède en 5 étapes :

1. On alloue un nouveau tableau à la nouvelle taille ;
2. On recopie les données de l'ancien tableau dans le nouveau (utiliser `strcpy`) ;
3. On ajoute le nouvel élément au nouveau tableau ;
4. On détruit l'ancien tableau ;
5. On retourne le nouveau tableau.

**Exercice 8** Faire une fonction de sauvegarde des données du tableau dans un fichier.

```
2
Lessive qui lave plus blanc
LQLPB
Gamma LTD
3.50
28
Lessive qui respecte les couleurs
LQRLC
Erectus
4.99
35
```

FIGURE 1: Exemple de fichier