

## I2010 : langage C (19)

### Les fichiers

Reprenez le programme `biblio` écrit lors de la séance 15 et modifiez-le afin de sauvegarder les enregistrements dans un fichier et de récupérer un fichier précédent pour ajouter de nouvelles entrées avant de le sauvegarder à nouveau.

L'usage du programme sera :

```
biblioMaj [fichierIn] fichierOut
```

Si un seul nom de fichier est passé en argument c'est le nom du fichier (`fichierOut`) où les enregistrements seront sauvegardés, si deux noms sont présents, le premier est celui du fichier (`fichierIn`) qui contient déjà des enregistrements qu'il conviendra d'aller lire et le second celui du fichier (`fichierOut`) où le programme sauvegardera l'ensemble des enregistrements.

La structure des enregistrements est la même que celle qui a été définie pour le programme `biblio` à savoir :

1. Une chaîne de 128 caractères pour le titre
2. Une chaîne de 80 caractères pour le(s) auteur(s)
3. Un entier long pour l'ISBN (International Standard Book Number ou, en français, numéro international normalisé du livre)
4. Une chaîne de 50 caractères pour l'éditeur
5. Un entier pour l'année d'édition
6. Un type énuméré pour le genre sachant que les genres à prendre en considération pour cette application sont : « BandeDessinée », « Poésie », « Théâtre », « Roman », « RomanHistorique », « LittératureFrançaise », « LittératureEtrangère », « Sciences », « Informatique », « Science-fiction », « Santé », « Histoire »

Le programme doit

1. s'il a reçu deux arguments sur la ligne de commande, lire tous les enregistrements de `fichierIn`, les mettre en table
2. comme le faisait le programme `biblio`, tant que la fin de fichier n'est pas atteinte sur l'entrée standard
  - a. y lire des lignes d'au plus 256 caractères qui contiennent les informations concernant un livre
  - b. les décomposer et les stocker dans une structure telle que celle qui est définie ci-dessus
  - c. les placer dans la table à la suite des éléments qui y ont été mis au point 1

3. trier la table grâce à la fonction `qsort`, la clé de tri primaire sera l'année d'édition, les clés secondaires seront l'éditeur puis l'auteur
4. afficher la table triée
5. écrire tous les éléments de la table triée dans le fichier dont le nom (`fichierOut`) est passé en argument sur la ligne de commande

La taille de la table doit être gérée dynamiquement en fonction du nombre de lignes lues.

Si le programme est appelé avec moins d'un argument ou plus de deux, il affiche l'usage sur la sortie d'erreur et s'arrête avec un code d'erreur égal à 1.

Si le `fichierIn` n'existe pas ou n'est pas ouvrable en lecture, un message d'erreur explicite est affiché à la sortie d'erreur et le programme se termine avec un code d'erreur égal à 10

Si le `fichierOut` n'existe pas, il est créé sinon il est ouvert en écriture et écrasé. S'il n'est pas possible de l'ouvrir en écriture, un message d'erreur spécifique est affiché à la sortie d'erreur et le programme se termine avec un code d'erreur égal à 11.

Si le programme rencontre des problèmes pour lire ou pour écrire dans un fichier l'utilisateur doit être averti par un message qui précise le type d'erreur avant que le programme ne se termine prématurément avec un code d'erreur à 12 (pour un problème en lecture) ou 13 (pour un problème en écriture).

Les éventuels problèmes d'allocation de la mémoire doivent aussi être signalés et provoquer l'arrêt du programme avec un code d'erreur à 20.

Si une ligne lue à l'entrée standard n'est pas conforme au format attendu, un message d'erreur explicite est affiché à la sortie d'erreur et la ligne est ignorée.