Travaux Pratiques de langage C n°6

Cours d'Informatique de l'École Centrale d'Électronique

—Ingénieur Première Année—

Le but de ce TP est de gérer un petit répertoire d'étudiants en s'appuyant sur les techniques liées aux listes chaînées. Ce répertoire est destiné aux enseignants en informatique. De fait, les opérations à réaliser seront notamment les suivantes :

- la saisie des coordonnées d'un étudiant ;
- l'ajout d'un étudiant dans le répertoire ;
- la suppression d'un étudiant du répertoire ;
- la visualisation et le tri du répertoire.

Dans ce TP, nous utilisons aussi les fonctions d'entrée-sortie sur fichier de la bibliothèque standard. Ces fonctions nous permettent de faire des opérations sur un fichier, par exemple la lecture, l'écriture ou encore la création. Nous allons donc voir :

- comment ouvrir un fichier pour accéder à son contenu ;
- comment parcourir ce contenu ;
- comment créer un fichier et le remplir.

▸ **Exercice 1. (saisie des coordonnées d'un étudiant)** *Un étudiant est caractérisé par les informations suivantes :*

- *son nom ;*
- *son prénom ;*
- *sa classe ;*
- *son groupe ;*
- *ses notes avec leurs coefficients.*

Créer le type de données `Etudiant` *approprié, puis écrire une fonction* `SaisirEtudiant` *qui saisit les coordonnées d'un étudiant.*

▸ **Exercice 2. (gestion du répertoire)** *Créer le type de données* `Rep` *approprié, puis écrire les fonctions* `CreerRep` *et* `DetruireRep` *qui respectivement créent et détruisent un répertoire.*

▸ **Exercice 3. (gestion d'un étudiant dans le répertoire)** *Écrire les fonctions* `AjoutEtudiant` *et* `SupprimeEtudiant` *qui respectivement créent et détruisent un étudiant.*

▸ **Exercice 4. (gestion des notes d'un étudiant)** *Écrire la fonction* `AjoutNote` *qui permet d'ajouter une note affectée d'un coefficient donné pour un étudiant donné.*

▸ **Exercice 5. (tri du répertoire)** *En vous servant d'un des algorithmes de tri vus en cours, donner une fonction* `TrierRep` *de tri du répertoire.*

▸ **Exercice 6. (affichage du répertoire)** *Écrire la fonction* `AfficheRep` *qui affiche le contenu du répertoire à l'écran.*

▸ **Exercice 7. (recherche d'un étudiant dans le répertoire)** *Écrire la fonction* `RechercheRep` *qui permet de rechercher un (ou plusieurs) étudiant(s) en fonction de ses caractéristiques (son nom, son prénom, etc.). Une manière de faire est de réutiliser le module de saisie des coordonnées d'un étudiant. Les informations saisies sont alors les critères de recherche. Si un champ de saisie n'est pas un critère de recherche, alors il suffira de taper sur retour chariot. On devra pouvoir visualiser simplement les réponses multiples.*

L'objet de la suite du TP est la réalisation effective du répertoire d'étudiants. Pour cela, nous allons devoir créer un petit logiciel de gestion du répertoire (lecture, sauvegarde sur fichier, etc.).

▸ **Exercice 8. (sauvegarde du répertoire)** *Écrire une fonction* `SauveRep` *qui permet de sauver un répertoire d'étudiants dans un fichier donné sous un format de votre choix.*

▸ **Exercice 9. (lecture du répertoire)** *Écrire une fonction* `LectureRep` *qui permet de lire un répertoire d'étudiants sauvé sur fichier selon le format choisi précédemment.*

▸ **Exercice 10. (petit langage de manipulation du répertoire)** *On souhaite désormais réaliser le logiciel de manipulation de répertoire. Pour cela, on demande de créer un programme simulant une ligne de commande.*

Une ligne de commande est composée d'un prompt *et de l'instruction rentrée par l'utilisateur ; le* prompt *affiché automatiquement en début de chaque instruction signale à l'utilisateur qu'il peut entrer une commande. Les commandes correspondent par exemple à une opération programmée dans les exercices précédents mais il peut aussi sagir d'une demande pour quitter le logiciel ou obtenir de l'aide sur son utilisation.*

Votre logiciel devra inclure les fonctionnalités de base utiles à la manipulation d'un répertoire d'étudiants ainsi que toutes les options que vous jugerez intéressantes.