



■■■■ Manipulation de fichiers

- 1 – Écrire un programme qui compte le nombre de lignes d'un fichier sur disque.
- 2 – Écrire un programme qui effectue l'écriture de données dans un fichier texte (organisées sous forme de lignes de textes, où chaque ligne est composée de caractères et terminée par un `\n`).
Ces informations sont le *nom*, *prénoms* et *adresse* d'un individu.
Elles sont saisies au clavier et enregistrées dans un fichier (« `carnet_adresse.txt` »):
 - une ligne par individu,
 - chaque ligne composée de la manière suivante :
`nom : prénoms : adresse`
- 3 – Écrire un programme qui ouvre un premier fichier et crée un nouveau fichier contenant une ligne sur deux du premier fichier.

■■■■ Gestion des listes

- 4 – Écrire un programme prenant la liste des fichiers contenus dans un répertoire, et qui ouvre et affiche la première ligne de chacun de ces fichiers.
La liste des fichiers d'un répertoire est obtenue par l'instruction : `ls *.py`
- 5 – Écrire un programme qui réalise l'insertion d'un élément à un emplacement donné par son indice.

■■■■ Utilisation des structures

- 6 – Écrire un programme lisant des données dans un fichier binaire enregistré suivant la structure C suivante :

```
struct {  
    int numero ;  
    char nom[10];  
    float moyenne; }
```

et qui les sauve dans un fichier texte sous la forme : `nom, numéro, moyenne`

■■■■ Utilisation des expressions rationnelles ou *Regular Expression*

- 7 – Écrire un programme qui récupère les informations enregistrées par le programme de l'exercice 2 (nom, prénom et adresse de chaque individu) et crée un nouveau fichier.
Pour chaque ligne :
 - affiche ces informations sur l'écran ;
 - change les délimiteurs « : » en « ; » ;
 - passe chaque information tout en majuscule ;
 - écrit ces nouvelles informations dans le nouveau fichier.Le fichier original est réécrit dans un nouveau fichier dont le nom est « `nom_original.traduit` ».

8 – Écrire un programme qui ouvre un fichier structuré de la manière suivante :

```
Nom = un_nom    # un_commentaire
Prenom = un_prenom                    # un_commentaire
Nom machine = un_nom_machine          # un_commentaire
Mot de passe = un_mot_de_passe        # un_commentaire
Champ_supplémentaire= une_valeur
...
Nom = un_nom    # un_commentaire
Prénom = un_prénom                    # un_commentaire
Champ_supplémentaire= une_valeur
```

Ce programme doit créer deux fichiers, « nom_original.name » et « nom_original.password », le premier contenant la liste des *noms*, et le second la liste des *mots de passe*.

Remarques :

- les lignes indiquant le champ « nom » et « mot de passe » ne sont pas séparés toujours par le même nombre de lignes.
- à chaque fois que l'on rencontre le champ « nom », on commence le groupe de lignes d'un nouvel individu.

■■■■■ Utilisation des opérations d'éclatement et de recomposition (split & join)

9 – Écrire un programme affichant les statistiques d'un fichier texte :

- nombre de caractères, de mots et de lignes ;
- taille moyenne des mots en caractère, taille moyenne en mot des lignes.

■■■■■ Manipulation des dictionnaires

- 10 – Écrire un programme qui détermine le nombre d'occurrence de chaque mot d'un fichier texte.
- 11 – Écrire un programme qui effectue la lecture et la construction d'un dictionnaire à partir du contenu d'un fichier d'options structuré de la façon suivante :

```
nom_option = valeur_option
nom_option_2 = valeur_option_2
...
```

Afficher la liste des associations (options, valeur).

■■■■■ Mise en œuvre des instructions de manipulation binaire

12 – Écrire un programme qui :

- ouvre un fichier texte en lecture ;
- ouvre en écriture un nouveau fichier ;
- traite chaque caractère de ce fichier de la manière suivante :
 - ◇ décomposition de ce caractère en sa représentation binaire
 - ◇ échange des bits de rang 3 & 4;
 - ◇ recomposition de cette représentation binaire en un caractère;
 - ◇ écriture de ce caractère dans le fichier.

13 – Écrire un programme réalisant l'encodage base64 d'un fichier conformément à la RFC 2045.