
Fiche TP N°1

Introduction aux fichiers

Exercice 1 :

Considérons le type enregistrement suivant :

```
Type Étudiant = Enregistrement
    Matricule : entier
    Nom, Prénom : chaîne [20]
    Moyenne : réel
Fin
```

Soit un tableau **T** de 100 étudiants.

- Écrire un programme permettant de recopier tous les étudiants admis appartenant à **T** dans un fichier **ADMIS** de **Étudiant**. Un étudiant est admis si sa moyenne est supérieure ou égale 10.

Exercice2 :

- 1) Ecrire une fonction qui supprime le dernier élément du fichier **F** de nombre entiers.
- 2) En utilisant la fonction précédente, écrire un programme pour vider un fichier **F** existant de nombre entiers et de nom physique 'ESSAI.DAT', élément par élément. Ce programme devra afficher, après la suppression de chaque élément, la moyenne des éléments restants de **F**.

Exercice3 :

Soit **CAR** un fichier de caractères contenant des mots séparés par un ou plusieurs espaces.

- Écrire une fonction **PAL** qui vérifie si un mot donné est un mot palindrome.
- En utilisant la fonction **PAL**, écrire un programme qui affiche le nombre de mots palindromes et le plus court mot palindrome.

Exercice 4 :

Soient **F1** et **F2** deux fichiers d'entiers strictement positifs et sans répétition.

1. Écrire un programme qui construit un fichier **G** d'entiers tel que **G** contient pour chaque valeur de **F1** la valeur et tous ses multiples appartenant à **F2**.

Exemple :

F1 : 3 10 20 17

F2 : 3 6 19 60 40 30

G : 3 3 6 60 30 10 60 40 30 20 60 40 17

2. Écrire un programme qui permet à partir du fichier résultat **G** de générer un autre fichier **H** contenant toutes les valeurs du fichier **G** (sans répétition) avec leur nombre.

Exemple :

H : 3 2 6 1 60 3 30 2 10 1 40 2 20 1 17 1

Exercice5 :

Soit un fichier **F** contenant les produits de différents magasins. Chaque élément du fichier **F** est constitué de :

Type Magasin = Enregistrement

CodeMagasin, CodeProduit: entier ;

PrixAchat, PrixVente : reel ;

QteVendue : entier ; // Quantite vendue

Fin

1. Pour un magasin (CodeMagasin donné), écrire une fonction **MaxBenefice** permettant de déterminer le produit (CodeProduit) ayant réalisé le meilleur bénéfice et la valeur de ce bénéfice.
2. En utilisant la fonction **MaxBenefice**, écrire une fonction permettant de créer un fichier **G** contenant pour chaque magasin le produit ayant le meilleur bénéfice et la valeur de ce bénéfice. Chaque élément du fichier **G** est structuré comme suit : CodeMagasin, CodeProduit et MaxBnf.
3. Ecrire une fonction **TrieG** permettant de trier le fichier **G** dans l'ordre décroissant selon le bénéfice réalisé.

N.B. : La formule de calcul de bénéfice est : (PrixVentre - PrixAchat) * QteVendue