



## Module : Structure de fichiers & Structure de données TP1 : Manipulation des fichiers

**Note :** Les programmes de cette série de travaux pratiques doivent être élaborés en langage C.

### Partie I : Manipulation des fichiers textes

#### Exercice 1 :

Écrire un programme qui lit le contenu d'un fichier texte et l'affiche sur la console par :

1. *Méthode 1* : lecture du fichier caractère par caractère
2. *Méthode 2* : lecture du fichier ligne par ligne.

#### Exercice 2 :

1. Écrire un programme qui prend une entrée utilisateur et la sauvegarde dans un fichier.
2. Modifier le programme pour copier le texte fourni 10,000 fois, créant ainsi un fichier texte étendu avec 10,000 lignes.
3. Modifier le programme de l'exercice 1 pour mesurer le temps nécessaire à la lecture du fichier créé. Comparez les performances entre les deux méthodes de lecture.

#### Exercice 3 :

Écrire un programme qui compte le nombre de mots dans un fichier texte donné.

#### Exercice 4 :

Écrire un programme qui copie le contenu d'un fichier texte source dans un fichier de destination.

### Partie II : Écriture de données structurées dans un fichier

#### Exercice 5 :

Considérons une structure qui représente des employés, comprenant leur nom, leur numéro d'employé et leur salaire.

1. Écrire un programme qui permet de recueillir les informations de plusieurs employés via la console, puis de les enregistrer dans un fichier.
2. Écrire un programme pour lire les données des employés à partir d'un fichier et les afficher à l'écran.

#### Exercice 6 :

Créer un programme de gestion de stock en utilisant des structures pour représenter les articles (nom, prix, quantité en stock). Le programme doit permettre à l'utilisateur, depuis un menu, d'ajouter, de modifier et de supprimer des articles et d'enregistrer les données dans un fichier. Utiliser le principe d'allocation dynamique pour la manipulation des données du fichier. Voici le format de Menu désiré :

Menu :  
1. Ajouter un article  
2. Modifier un article  
3. Supprimer un article  
4. Afficher tous les articles  
5. Quitter  
Choix :

### Partie III : Manipulation des fichiers de types spécifiques

#### Exercice 7 :

Écrire un programme qui lit un fichier CSV contenant des données de contact (nom, numéro de téléphone, email) et affiche ces données à l'écran.

#### Exercice 8 :

Écrire un programme qui permet de lire un fichier HTML donné par l'utilisateur, d'extraire toutes les balises HTML de ce fichier, et de les afficher à l'écran.

#### Exercice 9 :

Écrire un programme qui permet de lire et d'afficher le contenu d'un fichier exécutable (.exe) en mode hexadécimal. Le programme doit afficher 16 octets par ligne.

**Note :** Utiliser le fichier exécutable de l'un des exercices précédents.

#### Exercice 10 :

Écrire un programme qui lit un fichier Bitmap et affiche ses métadonnées, ainsi que les valeurs des 10 premiers pixels de l'image. Un fichier Bitmap est structuré comme suit :

#### Partie I. Métadonnées : Constituée de deux parties :

En-tête de fichier :

1. Signature (2 octets)
2. Taille totale du fichier en octets (4 octets)
3. Champ réservé (4 octets)
4. Offset de l'image (4 octets)

En-tête DIB (Device Independent Bitmap) :

1. Taille de l'en-tête de l'image en octets (4 octets)
2. Largeur de l'image (4 octets)
3. Hauteur de l'image (4 octets)
4. Nombre de plans (2 octets)
5. Profondeur de codage des couleurs (2 octets)
6. Méthode de compression (4 octets)
7. Taille totale de l'image en octets (4 octets)
8. Résolution horizontale (4 octets)
9. Résolution verticale (4 octets)
10. Nombre de couleurs de la palette (4 octets)
11. Nombre de couleurs importantes de la palette (4 octets)

**Partie II. Contenu de l'image :** Les pixels de l'image, sous forme de valeurs de trois couleurs de base codées en binaire.