



TD1

Exercice 1

Écrire un programme qui :

1. Crée une liste de nombres entiers saisis par l'utilisateur.
2. Affiche :
 - o Liste triée sans utiliser sort() ni sorted(),
 - o Deuxième plus grand élément,
 - o Somme des éléments aux indices pairs,
 - o Nouvelle liste contenant uniquement les valeurs uniques (sans utiliser set()).
3. Remplace dans la liste tous les éléments négatifs par 0.

Exercice 2 :

1. Créer un tuple contenant des entiers et des chaînes.
2. Tenter de modifier un élément et observer le message d'erreur.
3. Construire un **nouveau tuple** issu du précédent mais :
 - o Sans doublons,
 - o Trié selon la longueur des éléments (pour les chaînes).
4. Convertir ce tuple en liste, insérer un nouvel élément au milieu, puis reconvertis en tuple.

Exercice 3 :

Écrire un programme qui :

1. Lit une phrase saisie par l'utilisateur.
2. Affiche :
 - o Nombre de mots,
 - o Mot le plus long et le plus court,
 - o Fréquence de chaque lettre (sans collections.Counter),

- Phrase inversée mot par mot (sans split() inversé directement).

3. Vérifie si la phrase est un **pangramme** (contient toutes les lettres de l'alphabet).

Exercice 4 :

1. Saisir deux ensembles de nombres (par exemple A et B).

2. Afficher :

- Eléments communs (intersection),
- Eléments uniques à A et à B,
- Eléments pairs dans l'union des deux ensembles,
- Eléments présents dans un seul ensemble.

3. Construire un nouvel ensemble contenant les **carrés** de tous les nombres présents dans $A \cup B$.

Exercice 5 :

Écrire un programme qui :

1. Permet à l'utilisateur de saisir plusieurs paires nom : note.

2. Construit un dictionnaire associant chaque nom à sa note.

3. Affiche :

- Moyenne des notes,
- Le ou les étudiants ayant la meilleure note,
- Liste des étudiants triée par note décroissante (sans utiliser sorted() directement).

4. Supprime toutes les entrées dont la note est inférieure à la moyenne.

Exercice 6 :

Créer un petit programme qui :

1. Contient une liste de tuples :

`etudiants = [("Ali", 15), ("Sara", 18), ("Nadia", 12), ("Youssef", 17)]`

2. Construit :

- Dictionnaire notes où la clé est le nom et la valeur la note,
- Ensemble contenant les étudiants ayant obtenu une note ≥ 15 ,
- Liste triée des noms selon les notes décroissantes (sans utiliser sorted()).

3. Affiche les structures résultantes.