



TD2

Exercice 1 :

1. Écrivez un programme qui :
 - a. Demande une phrase à l'utilisateur.
 - b. Crée une fonction **analyse_texte(texte)** qui renvoie :
 - Nombre de mots,
 - Mot le plus long,
 - Fréquence de chaque mot (insensible à la casse).
 - c. Affiche le tout sous forme lisible.
2. Ajoutez un paramètre booléen *ignore_mots_vides* permettant d'exclure des mots comme "le", "la", "et", "de", etc.

Exercice 2 :

1. Écrivez une fonction **statistiques(liste)** qui renvoie un tuple contenant :
 - Moyenne,
 - Médiane,
 - Min et Max,
 - Écart-type (sans librairie externe).

Puis écrivez un programme qui génère une liste de n nombres aléatoires (entre 0 et 100) et affiche les résultats formatés.

2. Ajoutez une fonction **filtrer(liste, borne_min, borne_max)** pour ne garder que les valeurs comprises entre ces bornes avant de recalculer les stats.

Exercice 3 :

1. Un inventaire est une **liste de dictionnaires**, chaque dictionnaire représentant un produit avec les clés :

`{"id": int, "nom": str, "prix": float, "quantité": int}`

- a. Créez une fonction **ajouter_produit(inventaire, id, nom, prix, quantite)** qui ajoute un produit.
 - b. Créez une fonction **supprimer_produit(inventaire, id)** qui retire un produit par son identifiant.
 - c. Créez une fonction **valeur_totale(inventaire)** qui renvoie la valeur totale du stock.
 - d. Créez une fonction **produits_en_rupture(inventaire)** qui renvoie la liste des produits dont la quantité = 0.
2. Créez une fonction **recherche(inventaire, mot_cle)** qui renvoie les produits dont le nom contient le mot-clé.

Exercice 4 :

1. Écrire une fonction **est_palindrome(chaine)** qui retourne un booléen.
2. Écrire une fonction **sont_anagrammes(mot1, mot2)** qui retourne un booléen, en utilisant des ensembles ou des dictionnaires de fréquences.
3. Écrire une fonction **groupes_anagrammes(liste_mots)** qui regroupe les mots anagrammes dans une **liste de sous-listes**.

Exercice 5 :

1. On dispose d'un dictionnaire :

```

classement = {
    "Equipe A": {"victoires": 5, "defaites": 2, "nuls": 1},
    "Equipe B": {"victoires": 4, "defaites": 1, "nuls": 3},
    ...
}

```

- a. Créez une fonction **calcul_points(equipe)** qui retourne le nombre de points (victoire = 3 pts, nul = 1 pt).
 - b. Créez une fonction **trier_classement(classement)** qui renvoie une liste triée des équipes selon le nombre de points décroissant.
 - c. Créez une fonction **meilleure_equipe(classement)** qui retourne le nom de l'équipe la mieux classée.
2. Afficher un tableau formaté avec alignement du texte et des colonnes.

Exercice 6 :

On dispose de trois ensembles :

```
maths = {"Ali", "Sara", "Mounir", "Yassine"}
```

```
physique = {"Sara", "Amine", "Yassine"}
```

```
informatique = {"Ali", "Yassine", "Amal"}
```

1. Créez une fonction **etudiants_comuns(*ensembles)** qui renvoie les étudiants inscrits dans toutes les matières.
2. Créez une fonction **etudiants_uniques(*ensembles)** qui renvoie ceux inscrits dans une seule matière.
3. Créez une fonction **statistiques_groupes()** qui affiche le nombre d'étudiants par matière et le total unique.