

REPUBLIQUE TUNISIENNE

Ministère de l'enseignement Supérieure, de la
recherche scientifique



الجمهورية التونسية

وزارة التعليم العالي
والبحث العلمي

Institut Préparatoire aux Etudes d'Ingénieurs de
Sfax

المعهد التحضيري للدراسات الهندسية
بصفاقس

Epreuve d'informatique

2^{ème} semestre AU : 2023 – 2024

Section : BG1

Date : Mai 2024

Durée : 2H

Nombre de page : 4

L'utilisation des calculatrices n'est pas autorisée pour cette épreuve.

Le langage de programmation sera obligatoirement Python.

Exercice 1

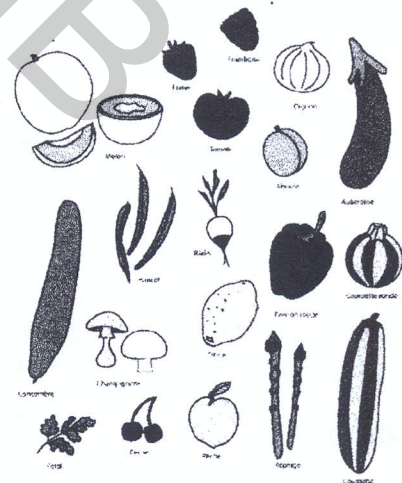
L'objectif de cet exercice est d'analyser des données végétaux (nom, couleur, famille, ...) stockées dans un fichier texte nommé 'vegetaux.txt' où chaque ligne est de la forme :

« Nom, couleur, famille »

Exemple :

```
##### vegetaux
Fichier  Modifier  Affichage

Carotte,Orange,Apiaceae
Tomate,Rouge,Solanaceae
Laitue,Vert,Asteraceae
Pomme de terre,Marron,Solanaceae
Banane,Jaune,Musaceae
Brocoli,Vert,Brassicaceae
```



Question 1 :

Ecrire une fonction **lire_Fichier()** qui lit le fichier 'vegetaux.txt' ligne par ligne, remplit et renvoie un dictionnaire où :

- Les clés sont les noms des végétaux
- Les valeurs sont des tuples de la forme (couleur, famille)

Question 2 :

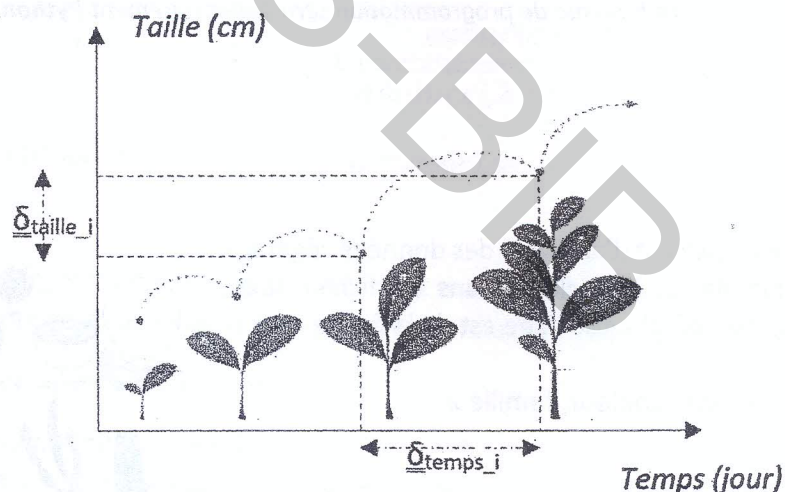
Ecrire une fonction **chercher_famille (nom)** qui prend en paramètre le nom du végétal et qui renvoie sa famille (en se basant sur le dictionnaire de la question1).

Question 3 :

Ecrire une fonction **analyse(famille)** qui prend en paramètre la famille du végétal et qui renvoie le nombre des végétaux de cette famille (en se basant sur le dictionnaire de la question1).

Exercice 2

L'objectif de cet exercice est d'analyser la croissance des plantes.



Question 1 :

Ecrire une fonction **delta(L)** qui prend en paramètre une liste triée et qui renvoie la somme des deltas :

$$\delta_L = \sum_{i=1}^{n-1} L[i] - L[i - 1]$$

Question 2 :

Ecrire une fonction **taux_croissance(L_temps, L_taille)** qui prend en paramètres la liste des temps et la liste des tailles d'une plante. Cette fonction renvoie le taux de croissance de la plante :

$$\text{Taux} = \delta_{L_taille} / \delta_{L_temps}$$

Question 3 :

Ecrire une fonction **croissance_logistique(t, taille, taux, t0)** qui prend en paramètres :

- **t** : un instant t (en jour)
- **taille** : la taille maximum de la plante (en cm)
- **taux** : le taux de croissance
- **t0** : un instant initial (en jour)

Cette fonction renvoie la taille de la plante dans un instant t selon la fonction suivante :

$$\text{taille}_t = \frac{\text{taille}}{(1 + e^{(-\text{taux} * (t - t_0))})}$$

Indication : utiliser la fonction $\exp(x)$ du module numpy pour calculer e^x

Question 4 :

Ecrire une fonction **tracer_courbe_croissance(taux, taille)** qui prend en paramètres le taux de croissance et la taille maximum de la plante. Cette fonction trace la courbe de croissance pour les 30 premiers jours tel que :

- l'axe des abscisses est l'axe du temps de 0 à 30 (jours)
- l'axe des ordonnées est l'axe des tailles pour chaque jour (en cm) (la taille est calculée à partir de la question 3 , avec $t_0 = 0$)

indication : utiliser la fonction $\text{plot}(L_x, L_y)$ du module matplotlib.pyplot

Exercice 3 :

L'objectif de cet exercice est de traiter la corrélation des différentes caractéristiques des animaux.

Par exemple :

Animal	Poids (kg)	Taille (cm)	Âge (années)	Consommation alimentaire (g/jour)	Température corporelle (°C)	Fréquence cardiaque (battements/min)
Chat	4.5	25	3	150	38.5	120
Chien	10	50	5	300	39	100
Lapin	2	20	2	100	38	150

Ces données seront représentées sous forme d'une matrice numpy de la forme :

$$A = \begin{pmatrix} 4.5 & 25 & 3 & 150 & 38.5 & 120 \\ 10 & 50 & 5 & 300 & 39 & 100 \\ 2 & 20 & 2 & 100 & 38 & 150 \end{pmatrix}$$

Où chaque colonne représente une caractéristique et chaque ligne représente un animal.

Question 1 :

Ecrire une fonction **variance(A, j)** qui calcule la variance de la colonne d'indice j dans la matrice A.

Indication : utiliser la fonction `var(X)` du module `numpy` pour calculer la variance d'un vecteur X.

Question 2 :

Ecrire la fonction **covariance(A, i, j)** qui calcule la covariance des colonnes i et j dans la matrice A.

Indication : utiliser la fonction `cov(X,Y)` du module `numpy` qui renvoie une matrice carrée d'ordre 2 de la forme :

$$\begin{pmatrix} cov(X,X) & cov(X,Y) \\ cov(Y,X) & cov(Y,Y) \end{pmatrix}$$

Question 3 :

Ecrire la fonction **correlation(A, i, j)** qui calcule la corrélation des colonnes i et j dans la matrice A, selon la formule suivante :

$$correlation(X,Y) = \frac{covariance(X,Y)}{\sqrt{variance(X)} * \sqrt{variance(Y)}}$$

Indication : utiliser la fonction `sqrt(X)` du module `math` pour calculer : $\sqrt{X}$

Question 4 :

Ecrire la fonction **Matrice_correlation (A)** qui remplit et renvoie une matrice carrée M d'ordre n , avec n est le nombre de colonne de A, où chaque M_{ij} égale à la corrélation des colonnes i et j dans A.

Indication : vous pouvez suivre cet algorithme :

- Récupérer le nombre de colonne de A
- Créer une matrice M carrée d'ordre n rempli de zéro
- Pour chaque colonne i dans A :
 - o Pour chaque colonne j dans A :
 - Calculer la corrélation des colonnes i et j
 - Affecter cette corrélation à M_{ij}
- Renvoyer M.