

BACCALAURÉAT

SESSION 2026

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°4

DURÉE DE L'ÉPREUVE : 1 heure

**Le sujet comporte 4 pages numérotées de 1 / 4 à 4 / 4
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

Cette situation d'évaluation comporte ce document ainsi que des fichiers de codes et de données présents sur l'ordinateur à la disposition du candidat. Le candidat doit restituer ce document avant de sortir de la salle d'examen. Le candidat doit agir en autonomie et faire preuve d'initiative tout au long de l'épreuve.

En cas de difficulté, le candidat peut solliciter l'examineur afin de lui permettre de continuer la tâche.

Des moments privilégiés pour solliciter l'examineur sont indiqués dans le document sous la forme d'appels professeur.

L'examineur peut intervenir à tout moment, s'il le juge utile.

Une équipe d'agronomes souhaite étudier la croissance de différentes plantes cultivées en serre afin de déterminer l'impact des conditions environnementales sur leur développement. Chaque jour, les chercheurs mesurent la hauteur de chaque plante, la température ambiante et l'humidité de la serre.

L'objectif de ce sujet est de concevoir un programme permettant d'analyser ces données afin de dégager des tendances sur la croissance des plantes. Deux fichiers Python sont fournis :

- plantes.py qui contient la description des différentes plantes cultivées ;
- mesures.py qui contient les relevés journaliers effectués dans la serre.

Le fichier plantes.py présente les plantes étudiées sous la forme de liste d'objet plantes. La chaque objet de type plantes est une instance de la classe Plante. Chaque instance de la classe plante contient les attributs suivant :

- son nom ;
- son espece ;
- sa durée moyenne de croissance (en jours) ;
- sa taille moyenne (en cm) ;
- son type d'exposition à la lumière parmi "ombre", "mi-ombre" ou "plein soleil".

Le fichier mesures.py contient les données collectées pendant plusieurs jours sous la forme d'une liste de dictionnaires. Chaque dictionnaire correspond à une mesure quotidienne. Chaque mesure comporte les champs suivants :

- plante nom d'une plante figurant dans plantes.py ;
- jour entier représentant le numéro du jour de culture ;
- hauteur entier correspondant à la hauteur de la plante en centimètres ;
- temperature entier représentant la température moyenne du jour (en °C) ;
- humidite entier représentant le taux d'humidité (en %) mesuré dans la serre.

Question 1

Écrire une fonction `croissance_moyenne(plantes)` qui prend en paramètre une liste d'instances de la classe `Plante` et renvoie la moyenne des durées de croissance de l'ensemble de ces plantes (en jours). Si la liste fournie est vide, la fonction doit renvoyer `None`.

Écrire au moins deux tests pour valider le bon fonctionnement de cette fonction, dont une traitant le cas d'une liste vide.

Hand-paper

Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

Question 2

Écrire une fonction `dictionnaire_mesure(plantes, mesures)` qui prend en paramètre la liste des plantes et la liste des mesures. Elle doit renvoyer un dictionnaire où chaque clé est le nom d'une plante (présente dans la liste `plantes`), et chaque valeur associée est la liste des mesures concernant cette plante spécifique.

Si une plante de la liste ne possède aucune mesure associée, la liste correspondante dans le dictionnaire devra être vide. Concevoir une série de tests pertinente pour vérifier le bon comportement de cette fonction. Exemple de retour :

```

{
  "plante1": [
    {
      "plante": "plante1",
      "jour": 1,
      "hauteur": 100,
      "temperature": 20,
      "humidite": 100,
    },
    {
      "plante": "plante1",
      "jour": 2,
      "hauteur": 101,
      "temperature": 20,
      "humidite": 100,
    },
  ],
  "plante2": [
    {
      "plante": "plante2",
      "jour": 1,
      "hauteur": 10,
      "temperature": 20,
      "humidite": 20,
    }
  ],
}

```

Afin de ne conserver que les données correspondant à des conditions climatiques idéales, un chercheur a rédigé la fonction `purger_mesures_extremes(liste_mesures)`. Cette fonction, fournie dans le fichier `culture.py`, est censée retirer de la liste toutes les mesures ayant été prises à une température strictement inférieure à 20°C ou strictement supérieure à 25°C. Cependant, lorsqu'on exécute cette fonction sur les données, on constate qu'un certain nombre de relevés pris à des températures extrêmes sont toujours présents dans la liste finale.

Question 3

Exécuter le test de la fonction `test_purger` et analyser son code pour identifier la source de cette erreur logique.

Question 4

Proposer une version corrigée de la fonction répondant parfaitement à l'objectif.

Hand paper

Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

Description du dossier :

Le dossier fourni au candidat comporte les éléments suivants :

- une version PDF de l'énoncé ;
- un fichier culture.py à compléter ;
- un jeu de données de plantes plantes.py ;
- un jeu de données de mesures mesures.py.

Correction

Exercice 1:

```
def croissance_moyenne(plantes):
```

```
    if plantes == []:
        return None
    total = 0
    plante_t = len(plantes)
    for plante in plantes:
        total += plante.croissance
    return total / plante_t
```

```
assert croissance_moyenne(plantes) == 395 / 5
assert croissance_moyenne([]) == None
```

Exercice 2:

```
def dictionnaire_mesure(plantes, mesures):
    dict1 = {}
    for plante in plantes:
        for mesure in mesures:
            if mesure['plante'] == plante.nom:
                if plante.nom in dict1:
                    dict1[plante.nom].append(mesure)
                else:
                    dict1[plante.nom] = [mesure]
            else:
                if not plante.nom in dict1:
                    dict1[plante.nom] = []
    return dict1
```

Exercice 3:

juste à un teste certaines valeur renvoyer avait une température entre 20 et 25, hors la fonction ne devrait pas, l'erreur vient de la condition du if

Exercice 4:

```
def purger_mesures_extremes(liste_mesures):  
    """  
    Supprime de la liste toutes les mesures dont la température  
    n'est pas comprise entre 20 et 25°C inclus.  
    """  
    for mesure in liste_mesures:  
        if mesure['temperature'] > 20 or mesure['temperature'] < 25:  
            liste_mesures.remove(mesure)  
    return liste_mesures
```