

# **BACCALAURÉAT**

**SESSION 2026**

---

**Épreuve de l'enseignement de spécialité**

## **NUMÉRIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°5**

---

**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.

On s'intéresse ici à la notion d'**empreinte carbone**, qui correspond à la production de CO<sub>2</sub> (un gaz à effet de serre) imputable à un individu ou à un groupe pendant une année, exprimé en kilogrammes. Dans le cadre de la lutte contre le réchauffement climatique, le site [nosgestesclimat.fr](https://nosgestesclimat.fr) propose un simulateur permettant d'obtenir une estimation de son empreinte carbone en répondant à des questions sur sa situation et sa consommation. Les résultats sont compilés dans un dictionnaire qui associe à des catégories (logement, alimentation etc.) l'empreinte carbone associée.

Une utilisatrice prénommée Ada a réalisé une simulation de son empreinte carbone, dont les résultats vous sont donnés dans les fichiers `empreinte_ada.json` sous format brute et `empreinte_ada_aggr.json` sous forme agrégée. Le JSON est un format de fichier permettant de stocker des listes ou des dictionnaires dont les clés soient des chaînes de caractères. Le module `json` permet de convertir des valeurs Python en `json` et réciproquement.

Dans un premier temps, on considère les résultats sous forme agrégée représentés par un simple dictionnaire associant à chaque catégorie un entier représentant l'empreinte carbone correspondante (en kilogrammes de CO<sub>2</sub>).

```
{
    "Logement": 2660,
    "Alimentation": 1500,
    ...
}
```

### Question 1

Compléter le corps de la fonction `total_simple`. Ajouter un test permettant d'afficher l'empreinte carbone totale d'Ada, en utilisant les fonctions fournies pour accéder au fichier `empreinte_ada_aggr.json`.



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

Le fichier réel obtenu par Ada est `empreinte_ada.json` qui contient une imbrication de dictionnaires qui permet de préciser les postes de consommation. Par exemple :

```
{
    "Logement": {
        "Energie": {
            "Electricité": 206,
            "Cuisson": 105,
```

```

        "Chauffage individuel": 1500
    },
    "Construction": 650,
    "Location": 37,
    "Ameublement": 162
},
...
}

```

### Question 2

En utilisant la fonction `est_dictionnaire` qui teste la nature d'une valeur, écrire le corps de la fonction récursive `total_rec` qui calcule la somme des valeurs numériques présentes dans des dictionnaires imbriqués. Des tests sont fournis dans la fonction `test_total_rec`, on pourra les compléter par un cas plus proche de la structure présente dans `empreinte_ada.json`.



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

Ada souhaite intégrer une fonctionnalité d'alerte. Il s'agit d'identifier si une source d'émission spécifique dépasse un certain seuil jugé critique.

Une fonction nommée `alerte_valeur_aberrante(empreinte, limite)` a été rédigée à cet effet et figure dans le fichier Python fourni. Elle est censée parcourir l'ensemble du dictionnaire, y compris les sous-catégories, et renvoyer la valeur booléenne `True` dès qu'une valeur strictement supérieure à la limite est rencontrée.

### Question 3

Lorsqu'on exécute la fonction `alerte_valeur_aberrante` sur le dictionnaire complet d'Ada avec une limite fixée à 1000, elle ne détecte aucune valeur aberrante, alors que le poste "Chauffage individuel" s'élève à 1500. Expliquer précisément l'origine de cette erreur de conception et proposer une version corrigée de la fonction.



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

### Question 4

Afin de prévenir toute régression future sur la fonction `alerte_valeur_aberrante` corrigée, il convient de définir une stratégie de validation robuste. Proposer un jeu de tests pertinent pour cette fonction. Pour chaque cas de test envisagé, préciser la structure du dictionnaire fourni en entrée, le résultat attendu et la particularité algorithmique que ce test permet de vérifier.



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

## Description du dossier

Le dossier fourni au candidat sur l'ordinateur comporte les éléments suivants :

- une version PDF de l'énoncé ;
- un code source de départ `empreinte.py` ;
- les deux fichiers `empreinte_ada.json` et `empreinte_ada_agr.json`.

## Préparation de l'environnement

Pour faire fonctionner le code fourni dans le dossier, les bibliothèques suivantes doivent être présentes : `json`.