

# **BACCALAURÉAT**

**SESSION 2026**

---

**Épreuve de l'enseignement de spécialité**

## **NUMÉRIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°23**

---

**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.

L'entreprise *PlovaNuviam* développe un module d'acquisition *DEMETER*, de transmission et de traitement de données environnementales (température, humidité...) pour la prévention de maladies dans les cultures agricoles comme les vignes.

Plusieurs fois par heure, chaque module mesure la température et l'humidité et transmet les données par ondes radio. Ces données sont réceptionnées sur un serveur dont le rôle est de vérifier, d'analyser et de traiter les informations.

Le but de ce sujet est de finaliser la programmation du serveur et de répondre à certaines problématiques.

Un code Python à corriger et à compléter ainsi qu'un jeu de données sont fournis avec ce document.

### Description des données envoyées

Les données mesurées par chacun des modules *DEMETER* sont envoyées par trames de 40 bits, dont la forme est la suivante :

ID (8 bits)	Clé de sécurité (8 bits)	Température (12 bits)	Humidité (8 bits)	Bits de contrôle (4 bits)
00101010	11001000	010010001100	01100010	1101

Pour une trame donnée notée *trame* :

- les 8 premiers bits *trame*[0:8] correspondent à l'identifiant (ID) du module en binaire classique ;
- les 8 bits suivants *trame*[8:16] sont une clé de sécurité ;
- les 12 bits suivants *trame*[16:28] encodent la température. Pour l'obtenir en °C, il faut convertir cette valeur binaire en décimal, lui soustraire 900, puis diviser le tout par 10. Par exemple, 010010001100 vaut 1164 en décimal, ce qui donne  $(1164 - 900) / 10 = 26.4$  °C ;
- les 8 bits suivants *trame*[28:36] encodent l'humidité en pourcentage. Attention, elle est codée en Décimal Codé Binaire (BCD) : chaque chiffre décimal est codé séparément sur 4 bits. Par exemple, 01100010 se sépare en 0110 (6) et 0010 (2), ce qui donne 62 %. Exception : 100% est codé 10100000 ;

- les 4 derniers bits `trame[36:40]` sont des bits de parité permettant de vérifier l'intégrité des 4 premiers blocs. Le bit de parité vaut 0 si le nombre de '1' dans le bloc correspondant est pair, sinon il vaut 1. Par exemple, pour l'ID 00101010, le nombre de '1' est impair (3), donc le premier bit de contrôle est 1.

*Indication :* en Python, l'expression `int(chaine, 2)` prend une chaîne de caractère contenant l'écriture binaire d'un entier et renvoie cet entier.

#### Question 1

Dans le fichier `transmission.py`, écrire en autonomie la fonction `decoder_temperature(trame)` qui prend en paramètre une chaîne de caractères représentant une trame et renvoie la température sous forme de flottant. Écrire ensuite la fonction `decoder_humidite(trame)` qui renvoie l'humidité sous forme d'entier. Écrire des tests pour vérifier que ces fonctions renvoient bien 26.4 et 62 avec la trame d'exemple de l'énoncé.



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

#### Question 2

Écrire la fonction `est_valide(trame)` qui prend une trame en paramètre et renvoie `True` si les 4 bits de contrôle correspondent bien à la parité des 4 blocs de données, et `False` sinon.



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

#### Question 3

Le fichier `data.txt` contient une série de plus de 300 transmissions réelles. Dans le fichier `analyse.py`, une fonction d'analyse a été écrite pour automatiser le traitement et l'affichage des données valides. Exécuter le programme et constater qu'il produit un message d'erreur. Analyser ce plantage. Identifier sa cause en observant, entre autres, le contenu du fichier `data.txt`.

#### Question 4

Proposer des corrections de la classe `Transmission` permettant de la rendre plus robuste afin de pouvoir effectuer l'analyse.



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 `transmission.py`, `analyse.py` ;
- un fichier `data.txt`.

### **Préparation de l'environnement**

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