

# **BACCALAURÉAT**

**SESSION 2026**

---

**Épreuve de l'enseignement de spécialité**

## **NUMÉRIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°17**

---

**DURÉE DE L'ÉPREUVE : 1 heure**

**Le sujet comporte 3 pages numérotées de 1 / 3 à 3 / 3  
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.

Un club de handball professionnel souhaite mieux comprendre et gérer ses dépenses et recettes annuelles. Les dirigeants veulent identifier les postes de dépenses les plus importants et s'assurer de l'exactitude de leur solde budgétaire de fin d'année.

On dispose de jeux de données sous forme de fichiers CSV représentant les mouvements budgétaires du club. Chaque mouvement est décrit par un enregistrement comportant les attributs suivants :

- 'type' : chaîne de caractères dont les valeurs possibles sont 'dépense' ou 'recette' ;
- 'catégorie' : chaîne de caractères dont les valeurs possibles sont 'subventions', 'sponsoring', 'billetterie', 'marketing', 'salaires', 'déplacements', 'fonctionnement', 'cotisations' ou 'autres\_charges' ;
- 'montant' : nombre flottant positif représentant la somme en euros ;
- 'mois' : nombre entier compris entre 1 et 12.

Une fois chargé par le script `analyse_budget.py`, ces jeux de données donnent une liste de dictionnaires mouvements.

Deux jeux de données sont fournis :

- un jeu de test défini directement dans le code dans la variable `mouvements_test` donne la liste de dictionnaires reproduite ci-dessous :

```
mouvements_test = [  
    {'type': 'recette', 'catégorie': 'cotisations', 'montant': 1200.0,  
↪   'mois': 1},  
    {'type': 'recette', 'catégorie': 'billetterie', 'montant': 300.0,  
↪   'mois': 6},  
    {'type': 'dépense', 'catégorie': 'fonctionnement', 'montant': 450.0,  
↪   'mois': 6},  
    {'type': 'dépense', 'catégorie': 'déplacements', 'montant': 200.0,  
↪   'mois': 12},  
    {'type': 'dépense', 'catégorie': 'salaires', 'montant': 1500.0,  
↪   'mois': 12},  
    {'type': 'recette', 'catégorie': 'subventions', 'montant': 800.0,  
↪   'mois': 12}  
]
```

- un jeu complet `budget_complet.csv` de plus de 2000 mouvements.

Le fichier `analyse_budget.py` contient plusieurs fonctions d'analyse que le sujet vise à compléter.

### Question 1

Écrire la fonction `total_par_type(mouvements, type_mouvement)` qui prend en paramètres une liste de mouvements et une chaîne de caractères (parmi 'dépense' ou 'recette'). Cette fonction doit renvoyer la somme totale des montants correspondant à ce type précis. La fonction renverra 0 si aucun mouvement ne correspond.

Créer ensuite une fonction `test_total()` contenant au moins deux assertions pour valider le bon fonctionnement de votre code sur le jeu de données `mouvements_test` (par exemple, le total des recettes attendu est de 2300.0).



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

La fonction `solde_annuel(mouvements)`, déjà codée dans le fichier fourni, est censée calculer le bilan financier de l'année en faisant la somme des soldes de chaque mois (recettes totales moins dépenses totales). Si le solde annuel est positif, le club dégage des bénéfices. S'il est négatif, le club est en déficit

### Question 2

Calculer manuellement le solde annuel attendu pour la liste `mouvements_test`. Écrire ensuite une fonction `test_solde_annuel()` contenant une assertion qui vérifie que la fonction `solde_annuel(mouvements_test)` renvoie bien ce résultat théorique. Exécuter ce test.

L'exécution du test précédent lève une erreur : le résultat renvoyé par la fonction `solde_annuel` ne correspond pas à la réalité comptable.

### Question 3

Analyser le code de la fonction `solde_annuel` pour identifier la source de cette erreur logique, expliquer pourquoi certains mouvements ont été ignorés, puis proposer une correction. Appliquer ensuite votre fonction corrigée sur le fichier `budget_complet.csv` pour annoncer le véritable solde du club au professeur.



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

## Description du dossier fourni

Le dossier mis à disposition du candidat contient :

- une version PDF de cet énoncé ;
- un code source à analyser et compléter : `analyse_budget.py` ;
- un jeu de données complet plus de 2000 mouvements : `budget_complet.csv`.