

# **BACCALAURÉAT**

**SESSION 2026**

---

**Épreuve de l'enseignement de spécialité**

## **NUMÉRIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°21**

---

**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.

Les cartes mémoire (ou *flashcards*) sont des supports de révision comportant au recto une question et au verso la réponse. Elles permettent des révisions actives très efficaces.

On souhaite développer une application utilisant le système des boîtes de *Leitner*, qui permet de gérer ces cartes en se basant sur un principe de répétition espacée.

Dans cette méthode, chaque carte possède un niveau d'avancement (de 0 à 4). Plus le niveau est élevé, plus le délai avant la prochaine révision est grand. Les délais sont donnés par le tableau suivant :

Niveau de la carte	Délai avant révision
0	1 jour
1	3 jours
2	7 jours
3	15 jours
4	30 jours

En pratique, la méthode fonctionne ainsi : lorsque l'utilisateur révise une carte, s'il répond correctement, la carte passe au niveau supérieur (sans dépasser le niveau 4 maximum). S'il se trompe, la carte retombe immédiatement au niveau 0, quel que soit son niveau précédent. La prochaine date de révision est ensuite calculée en ajoutant le délai du nouveau niveau à la date du jour.

On modélise ce fonctionnement à l'aide de la Programmation Orientée Objet. Dans le fichier `cartes.py`, la classe `Carte` a été commencée.

### Question 1

Écrire la méthode `traiter_reponse(self, succes)` qui prend en paramètre un booléen `succes` (`True` si l'utilisateur a bien répondu, `False` sinon). Cette méthode doit mettre à jour l'attribut `self.niveau` de la carte selon les règles de *Leitner* énoncées accessible dans la liste globale `DELAIS`, puis calculer et mettre à jour l'attribut `self.date_prochaine`.

*Indication : Pour ajouter des jours à la date d'aujourd'hui, on utilisera la fonction fournie*

`date_future(nb_jours)` qui renvoie la date située `nb_jours` après aujourd'hui.



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

On considère que la base de révision, le *paquet* de cartes, est une liste d'instances de la classe *Carte*.

### Question 2

Écrire une fonction `extraire_cartes_du_jour(paquet, date_jour)` qui prend en paramètres une liste de cartes `paquet` et une date de référence `date_jour` et qui renvoie une nouvelle liste contenant uniquement les cartes dont la `date_prochaine` est inférieure ou égale à `date_jour`.

On admet qu'on peut comparer des dates avec les opérateurs usuels `<`, `<=`, `==`, `>=` et `>`.



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

Afin d'aider l'étudiant à cibler ses lacunes, on souhaite extraire du *paquet* les cartes qui lui posent le plus de problèmes, c'est-à-dire celles dont le niveau est le plus bas parmi toutes les cartes du *paquet*.

La fonction `extraire_cartes_a_renforcer(paquet)` a été rédigée dans ce but. Cependant, elle contient une faille logique.

### Question 3

Exécuter la fonction `test_renforcement()` fournie. Observer le résultat affiché dans la console et constater l'incohérence.

Analyser le code de la fonction `extraire_cartes_a_renforcer(paquet)`, identifier la source de cette erreur logique, puis corriger le code afin qu'il ne renvoie que les cartes possédant rigoureusement le niveau minimum.



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 `cartes.py`.