

EXERCICE 3 (8 points)

Cet exercice porte sur la programmation Python, la modularité, les bases de données relationnelles et les requêtes SQL.

Une *flashcard*, autrement appelée *carte de mémorisation*, est une carte papier sur laquelle se trouve au recto une question et au verso la réponse à cette question. On les utilise en lisant la question du recto puis en vérifiant notre réponse à celle du verso. Une étudiante souhaite réaliser des *flashcards* numériquement.

Partie A

L'étudiante souhaite stocker les questions/réponses de ses *flashcards* dans un fichier au format `csv`. Ce format permet de stocker textuellement des données tabulaires. La première ligne du fichier contient les descripteurs : les noms des champs renseignés par la suite. Pour être en mesure de les identifier, chaque champ est séparé par un caractère appelé séparateur. C'est la virgule qui est le plus couramment utilisée, mais cela peut être d'autres caractères de ponctuation.

Le langage Python dispose d'un module natif nommé `csv` qui permet de traiter de tels fichiers. La méthode `DictReader` de ce module prend en argument un fichier `csv` et le séparateur utilisé. Elle permet d'extraire les données contenues dans le fichier. Voici un exemple de fonctionnement.

fichier `exemple.csv`

```
champ1, champ2
a, 7
b, 8
c, 9
```

code Python

```
import csv
with open('exemple.csv', 'r') as fichier:
    donnees = list(csv.DictReader(fichier, delimiter=','))
print(donnees)
```

affichage généré à l'exécution

```
[{'champ1': 'a', 'champ2': '7'},
 {'champ1': 'b', 'champ2': '8'},
 {'champ1': 'c', 'champ2': '9'}]
```

Voici un extrait du fichier `flashcards.csv` réalisé par l'étudiante :

```
discipline; chapitre; question; réponse
histoire; crise de 1929; jeudi noir - date; 24 octobre 1929
histoire; crise de 1929; jeudi noir - quoi; krach boursier
histoire; 2GM; l'Axe; Allemagne, Italie, Japon
histoire; 2GM; les Alliés; Chine, États-Unis, France, Royaume-Uni, URSS
```

histoire;2GM;Pearl Harbor - date;7 décembre 1941
philosophie;travail;Marx;aliénation de l'ouvrier
philosophie;travail;Beauvoir;donne de la valeur à l'homme
philosophie;travail;Locke;permet de fonder le droit de propriété
philosophie;travail;Crawford;satisfaction et estime de soi

1. Donner le séparateur choisi par l'étudiante pour son fichier `flashcards.csv`.
2. Justifier pourquoi l'étudiante a choisi ce séparateur.

Voici le code écrit par l'étudiante pour utiliser ses flashcards.

```
1  import csv
2  import time
3
4  def charger(nom_fichier):
5      with ...
6          donnees = ...
7      return ...
8
9  def choix_discipline(donnees):
10     disciplines = []
11     for i in range(len(donnees)):
12         disc = donnees[i]['discipline']
13         if not disc in disciplines:
14             disciplines.append(disc)
15     for i in range(len(disciplines)):
16         print(i + 1, disciplines[i])
17     num_disc = int(input('numéro de la discipline ? '))
18     return disciplines[num_disc - 1]
19
20 def choix_chapitre(donnees, disc):
21     chapitres = []
22     for i in range(len(donnees)):
23         if flashcard[i]['discipline'] == disc:
24             ch = flashcard[i]['chapitre']
25             if not ch in chapitres:
26                 chapitres.append(ch)
27     for i in range(len(chapitres)):
28         print(i + 1, chapitres[i])
29     num_ch = int(input('numéro du chapitre ? '))
30     return chapitres[num_ch - 1]
31
32 def entrainement(donnees, disc, ch):
33     for i in range(len(donnees)):
34         if donnees[i]['discipline'] == disc \
35         and donnees[i]['chapitre'] == ch:
36             print('QUESTION : ', donnees[i]['question'])
37             time.sleep(5)
38             print(donnees[i]['réponse'])
```

```

39             time.sleep(1)
40
41 flashcard = ...
42 d = ...
43 c = ...
44 entraînement(...)

```

3. Recopier et compléter le code de la fonction `charger(nom_fichier)` qui lit le fichier dont le nom est fourni en argument et qui renvoie les données lues sous la forme d'un dictionnaire comme dans l'exemple fourni précédemment.
4. Le module `time` est importé à la ligne 2 de ce programme. Quelle est la méthode du module `time` utilisée dans ce code ?
5. Donner le type de la variable `donnees[i]` (par exemple ligne 12).
6. Recopier et compléter les lignes 41 à 44.

Partie B

Pour améliorer sa mémorisation sur le long terme, l'étudiante décide de mettre en œuvre le concept des boîtes de Leitner. Dans cette méthode, il s'agit d'espacer dans le temps la révision des *flashcards* si l'étudiante répond correctement. Elle imagine donc une base de données qui lui permettra de conserver pour chaque question la date à laquelle elle doit de nouveau être posée. Elle décide que les questions seront réparties en 5 boîtes. Initialement, tous les questions seront placées dans la boîte 1. Les questions de la boîte 1 sont posées tous les jours, celles de la boîte 2 tous les deux jours, celles de la boîte 3 tous les quatre jours, celles de la boîte 4 tous les huit jours et celles de la boîte 5 tous les quinze jours. Si l'étudiante donne la bonne réponse à une question et que la question n'appartient pas à la boîte 5, son numéro de boîte est incrémenté (augmenté de 1). Si l'étudiante ne donne pas la bonne réponse, la question revient dans la boîte 1.

Elle met en œuvre une base de données relationnelle contenant 4 tables `discipline`, `chapitre`, `boite` et `question`.

La table `discipline` contient la liste des disciplines étudiées. Elle a deux attributs :

- `id`, de type `INT`, l'identifiant de la discipline qui est une clé primaire pour cette table ;
- `lib`, de type `TEXT`, le libellé de la discipline.

La table `chapitre` contient la liste des chapitres des disciplines étudiées. Elle a trois attributs :

- `id`, de type `INT`, l'identifiant du chapitre qui est une clé primaire pour cette table ;

- `lib`, de type `TEXT`, le libellé du chapitre ;
- `id_disc`, de type `INT`, l'identifiant de la discipline à laquelle appartient ce chapitre.

La table `boite` contient l'ensemble des cinq boites existantes. Elle a trois attributs :

- `id`, de type `INT`, l'identifiant numéro de la boite qui est une clé primaire pour cette table ;
- `lib`, de type `TEXT`, le libellé de la boite ;
- `frequence`, de type `INT`, indiquant le nombre de jours séparant deux interrogations d'une question appartenant à cette boite.

La table `flashcard` contient les questions-réponses. Elle a six attributs :

- `id`, de type `INT`, l'identifiant de la *flashcard* qui est une clé primaire pour cette table ;
- `id_ch`, de type `INT`, l'identifiant du chapitre auquel appartient la *flashcard* ;
- `id_boite`, de type `INT`, l'identifiant numéro de la boite de la *flashcard* ;
- `question`, de type `TEXT`, le texte au recto de la *flashcard* ;
- `reponse`, de type `TEXT`, le texte au verso de la *flashcard* ;
- `date_interro`, de type `DATE`, la date de la prochaine interrogation pour cette question.

Initialement `date_interro` sera la date d'insertion de la question dans la base de données.

Table boite		
Id	lib	frequence
1	tous les jours	1
2	tous les deux jours	2
3	tous les quatre jours	4
4	tous les huit jours	8

7. Écrire une requête SQL qui complète la table `boite` et insère la boite 5 de libellé 'tous les quinze jours' et de fréquence 15.

Une requête sur la table `flashcard` affiche l'enregistrement suivant :

5, 2, 1, Pearl Harbor - date, 6 décembre 1941

8. Écrire une requête SQL pour mettre à jour la date de Pearl Harbor renvoyée. La bonne date est le 7 décembre 1941.
9. Écrire une requête SQL qui permet d'obtenir la liste des libellés des disciplines.
10. Écrire une requête SQL qui permet d'obtenir la liste des libellés des chapitres de la discipline 'histoire'.
11. Écrire une requête SQL qui permet d'obtenir la liste des identifiants des flashcards de la discipline 'histoire'.
12. Écrire une requête SQL pour supprimer toutes les flashcards de la boîte d'identifiant 3.