# Devoir Surveillé

Traitements de données en tables

Durée : 2 heures — Barème indicatif sur 20 points

*La calculatrice n'est pas autorisée. L'usage de tout document ou appareil électronique est interdit.*
*La qualité de la rédaction et la clarté des explications seront prises en compte.*

**Nom :** ..... **Prénom :** ..... **Classe :** .....

## Exercice 1 – Vocabulaire et format CSV *(4 pts)*

On considère l'extrait ci-dessous d'un fichier `meteo.csv` contenant des relevés météorologiques quotidiens dans plusieurs villes :

```
ville;date;temperature;humidite;vent
Paris;2025-01-15;3.2;82;12
Lyon;2025-01-15;1.8;75;8
Marseille;2025-01-15;8.5;60;25
Paris;2025-01-16;2.1;88;15
Lyon;2025-01-16;0.5;79;6
Marseille;2025-01-16;9.1;58;30
```

1. Que signifie l'acronyme CSV ? Quel est le délimiteur utilisé dans ce fichier ? *(1 pt)*
2. Donner la liste des **attributs** (ou descripteurs) de cette table. Combien d'**enregistrements** contient-elle ? *(1 pt)*
3. Pour chaque attribut, préciser son **domaine de valeurs** (type et ensemble de valeurs possibles). *(1 pt)*
4. On souhaite ajouter un nouvel enregistrement correspondant à un relevé à Toulouse le 15 janvier 2025 avec une température de 5.3řC, une humidité de 70% et un vent de 18 km/h. Écrire la ligne à ajouter au fichier CSV. *(1 pt)*

## Exercice 2 – Lecture et indexation d'un fichier CSV en Python *(4 pts)*

On souhaite exploiter le fichier `meteo.csv` décrit dans l'exercice 1.

1. Compléter le code suivant qui lit le fichier `meteo.csv` et stocke les données dans un **tableau de tableaux** (sans utiliser le module `csv`). *(1,5 pt)*

```
fichier = open('meteo.csv', mode='r', encoding=______, newline='')
attributs = fichier.____________.rstrip().___________(______)
table = [ligne.____________.____________(______) for ligne in fichier]
fichier.close()
```

2. On utilise maintenant le module `csv` pour obtenir un **tableau de dictionnaires**. Compléter le code suivant : *(1 pt)*

```
import csv

f = open('meteo.csv', mode='r', encoding='utf8', newline='')
reader = csv.____________(f, delimiter=______)
```

```
table = [enregistrement for enregistrement in reader]
f.close()
```

**3.** Après exécution du code de la question 2, quelle est la valeur de chacune des expressions suivantes ? *(1 pt)*

a) `len(table)`
b) `table[0]['ville']`
c) `table[2]['temperature']`
d) `type(table[2]['temperature'])`

**4.** Expliquer pourquoi, pour comparer des températures stockées dans cette table, il est nécessaire d'effectuer une conversion. Quelle fonction Python utiliser pour cette conversion ? *(0,5 pt,5>1 s)*

## Exercice 3 – Sélection et projection *(4,5 pt)*

On suppose que le fichier `meteo.csv` a été correctement chargé dans un tableau de dictionnaires nommé `table`, comme à la question 2 de l'exercice 2.

**1. Sélection**

a) Écrire, en utilisant **une boucle `for`**, les instructions Python permettant de construire un tableau `releves_paris` contenant tous les enregistrements correspondant à la ville de Paris. *(1 pt)*
b) Écrire, en utilisant **une liste en compréhension**, une instruction Python permettant de construire un tableau `jours_chauds` contenant tous les enregistrements pour lesquels la température est strictement supérieure à 5řC. On n'oubliera pas d'effectuer la conversion de type nécessaire. *(1 pt)*

**2. Projection**

a) Écrire, en utilisant **une liste en compréhension**, une instruction Python permettant de construire le tableau `toutes_villes` contenant toutes les valeurs de l'attribut `ville` apparaissant dans la table. *(0,5 pt,5>1 s)*
b) Le tableau `toutes_villes` contient des doublons. Proposer une solution en Python pour obtenir un tableau `villes_uniques` ne contenant chaque nom de ville qu'une seule fois. *(1 pt)*

**3.** Écrire, en utilisant **une liste en compréhension**, une instruction Python permettant d'obtenir le tableau des couples `(ville, temperature)` de tous les enregistrements de la table, la température étant convertie en flottant. *(1 pt)*

## Exercice 4 – Tri d'une table *(3,5 pt)*

On travaille toujours avec le tableau de dictionnaires `table` de l'exercice précédent.

**1.** Rappeler le rôle des deux paramètres nommés `key` et `reverse` de la fonction `sorted`. *(1 pt)*

**2.** On souhaite trier la table par ordre croissant de température. Écrire la fonction `cle_temperature` à passer en argument `key`, puis l'instruction de tri. On veillera à effectuer la conversion de type adéquate. *(1 pt)*

**3.** On souhaite trier la table selon **deux critères** : d'abord par ville dans l'ordre alphabétique, puis à l'intérieur de chaque ville, par température décroissante. On procède en deux étapes.

a) Écrire la fonction `cle_ville` et effectuer le premier tri (par ville, ordre alphabétique). *(0,5 pt,5>1 s)*

b) Écrire la fonction `cle_temp` et effectuer le second tri (par température décroissante) sur le résultat du premier tri. *(1 pt)*

## Exercice 5 – Jointure et écriture CSV *(4 pts)*

On dispose d'une seconde table, stockée dans un tableau de dictionnaires nommé `infos_villes`, qui contient des informations sur les villes :

```
infos_villes = [
    {'ville': 'Paris', 'region': 'Île-de-France', 'population': 2165423},
    {'ville': 'Lyon', 'region': 'Auvergne-Rhône-Alpes', 'population':
       522250},
    {'ville': 'Marseille', 'region': 'PACA', 'population': 870731},
    {'ville': 'Toulouse', 'region': 'Occitanie', 'population': 493465}
]
```

**1.** Qu'appelle-t-on une **jointure** entre deux tables ? Quel est l'attribut commun qui permet de réaliser la jointure entre les tables `table` et `infos_villes` ? *(1 pt)*

**2.** Compléter la fonction suivante qui réalise la jointure entre deux tables selon un attribut commun. La table résultante contient, pour chaque couple d'enregistrements ayant la même valeur pour l'attribut commun, la fusion de tous les attributs. *(2 pts)*

```
from copy import deepcopy

def jointure(t1, t2, attribut):
    resultat = []
    for enreg1 in ______:
        for enreg2 in ______:
            if ______ == ______:
                new = deepcopy(______)
                for cle in ______:
                    if cle != ______:
                        new[______] = ______
                resultat.append(______)
    return resultat
```

**3.** Donner l'instruction permettant d'appeler cette fonction pour effectuer la jointure entre `table` et `infos_villes` selon l'attribut `'ville'`. On stockera le résultat dans une variable `table_complete`. *(0,5 pt,5>1 s)*

**4.** Écrire les instructions Python utilisant le module `csv` permettant d'enregistrer le contenu de `table_complete` dans un fichier `meteo_complete.csv` avec le délimiteur `';'`. On écrira d'abord la ligne des attributs puis les enregistrements. *(0,5 pt,5>1 s)*

---

*Fin du sujet*