

# LE TRAITEMENT DE DONNEES EN TABLES

## Partie 2 : Exercices d'application

### Exercice 1 : Exploitation d'un fichier csv.

On dispose de données dans un fichier CSV nommé « **detectionParticulesCSV** ». Chaque ligne du fichier comporte 7 valeurs séparées par des virgules. La première ligne contient les noms des champs, soit la durée de mesure depuis la date t=0, et le diamètre des particules ; 6 catégories possibles.

| A1 | :  | x | ✓ | f <sub>x</sub> | duree,0-diametre-0.3,0.3-diametre-0.5,0.5-diametre-0.7,0.7-diametre-1.0,1.0-diametre-5.0,diametre>10 |   |   |   |   |   |   |
|----|--|---|---|----------------|--|---|---|---|---|---|---|
|    | A  | B | C | D              | E  | F | G | H | I | J | K |
| 1  | duree,0-diametre-0.3,0.3-diametre-0.5,0.5-diametre-0.7,0.7-diametre-1.0,1.0-diametre-5.0,diametre>10 |   |   |                |  |   |   |   |   |   |   |
| 2  | 3600,498525,184262,81877,107093,728,3  |   |   |                |  |   |   |   |   |   |   |
| 3  | 7200,573451,203532,94872,106913,765,4  |   |   |                |  |   |   |   |   |   |   |
| 4  | 10800,551354,226528,106223,132096,883,6  |   |   |                |  |   |   |   |   |   |   |
| 5  | 14400,454884,157066,78521,95805,482,3  |   |   |                |  |   |   |   |   |   |   |
| 6  | 18000,544187,136212,66352,89886,452,3  |   |   |                |  |   |   |   |   |   |   |
| 7  | 21600,527189,163921,78530,101519,709,5   |   |   |                |  |   |   |   |   |   |   |
| 8  | 25200,576910,211717,109825,124123,899,8  |   |   |                |  |   |   |   |   |   |   |
| 9  | 28800,781030,223858,114841,126497,945,3  |   |   |                |  |   |   |   |   |   |   |
| 10 | 32400,833212,223793,103831,128840,579,3  |   |   |                |  |   |   |   |   |   |   |
| 11 | 36000,793722,194084,84447,102853,420,1   |   |   |                |  |   |   |   |   |   |   |
| 12 | 39600,739674,193861,82417,122906,595,2   |   |   |                |  |   |   |   |   |   |   |

Ecrire un programme qui permette de récupérer une ligne de données sur 10. (La première ligne de données ; la onzième,...)

L'ensemble des lignes obtenues doit être stocké dans une liste dont chaque élément est une liste constitué des valeurs d'une ligne.

Conseil : pour récupérer une ligne sur 10, vous pouvez utiliser «  $j \% 10 == 1$  »

### Exercice 2 : Pairs ou impairs ?

Ecrire une fonction qui crée une liste « **listeNombre** » de 100 nombre entiers compris entre 0 et 500.

Ecrire une fonction qui prend en argument une liste composée d'entiers et renvoie 2 listes, une première liste « **listePairs** » qui renvoie les nombres pairs classés par ordre croissant et une deuxième liste « **listeImpairs** » qui renvoie les nombres impairs classés par ordre décroissant.

### **Exercices 3 : Fusion de listes**

Ecrire une fonction « **fusion(liste1,liste2)** » qui prend en arguments 2 listes (liste1 et liste2) et qui renvoie une seule liste « **listeFusion** » qui contient tous les éléments des 2 listes rangés par ordre croissant.

### **Exercice 4 : Fonction Mystère**

Que renvoie cette fonction « mystère » ?

Répondre à la question, puis vérifier votre réponse en testant le code sur PyScripter.

```
liste1=[26, 36, 42, 44, 46, 50]
liste2=[5, 14, 22, 29, 38, 40, 47]

def mystere(liste1,liste2):
    liste=[]
    j, k = 0, 0
    while j<len(liste1) and k<len(liste2):
        if liste1[j] < liste2[k]:
            liste.append(liste1[j])
            j=j+1
        else:
            liste.append(liste2[k])
            k=k+1
    return liste
```

### **Exercice 5 : QCM**

On construit un tableau avec le code suivant :

```
tableau = [3*[0] for i in range(3)]
for j in range(3):
    tableau[j][j] = j + 1
    tableau[0][j] = tableau[0][j] + j + 1
    tableau[j][2] = tableau[j][2] + j + 1
print (tableau)
```

Quel est le résultat obtenu ?

- A : [[1, 2, 4], [0, 2, 2], [0, 0, 6]]
- B : [[2, 2, 4], [0, 2, 2], [0, 0, 4]]
- C : [[1, 2, 4], [0, 2, 4], [0, 0, 6]]
- D : [[2, 2, 4], [0, 2, 2], [0, 0, 6]]