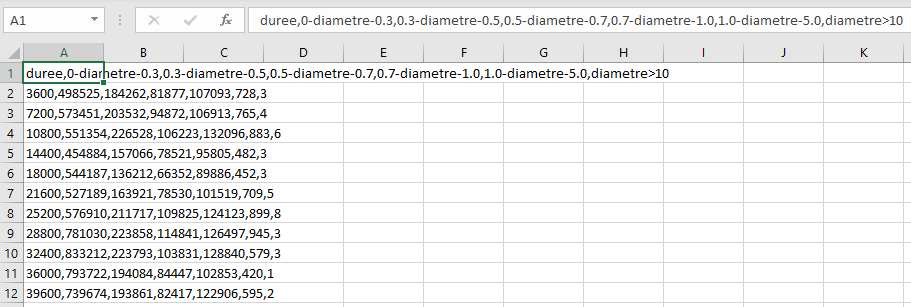
**LE TRAITEMENT DE DONNEES EN TABLES**

**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.



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.

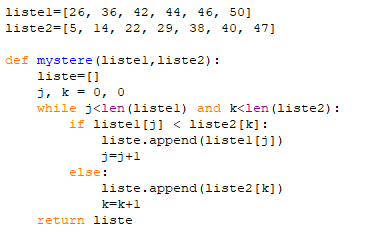
**Exercices 3 : Fusion de listes**

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**

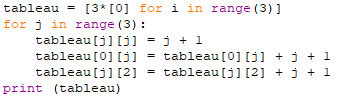
Que renvoie cette fonction « mystère » ?

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



**Exercice 5 : QCM**

On construit un tableau avec le code suivant :



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]]