

## TP 2: GESTION DES FICHIERS ET DES BASES DE DONNEES EN PYTHON

---

### Exercice 1

L'exercice consiste à créer une fonction **NbrLigne** qui a pour paramètre le *nom d'un fichier* (texte) et qui renvoie le nombre de lignes de ce fichier.

### Exercice 2

Dans une société multinationale, on reçoit plusieurs candidatures. Pour bien mener la campagne de recrutement, on souhaite gérer un fichier typé intitulé `concours.txt` qui comporte les enregistrements relatifs aux candidats d'un concours. Chaque enregistrement est composé de : NCIN, NOM, PRENOM, AGE, DECISION : (type contenant les identificateurs suivants : Ad (admis), R (refusé), Aj (ajourné), et séparés par point-virgule (;).

1. Définir la fonction `saisir()` qui permet de **remplir** les données relatives aux candidats dans le fichier `concours.txt` en contrôlant la saisie de la décision.
2. Définir la fonction `admis()` qui permet créer le fichier `admis.txt` comportant les données relatives aux candidat admis
3. Afin d'aider la prise de décision, la société souhaite sélectionner en priorité les candidats admis et âgés moins de 30 ans, créer la fonction `attente()` qui produira à partir du fichier `admis.txt`, un nouveau fichier intitulé `attente.txt` comportant les données relatives aux candidats admis et âgés plus que 30 ans. Une ligne du fichier `attente.txt` comprend le NCIN, le NOM et PRENOM d'un candidat, séparés par point-virgule (;).
4. Définir la fonction `statistiques(dec)` qui permet de calculer le pourcentage des candidats pour la décision `dec` (admis, refusé et ajourné). Exemple :  
Le pourcentage des candidats admis = (Nombre des candidats admis / Nombre des candidats) \* 100
5. Définir la fonction `supprimer()` qui supprimera du fichier `admis.txt` les candidats âgés plus que 30 ans.

N.B : On suppose que les fichiers seront mis à la racine du lecteur D.

## Exercice 3

1) Créer un fichier CSV "Elections.csv" contenant tous les candidats d'une campagne électorale, triés par ordre croissant sur le nom.

Chaque ligne a les colonnes suivantes :

- NOM : nom du candidat,
- PRENOM : prénom du candidat,
- AGE : âge du candidat,
- Nombre de voix : Nombre de voix obtenues.

2) Écrire le code python qui lit le fichier CSV et qui écrit :

- le nombre de candidats ayant plus que 10 000 voix,
- le nombre de candidats âgés de moins de 40 ans,
- la moyenne de voix obtenues,
- le gagnant dans cette campagne,
- le candidat ayant le moins de voix.

3) Écrire une fonction qui trie les résultats issus du fichier "elections.csv" suivant le nombre de voix par ordre décroissant et qui sauvegarde le résultat dans un nouveau fichier CSV "resultatsElections.csv".

## Exercice 4

Un opérateur téléphonique propose de créer un répertoire téléphonique contenant les noms des personnes et leurs numéros de téléphones.

Ce répertoire est représenté par un fichier texte nommé 'Contacts.txt'. Ce fichier contiendra des objets de type Contact. Ce dernier est caractérisé par un nom et le numéro de téléphone.

1. Écrire la procédure CreerFichier() qui permet de créer ce fichier en utilisant le package Pickle contenant N contacts.

2. Écrire la procédure ajouterContact(nom,telephone) qui ajoute les données d'une personne (nom, téléphone) sur un enregistrement à la fin du fichier Contacts.txt. Si le numéro de téléphone ne contient pas huit chiffres un message d'erreur sera affiché.

3. Écrire la fonction Recherche(nom) qui renvoi le numéro de la personne dont le nom est passé en paramètre. Si le nom de la personne n'existe pas dans le répertoire alors la fonction renvoi la constante None (rien).

4. Écrire la fonction modifierContact(nom,num) qui modifie le numéro du contact dont le nom est transmis en paramètre, cette fonction retourne True si la modification est effectuée correctement sinon False.

5. Écrire la procédure Copie() qui crée une copie du fichier Contacts.txt

6. Écrire la procédure affiche(nomFich) qui affiche à l'écran (objet par objet) les données de tous les contacts contenus dans le fichier nomFich.

7. (À rendre)

Dans un répertoire téléphonique un numéro de téléphone doit exister une seule fois, on suppose que par erreur nous avons répété quelques numéros sous des noms différents. Écrire la procédure suppDoublants(nomFich), qui affiche à l'écran les numéros en double avant de les supprimer.

**8. (À rendre)**

Écrire la procédure `Trier()` qui permet de trier les contacts du fichier « Contact.txt » selon l'ordre alphabétique des noms.