TP

Les fichiers

1 – Présentation du TP

Objet du TP

Ce TP consiste à manipuler des fichiers en Python.

Dans PyCharm, créez un projet appelé « tp10 ».

Récupérez le fichier « fichiers.zip » sur la page Moodle du cours et décompressez-le. Intégrez les fichiers correspondants dans votre projet.

Créez ensuite un fichier Python par exercice.

Compétences travaillées dans ce TP

- Opérations de base sur un fichier.
- Manipulation de données au format CSV

> ⚠ **Consigne générale :**
> Dans tous les exercices, ne pas oublier de fermer les fichiers !

2 – Exercice 1 : Nombre de lignes d'un fichier texte

Ecrivez une fonction :

```
def calculer_nb_lignes(nom_fichier) :
```

qui renvoie le nombre de lignes du fichier dont le nom est donné en paramètre.

Pour tester votre fonction, récupérez le fichier lorem_ipsum.txt sur la page Moodle du cours et placez-le dans le même répertoire que votre programme.

Exemple d'utilisation

Code	Affichage
`nb_lignes = calculer_nb_lignes("lorem_ipsum.txt")` `print(nb_lignes)`	4

> 🔍 **Lorem ipsum**
>
> Le lorem ipsum est un faux texte utilisé pour calibrer des mises en page. Il ressemble à du latin mais ne veut rien dire.

3 – Exercice 2 : Recopie d'un fichier

Ecrivez une fonction :

```
def recopier_fichier(nom_fichier_source, nom_fichier_destination):
```

qui lit les lignes du fichier nommé `nom_fichier_source` et les écrit à l'identique dans le fichier nommé `nom_fichier_destination`.

Exemple d'utilisation

Si on fait l'appel :

```
recopier_fichier("lorem_ipsum.txt", "lorem_ipsum_copie.txt")
```

alors le fichier `lorem_ipsum_copie.txt` contiendra une copie du fichier `lorem_ipsum.txt`.

4 – Exercice 3 : Inversion du contenu d'un fichier

Ecrivez une fonction :

```
def inverser_fichier(nom_fichier_source, nom_fichier_destination) :
```

qui lit les lignes du fichier nommé `nom_fichier_source` et les écrit dans l'ordre inverse dans le fichier nommé `nom_fichier_destination`.

Exemple d'utilisation

Supposons que le fichier `lorem_ipsum.txt` contienne les lignes suivantes :

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Quisque placerat ullamcorper blandit.
Integer eu dignissim sapien, commodo mattis nunc.
Nulla quam felis, scelerisque ut velit sit amet, aliquam sodales velit.
```

Alors l'appel suivant :

```
inverser_fichier("lorem_ipsum.txt", "lorem_ipsum_inv.txt")
```

doit produire un fichier `lorem_ipsum_inv.txt` dans le même répertoire dont le contenu sera :

```
Nulla quam felis, scelerisque ut velit sit amet, aliquam sodales velit.
Integer eu dignissim sapien, commodo mattis nunc.
Quisque placerat ullamcorper blandit.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

5 – Exercice 4 : Extraction des personnes majeures

On vous fournit un fichier `ages.csv` qui contient l'âge de personnes identifiées par leur nom et leur prénom. Ce fichier est au format CSV, décrit dans l'encadré ci-dessous.

Format CSV

CSV (pour *Comma-Separated Values*) est un format permettant de représenter des tableaux sous forme de texte. Les colonnes sont séparées par des virgules. Il existe des variantes dans lesquelles un autre séparateur que la virgule est utilisé pour délimiter les colonnes. En France, on utilise plutôt des points virgules.

Exemple :

Tableau

NOM	PRENOM	AGE
Marie	Dupond	12
Pierre	Durand	46

⟺

Format CSV

```
NOM;PRENOM;AGE
Marie;Dupond;12
Pierre;Durand;46
```

La première ligne, qui contient le nom des colonnes, est appelée *entête*.

Si vous ouvrez le fichier `ages.csv` dans un tableur comme Excel par exemple, vous devriez le voir s'afficher comme ceci :

	A	B	C
1	NOM	PRENOM	AGE
2	Marie	Dupond	12
3	Pierre	Durand	46
4	Nathalie	Martin	35
5	Jean	Thomas	17
6	Isabelle	Dubois	9
7	Nicolas	Petit	70

Ecrivez une fonction :

```
def extraire_personnes_majeures(nom_fichier_source, nom_fichier_destination):
```

qui lit les données du fichier nommé `nom_fichier_source`, ne conserve que les lignes correspondant à des personnes majeures, et les écrit dans le fichier nommé `nom_fichier_destination`.

L'entête du tableau initial devra être conservé dans le tableau final.

Aide :

- N'hésitez pas à faire des fonctions utilitaires, par exemple pour découper les lignes
- Les fonctions suivantes peuvent vous être utiles :

`s.split(sep)`	Découpe la chaîne `s` à l'aide du séparateur `sep`, et renvoie les morceaux sous la forme d'une liste de chaînes de caractères.
`sep.join(l)`	Construit une chaîne de caractères contenant tous les éléments de la liste `l`, reliés par le séparateur `sep`.

Exemple d'utilisation

Supposons que le fichier `ages.csv` contienne les lignes suivantes :

```
NOM;PRENOM;AGE
Marie;Dupond;12
Pierre;Durand;46
Nathalie;Martin;35
Jean;Thomas;17
Isabelle;Dubois;9
Nicolas;Petit;70
```

Alors l'appel suivant :

```
extraire_personnes_majeures("ages.csv", "ages_majeurs.csv")
```

doit produire un fichier `ages_majeurs.csv` dans le même répertoire dont le contenu sera :

```
NOM;PRENOM;AGE
Pierre;Durand;46
Nathalie;Martin;35
Nicolas;Petit;70
```

6 – Exercice 5 : Extraction des notes d'une matière

On vous fournit un fichier `notes.csv` qui contient les notes d'étudiants anonymisés dans différentes matières, dont un aperçu dans un tableur est donné ci-dessous :

	A	B	C	D
1	ID ETUDIANT	MATHS	PHYSIQUE	INFO
2	984061	19	11	12
3	570962	2	15	19
4	957416	17	13	9
5	234156	18	0	6
6	521084	11	15	2
7	671350	16	3	17

Ecrivez une fonction :

```
def extraire_notes_matiere(nom_fichier_source, nom_fichier_destination,
nom_matiere):
```

qui lit les données du fichier nommé `nom_fichier_source`, ne conserve que les notes de la matière nommée `nom_matiere`, et les écrit dans le fichier nommé `nom_fichier_destination`.

L'entête du tableau initial devra être conservé dans le tableau final, ainsi que l'identifiant des étudiants.

Aide :

Les fonctions suivantes peuvent vous être utiles :

`s.strip()`	Retire tous « blancs » (espaces, tabulations, sauts de ligne, …) au début et à la fin de la chaîne `s`.
`l.index(v)`	Renvoie l'indice de la valeur `v` dans la liste `l`. Produit une erreur si la valeur n'est pas trouvée.

Exemple d'utilisation

Supposons que le fichier `notes.csv` contienne les lignes suivantes :

```
ID ETUDIANT;MATHS;PHYSIQUE;INFO
984061;19;11;12
570962;2;15;19
957416;17;13;9
234156;18;0;6
521084;11;15;2
671350;16;3;17
```

Alors l'appel suivant :

```
extraire_notes_matiere("notes.csv", "notes_matiere.csv", "PHYSIQUE")
```

doit produire un fichier `notes_matiere.csv` dans le même répertoire dont le contenu sera :

```
ID ETUDIANT;PHYSIQUE
984061;11
570962;15
957416;13
234156;0
521084;15
671350;3
```

Amélioration

En utilisant des opérations usuelles sur les chaînes, faites en sorte que le nom de la matière soit insensible à la casse (minuscules / majuscules).