

TP : «Utilisation de fichiers»**Objectifs du TP**

- Manipuler des chaînes de caractères
- savoir lire, modifier ou écrire sur des fichiers de texte à l'aide de Python. On pourra regarder le document de cours sur les chaînes de caractères et fichiers.

Dans tout le TP, les fichiers à utiliser le seront à l'emplacement suivant :

W:/MPSI/informatique/TP/matos/

Si vous utilisez les ordinateurs du lycée, vous pouvez sauver les fichiers que vous créez dans votre lecteur personnel, Z:/.

Exercice 1 Copier le répertoire `matos` du lecteur commun dans la racine de votre lecteur personnel. **Vous travaillerez désormais avec les fichiers du répertoire `matos` qui sont sur votre lecteur personnel.** Pourquoi ne travaille-on pas directement sur les fichiers du lecteur commun ?

Exercice 2 Créer un fichier texte '`facto.txt`' de 20 lignes dont la i -ligne est la valeur de $i!$.

```
1
1
2
6
24
120
720
5040
40320
362880
3628800
39916800
479001600
6227020800
87178291200
1307674368000
20922789888000
355687428096000
6402373705728000
121645100408832000
```

Exercice 3 (Lecture d'un fichier)

1. Exécuter le script suivant ?

```
f = open("Z:/matos/fichier1.txt",'r')
for L in f:
    print(L)
    print(len(L))
f.close()
```

2. Vérifier en ouvrant le fichier `fichier1.txt` à l'aide d'un éditeur de texte (par exemple le bloc-notes). Que-se-passe-t-il ?
3. Écrire une fonction `nbreCaracteres` qui calcule le nombre de caractères d'un fichier.
4. Ouvrir avec un éditeur de textes le fichier `exos.tex`, parcourir son contenu, puis récupérer sa taille en octets. Estimer ainsi son nombre de caractères. Vérifier votre estimation avec votre fonction `nbreCaracteres`.
5. Faire afficher la dixième ligne en utilisant la méthode `readline`.
6. Écrire une fonction `nbreVoyelles` qui donne le nombre de voyelles d'une chaîne de caractères.
7. En déduire, une fonction qui donne le nombre de voyelles d'une ligne d'un fichier. Tester avec la ligne 10 du fichier `exos.tex` (on doit trouver 6).
8. Créer un dictionnaire qui stocke les différentes occurrences de chaque caractère.

Exercice 4 (Liste des nombres premiers) Créer un fichier nommé `premiers.txt` qui contient l'ensemble des nombres premiers inférieurs à 100000. On séparera les nombres premiers par une virgule.

On pourra utiliser librement la fonction `est_premier` du fichier `test_primalite.py` dans le matos.

Vérifier en ouvrant votre fichier.

Exercice 5 (Un peu de littérature) Georges Sand (nom d'apparat, c'est-en-réalité une femme) a adressé une lettre à Alfred de Musset. Celui-ci lui a répondu. Les textes sont dans des fichiers qui se trouvent dans le répertoire Sand de l'emplacement du TP.

1. Afficher une ligne sur deux du fichier `lettre_sand.txt`.
2. Lire la réponse de Mr Musset contenue dans le fichier `reponse_musset.txt`. Deviner le code.
3. Écrire un script Python qui réalise ce décodage.
4. Décoder enfin la réponse de Sand.

Exercice 6 Résoudre le problème 81 du site EulerProblem.

Exercice 7 (Admissibles) Il s'agit du vrai listing des candidats du concours Mines-Ponts 2013, filière MP, récupéré sur Internet. Il y a 5398 lignes. Chaque ligne est constituée de quatre champs : numéro de candidat, Nom et Prénom, Admissible ou refusé, numéro de série d'oral si admissible.

1. Ouvrir avec un éditeur de textes le fichier `admissibles.txt`. Quel semble être le caractère qui sépare les quatre champs ?
2. Compléter la première ligne du script suivant pour ouvrir avec Python en mode lecture, votre fichier `admissibles.txt` du répertoire `matos`.

```
f = open( ....  
a= f.readline() # la première ligne du fichier  
b= f.readline() # la deuxième ligne du fichier  
f.close()
```

Exécuter ce script, puis à l'intérieur de la console taper `print(a)` puis `a`. Quel est donc le caractère qui sépare les quatre colonnes ?

3. Déterminer le nombre d'admissibles.
4. Écrire un fichier contenant tous les prénoms des admissibles.