
TD 7 (programmation) - Fichiers.

Exercice 1 (Mélange)

Écrivez une fonction qui prend trois noms de fichiers en paramètres, et qui écrit dans le troisième fichier le mélange des deux premiers, c'est-à-dire la ligne 1 du premier fichier, puis la ligne 1 du deuxième fichier, puis la ligne 2 du premier fichier, ... Attention, les fichiers d'entrée peuvent ne pas posséder le même nombre de lignes !

Exercice 2 (Unicité)

Soit `dico1.txt` et `dico2.txt` deux fichiers contenant des mots de la langue française. Les deux dictionnaires étant incomplets, on veut les fusionner en un seul fichier `fusion.txt` pour combiner leurs informations mais en éliminant les doublons. On suppose que chaque dictionnaire contient exactement un mot par ligne, et que chaque ligne se termine par `'\n'`. Écrivez un programme qui réalise cette tâche. Vous pouvez utiliser `sorted(iterable)`, qui renvoie la liste des éléments d'`iterable` triée de manière croissante.

Exercice 3 (Effaçage de commentaires)

Écrivez un programme qui affiche le contenu d'un fichier source Python sans les commentaires (on les remplace par des retours à la ligne). En cas de besoin, vous pouvez utiliser le paramètre `end` de la fonction `print` pour spécifier par quoi la ligne affichée doit se terminer (par défaut, `end == '\n'`). On suppose que les `#` n'apparaissent pas en dehors des commentaires.

Exercice 4 (grep)

Écrivez un programme qui prend en paramètres un fichier et un mot, et qui affiche toutes les lignes du fichier contenant ce mot avec leur numéro. Par exemple, si le texte du fichier est `"ce fichier ne contient\npas d'informations\nintéressantes"` et que la chaîne à chercher est `"nt"`, le programme affichera :

```
ligne 1: ce fichier ne contient
ligne 3: intéressantes
```

lorsqu'on exécutera `python3 programme.py fichier.txt nt`

Exercice 5

Écrivez un programme qui prend en paramètre le nom d'un fichier texte et qui affiche le mot le plus fréquent de ce fichier avec sa fréquence. On suppose qu'il n'y a pas de ponctuation dans le fichier.

Exercice 6 (Correction d'oublis)

Écrivez un prototype de correcteur orthographique, qui demande à l'utilisateur d'entrer un texte et qui suggère des corrections pour tous les mots mal orthographiés. Pour simplifier le problème, on suppose que :

- aucun caractère n'est accentué ou spécial ;
- les seules erreurs qu'un utilisateur commet sont l'oubli d'une lettre dans un mot (on devra donc suggérer la correction `"lettre"` pour `"letre"`, mais on sera incapable de la suggérer pour `"ttre"`) ;
- une liste des mots du français est fournie dans un fichier texte, contenant exactement un mot par ligne ;
- une fonction `voisins(mot)` existe et nous renvoie l'ensemble des mots qu'on peut obtenir en effaçant un seul caractère du mot fourni.

Attention : il peut y avoir plusieurs corrections à suggérer pour un même mot (par exemple : `"rre"` peut correspondre à `"rare"` ou à `"rire"`), et on les veut **toutes**.