

## Exercice 1

Écrivez une fonction `compter(texte1, texte2)` qui prend en paramètre deux textes et qui retourne le nombre de fois que le deuxième texte (qui doit être non vide) apparaît dans le premier texte sans chevauchement. Écrivez aussi des tests unitaires.

## Exercice 2

Écrivez une fonction `enCommun(texte1, texte2)` qui prend en paramètre deux textes et qui retourne `True` si les deux textes ont un caractère en commun (c'est-à-dire qu'il existe un caractère dans `texte1` qui apparaît aussi dans `texte2`). Écrivez aussi des tests unitaires.

## Exercice 3

Expérimentez avec l'utilisation des commandes `shell` suivantes avec des chemins d'accès absolus et relatifs :

```
% pwd
% cd
% ls
% cp fichier1 fichier2
% mv fichier1 fichier2
% rm fichier
% mkdir directory
% rmdir directory
% cat fichier
% less fichier
```

## Exercice 4

Installez CPython 3 (<https://www.python.org/downloads/>) et utilisez le pour créer et exécuter un programme qui copie un fichier `a.txt` à `b.txt` en remplaçant toutes les lettres minuscules par des lettres majuscules. Utilisez les fonctions `readFile` et `writeFile` montrées en cours.

## Exercice 5

Écrivez un programme qui fait comme l'exercice précédent, mais qui préserve la casse des lettres qui sont entre apostrophes. Donc « `Il a dit 'Bonjour'` » doit être converti en « `IL A DIT 'Bonjour'` ».

## Exercice 6

Écrivez un programme qui copie le fichier `a.txt` à `b.txt` et qui insère des blancs au début de chaque ligne pour que chaque ligne plus courte que 79 caractères dans `a.txt` ait 79 caractères dans `b.txt`.

## Exercice 7

Un fichier contient des phrases formées de mots (séquence de a-z/A-Z) et de ponctuations (point et virgule), et ayant des sauts de ligne à n'importe quel endroit au milieu du texte, possiblement au milieu d'une phrase. Codez une fonction `justifier(path)` qui prends un tel fichier et produit un autre fichier où chaque phrase (se terminant par un point) se trouve sur sa propre ligne du fichier. Donc sur chaque ligne il y aura exactement une phrase. Les espaces entre les phrases doivent être préservés.

## Exercice 8

Écrivez la fonction `sommeFichier(path)` qui retourne la somme de tous les nombres dans le fichier ayant le chemin d'accès `path`. Le fichier contient uniquement des nombres entiers, des espaces et des fins de ligne.

## Exercice 9

Le fichier `bottin.csv` est un fichier `csv` qui contient un bottin téléphonique associant à chaque nom un numéro de téléphone. Voici un exemple de fichier :

```
jean,450 831-5133  
luc,514 824-9913  
marie,819 123-4567
```

Écrivez un programme interactif qui va permettre de consulter le bottin, soit en cherchant un nom (pour trouver le numéro de téléphone de la personne) ou un numéro de téléphone (pour trouver le nom de la personne), ou bien pour ajouter des personnes au bottin. Le programme doit utiliser `input` pour demander la commande à l'utilisateur et pour afficher le résultat. La commande `fin` ou une ligne vide doit terminer le programme. Lorsque le programme termine le nouveau fichier `bottin.csv` doit être sauvé.