

Exercice 1

Ecrire un programme `compo_texte.py` qui lit deux fichiers :

- `perec.txt` contenant un texte issu de "La disparition" de Georges Perec
- `zola.txt` contenant un texte issu de "L'assommoir" d'Emile Zola

puis utilise la fonction `calc_composition(chaine)` (voir ci-dessous) pour calculer la fréquence de chaque lettre dans les deux textes. Au final, le programme affiche à l'écran la sortie suivante en colonne comparant la fréquence des lettres dans les deux textes :

```
Perec      Zola
a   x.xxx   y.yyy
b   .....   .....
etc
```

Dans le programme vous écrirez une fonction `calc_composition(chaine)` où `chaine` est une chaîne de caractères contenant un texte sans lettre accentuée. La fonction, convertira le texte en minuscule puis ôtera les espaces et les signes de ponctuation (. , ; ' ! ? - etc) ainsi que les retours à la ligne, pour ne garder au final que les caractères de l'alphabet. La fonction doit **renvoyer un dictionnaire** contenant les fréquences des 26 lettres de l'alphabet (même pour celles qui n'existent pas dans `chaine`). Attention, les fréquences des majuscules et minuscules doivent s'accumuler (compter pour la même lettre).

Que constatez-vous concernant la fréquence de la lettre e dans les deux textes? Vous pouvez répondre à cette question en mettant un commentaire à la fin de votre script python.

NB : les accents ou autres cédilles ont été retirés de ces textes afin de pouvoir réaliser l'exercice.

Exercice 2

On fournit `seq_sacch.fasta` contenant un fragment du brin d'ADN sens du chromosome I de la souche *saccharomyces cerevisiae* S288c. On fournit également `code.py` contenant un dictionnaire avec le code génétique (brin d'ADN direct → protéine). On sait que cette séquence d'ADN contient un gène codant pour une protéine dans le cadre de lecture direct (le premier codon correspond aux 3 premières bases dans le fichier `seq_sacch.fasta`).

Ecrire le programme `translate.py` qui recherche ce gène et le traduit en protéine (en important `code.py` comme un module). Le programme doit écrire dans le fichier de sortie `translate.out` les éléments suivants :

- la séquence de cette protéine
- la longueur de cette protéine
- la composition en acide-aminé de cette protéine

Vous êtes libre de formater ces informations dans le fichier `translate.out` comme bon vous semble.

Exercice 3

Le fichier `1gzz.pdb` contient la structure tridimensionnelle de l'hémoglobine résolue par cristallographie de rayons X (fichier fourni). Cette protéine contient 4 sous-unités, symbolisées par des chaînes différentes

dans le fichier PDB, précisément A, B, C et D. On souhaite écrire un programme `dist_subunit.py` qui calcule la distance entre les centres de masse de toutes les paires de sous-unités et écrit ces distances dans le fichier `dist_subunit.out`, par exemple :

```
distance chaine A - chaine B = xx.xx Ang
distance chaine A - chaine C = xx.xx Ang
[...]
```

Ce programme devra contenir plusieurs fonctions :

- `get_coor(nom_fichier, chaine)` où `nom_fichier` est le nom du fichier PDB et `chaine` est le nom de la chaîne (A, B, C et D). Cette fonction doit renvoyer les coordonnées des C_α sous forme d'une liste de dictionnaires. Chaque dictionnaire contient les coordonnées x, y, z d'un C_α et son numéro de résidu (comme vu en TP).
- `get_COM(liste_coor)` où `liste_coor` est la liste de dictionnaires renvoyée par la fonction précédente. Cette fonction `get_COM(liste_coor)` doit renvoyer les coordonnées x, y, z (sous forme de liste) du centre de masse des coordonnées contenues dans `liste_coor`.

Autre pièce fournie : `pdb_format.pdf` permettant de parser efficacement un fichier PDB.

Exercice 4

Soit les deux fonctions mathématiques f et g définies comme suit :

$$f(x) = \frac{1}{2} \sin\left(\frac{3}{2}x - \pi\right)$$

$$g(x) = e^{\frac{1}{1+x^2}} - \frac{\pi}{2}$$

On souhaite développer un programme python `search_root.py` qui évalue numériquement les racines de f et g sur l'intervalle $[-\pi, \pi]$ avec une précision δ de 0.01. On rappelle que les racines de f correspondent aux valeurs de x pour lesquelles $f(x) = 0$. Attention, on ne cherche pas une valeur analytique de ces racines, mais une valeur approchée calculée numériquement, c'est-à-dire que votre programme doit déterminer les bornes inférieure et supérieure tel que :

$$r_{\text{inf}} < r < r_{\text{sup}}$$

où r_{inf} est la borne inférieure, r_{sup} la borne supérieure et r une racine de la fonction que votre programme devra déterminer. On fera en sorte que $r_{\text{sup}} - r_{\text{inf}} = \delta$.

Votre programme affichera à l'écran toutes les racines de f et g comprises dans l'intervalle demandé. Une sortie possible pourrait être (les nombres ci-dessous ne sont pas une solution du problème) :

```
Une racine de f est comprise entre -0.14 et -0.15
```

```
...
```

```
Une racine de g est comprise entre ...
```

Conseils :

- Faire deux fonctions qui prennent x en argument et qui renvoie respectivement la valeur de $f(x)$ et de $g(x)$.
- Penser à utiliser une boucle qui balaye l'intervalle demandé sous forme de *floats*.
- Bien noter que dans la sortie les valeurs sont arrondies au nombre de décimales voulues dans la précision δ .