

Recherche d'une séquence d'ADN

1. Écrire une fonction `SequenceHasard(n)` qui renvoie une chaîne de caractère de longueur n dont les caractères sont "A", "C", "G" ou "T"
2. Écrire une fonction `SequenceEgale(seq1,seq2)` qui vérifie si les séquences `seq1` et `seq2` sont égales : pour cela on vérifiera d'abord si elles ont la même longueur, puis on testera si le i -ième caractère de `seq1` est bien égale au i -ième caractère de `seq2`.
3. Considérons une grande séquence comme `Seq="ACGAAGTACACA"` et une petite séquence comme `motif="AAGT"`, on souhaite écrire une fonction `Presence(motif,Seq,i)` qui renvoie `True` si `motif` se trouve dans `Seq` en partant du i -ième caractère et `False` sinon. Dans l'exemple, `Presence(motif,Seq,3)` doit renvoyer `True`, en effet, `Seq[3]=motif[0]`, `Seq[4]=motif[1]`, `Seq[5]=motif[2]` et `Seq[6]=motif[3]`. En revanche, elle renverra `False` pour valeur de $i \neq 3$. Pour cela, on ne fera que des comparaisons caractère à caractère. Pour tous les i possibles, on vérifiera si `motif` se trouve en position i , c'est-à-dire si `Seq[i]=motif[0]`, `Seq[i+1]=motif[1]` etc.
4. En déduire une fonction `Presence(motif,Seq)` qui renvoie `True` si `motif` est dans `Seq` (peu importe l'emplacement).
5. Créer `motif` une courte séquence de longueur 3 au hasard, ainsi que `Seq` une séquence de longueur 60 au hasard, en répétant un «grand» nombre de fois (disons 1000) cette expérience, donner une approximation de la probabilité qu'un `motif` au hasard se trouve dans une `Seq` au hasard.

Étude des mots de la langue française

1. Écrire une fonction `CompterVoyelles(mot)` qui, étant donné `mot` renvoie le nombre de voyelles¹ dans ce mot, par exemple, `biologie` contient 5 voyelles.
2. Téléchargez le fichier `TP7_Liste_Mots_Francais` qui se trouve sur CDP et mettez le dans le même dossier que votre fichier actuel (qui s'appelle forcément `TP7.py`)
3. Les commandes suivantes permettent d'ouvrir ce fichier et de créer une liste où chaque élément un mot de la liste :

1. On va considérer que y est toujours une voyelle.

```
with open("TP7_Liste_Mots_Francais.txt", "r") as Fichier:  
    Liste = Fichier.read().splitlines()
```

4. Afficher le nombre de mots.
5. Déterminer le nombre maximal de lettres dans un seul mot.
6. Afficher l'histogramme du nombre de lettres par mot.
7. Déterminer le nombre moyen de lettres dans un mot ainsi que l'écart-type (on n'utilisera ni `sum`, ni `mean` ou `std`)
8. Déterminer le nombre moyen de voyelles par mot ainsi que son écart-type
9. Déterminer la covariance du nombre de voyelles et du nombre de mot de la liste.
10. En déduire le coefficient de corrélation, que peut-on en conclure ? Est-ce logique ?
11. Écrire une fonction `NbMotsParLettre()` qui renvoie une liste `L` tel que `L[i]` soit égale au nombre de mots ayant i lettres.
12. Quel est le nombre médian de lettres par mot ?
13. Écrire une fonction `Verlan(mot)` qui à un mot le renvoie à l'envers, par exemple `Verlan("info")` renverra `"ofni"`.
14. En déduire une fonction `Palindrome(mot)` qui renvoie `True` si `mot` est un palindrome (comme radar) et `False`.
15. En déduire la liste des palindromes de langue française.

Apprends à parler comme Victor Hugo !

1. Importer la bibliothèque `re` grâce à la commande `import re`
2. Téléchargez le fichier `TP7_Les_Miserables` qui se trouve sur CDP et mettez le dans le même dossier que votre fichier actuel.
3. Recopiez les commandes suivantes :

```
with open("TP7_Les_Miserables.txt", "r") as Fichier:  
    Chaine = Fichier.read()
```

```
Liste = re.split(";|,|\n| |\?|\!|\-|\-|'",Chaine)  
Liste = [k.lower() for k in Liste if len(k)>0]
```

Ces commandes, ont transformé le fichier en une liste `Liste` où chaque élément de la liste est un des mots du roman en question.

4. Écrire une fonction `DictionnaireSuivant(mot)` qui à un mot (présent dans le roman) renvoie le dictionnaire dont les clés sont les mots qui suivent `mot` et dont la valeur associée à `motbis` est le nombre de fois où `motbis` est le mot suivant de `mot`. Par exemple, si `mot` est le mot `"rue"` alors `"rambuteau"` sera une clé de ce dictionnaire et la valeur vaut trois, car il y aurait trois fois, dans le roman des Misérables l'expression `"rue rambuteau"`.
5. Écrire une fonction `MaxMotSuivant(mot)` qui renvoie la valeur maximal du dictionnaire créée par la fonction écrite lors de la question précédente.
6. Écrire une fonction `ListeMotSuivantMax(mot)` qui renvoie la liste de tous les mots dont la valeur dans le dictionnaire est maximal.
7. Écrire une fonction `Completer(mot,n)` qui va créer une phrase comportant n mots partant du mot `mot`, le mot suivant sera l'un des mots qui est le plus susceptible de succéder à `mot` dans les misérables, puis on continue à partir de ce nouveau mot.
8. Comme on se rend compte qu'on tombe malheureusement dans une boucle, modifier la fonction précédente pour rajouter un peu d'aléa (après avoir compris ce qui causait ce phénomène).