I Commandes de manipulation des chaînes

1.1 Chaînes de caractères

Une chaîne de caractère est une liste de lettres (une lettre est aussi appelée caractère). On les crée par

```
s = ''ceci est une chaine''
s2 = 'ou avec des simples guillemets'  # attention aux apostrophes !
s +  s2  # concatène les deux chaînes.
s[1]  # accède au deuxième caractère.
```

1.2 Slice

On peut sélectionner toute une partie d'une chaîne ou d'une liste avec la syntaxe suivante

```
s[1:4]  # retourne la chaîne  dont les caractères sont les
# caractères de s d'indices range(1, 4)
s[:6]  # commence la sélection au début
s[3:]  # sélectionne jusqu'à la fin
```

II Algorithmes autour des chaînes

Créer un fichier "chaines.py" dan un dossier TD3 sur le réseau.

2.1 A l'envers, à l'endroit

2.1.1 Renverser

Ecrire une fonction qui renverse la chaîne passé en paramètre.
Donnée : une chaîne de caractère s.
Sortie : s renversée.
Exemple : renverse('chaine') retourne 'eniahc'

2.1.2 Palindrome

Une chaîne est un palindrome ssi on peut la lire de gauche à droite ou de droite à gauche de la même façon. Par exemple, "radar" est un palindrome. Créer une fonction qui teste si une chaîne donnée est un palindrome.
Donnée : une chaîne de caractère s.
Sortie : True ou False suivant que s est un palindrome ou non.
On attend une version utilisant la fonction précédente, et une ne faisant que $\frac{n}{2}$ comparaisons sans construire de chaînes supplémentaire (n est la longueur de s).

2.2 Motifs

2.2.1 Plus long préfixe commun

Créer une fonction qui prend deux chaînes s1 et s2 et qui retourne le plus grand préfixe commun à ces deux chaînes, c'est à dire la plus longue chaîne de caractère qui soit égale à la fois au début de s1 et au début de s2.
Par exemple, le plus grand préfixe commun de "abbaaab" et "abbbaab" est "abb"
Données : deux chaînes $s1$ et $s2$.
Sortie : la chaîne de caractère qui est le plus long préfixe commun à $s1$ et $s2$, éventuellement la chaîne vide.

2.2.2 Occurrences

Créer une fonction qui renvoie le nombre d'occurrences d'un motif dans une chaîne. Par exemple le motif "aba" apparaît 2 fois dans "ababa".
Donnée : une chaîne de caractère s et un motif mot.
Sortie : le nombre de fois où mot apparaît dans s.
Indication : ici, on pourra avantageusement créer une fonction intermédiaire `motif_a_partir(s, mot, indice)` qui teste si les caractères de s à partir de l'indice donné ont pour préfixe mot.

2.2.3 Une suite étrange

Vous connaissez peut-être la suite :

```
1
11
21
1211
111221
```

appelée suite de Conway. Pour comprendre son fonctionnement, lire à haute voix chaque chiffre d'un terme en regardant le suivant...
Donnée : un entier naturel $n > 0$.
Sortie : le nième terme de la suite, sous forme de chaîne de caractères.
On veillera à ne pas prendre n trop grand dans les tests.

2.3 Séparation

2.3.1 Nombre de mots

Créer une fonction qui compte le nombre de mots dans une chaîne. Deux mots sont séparés par un ou plusieurs espaces.
Donnée : une chaîne de caractère s.
Sortie : le nombre de mots de s.
Tester votre fonction avec plusieurs cas "pénibles" : plusieurs espaces entre les mots, espace pour commencer ou finir la chaîne.

2.3.2 Parenthèses

Créer une fonction qui vérifie si la chaîne passée en argument est bien parenthésée.
Donnée : une chaîne de caractère s.
Sortie : un booléen.
Pour découvrir les règles, on pourra s'intéresser aux exemples suivants :

- (a + b) - c)
- (a + (b - c)
- a +) b ((- c)