



## 7 TD : Les Chaînes de caractères

### 7.1 Manipulation des chaînes de caractères

1. Écrire une fonction qui prend en entrée une chaîne de caractère et affiche ses caractères séparément.
2. Écrire une fonction `supprime_Espace` qui supprime les espaces et les tabulations d'une chaîne de caractères. Cette fonction ne doit pas utiliser de tableau intermédiaire. **Exemple** : "ceci est une phrase !" devient "ceciestunephrase !"
3. Écrire une fonction permettant de supprimer toutes les occurrences d'un caractère dans une chaîne de caractère (les deux passés en paramètres).
4. Écrire une fonction permettant d'inverser une chaîne de caractère. Par exemple, la chaîne : 'looc' devient 'cool'
5. Écrire une fonction qui affiche le nombre de lettres, de chiffres et de caractères spéciaux dans une chaîne de caractères passée comme paramètre.
6. Écrire un programme pour tester toutes les fonctions précédentes.

### 7.2 Palindrome

Un palindrome strict est une séquence de caractères symétrique. Par exemple il peut s'agir d'un mot qui peut-être lu indifféremment dans un sens ou dans l'autre. Par exemple : radar, elle, kayak, *etc.*

1. Écrire une fonction `est_Palindrome_Strict` qui permet de tester si une chaîne de caractères est un palindrome strict, elle renvoie 1 si c'est le cas et 0 sinon.

D'autres chaînes peuvent également être qualifiées de palindromes non strict : ce sont les phrases dont la séquence des lettres formant les mots forment un palindrome strict lorsque l'on supprime les signes de ponctuation.

**Par exemple :**

— "Engage le jeu que je le gagne"

— "Eh! ca va la vache?"

2. Écrire une fonction `est_Palindrome` prenant pour paramètre une chaîne de caractères et renvoyant 1 s'il s'agit d'un palindrome non strict et 0 sinon.

**Remarque.**

— On ne considère ici aucune lettre accentuée.

— Pensez à réutiliser les fonctions définies dans l'exercice précédent.

### 7.3 Chaînes de caractères vs Nombres

1. Écrire la fonction `stringToInt` qui retourne la valeur numérique d'une chaîne de caractères représentant un entier (positif ou négatif) du type long. Si la chaîne ne représente pas une valeur entière correcte, la fonction arrête la conversion et fournit la valeur qu'elle a su reconnaître jusqu'à ce point.
2. Écrire la fonction `stringToDouble` qui retourne la valeur numérique d'une chaîne de caractères représentant un réel en notation décimale. Si la chaîne ne représente pas une valeur décimale correcte, la fonction arrête la conversion et fournit la valeur qu'elle a su reconnaître jusqu'à ce point.
3. Écrire la fonction `intToString` qui construit une chaîne de caractères S qui représente un nombre entier N du type long.
4. Écrire la fonction `doubleToString` qui construit une chaîne de caractères S qui représente un nombre réel N avec 4 positions derrière la virgule.
5. Écrire un programme principal pour tester les fonctions ci-dessus.

.....

### 7.4 Bio-informatique (Supplémentaire)

En biologie, un problème courant consiste à comprendre la structure des molécules d'ADN, et le rôle de structures spécifiques dans le fonctionnement de la molécule. Une séquence ADN est représentée par une suite  $c_0c_1c_2\dots$  de caractères choisis parmi les quatre nucléotides : adenine (A), cytosine (C), guanine (G) et thymine (T). Par exemple, la chaîne de caractères 'AAA-CAACTTCGTAAGTATA' représente un brin d'ADN.

1. Écrire une fonction `valide` qui prend une chaîne en paramètre, et qui renvoie 1 si la chaîne contient une chaîne ADN, et 0 sinon.
2. Écrire une fonction `cherche_base` qui retourne le rang où apparaît pour la première fois la base représentée par un caractère c dans la séquence ADN S, tous deux passés en paramètres. Si la base n'apparaît pas, la fonction devra renvoyer -1 pour indiquer une erreur.  
Exemples : `cherche_base("ATTGCC", 'C')`; retourne 4; et `cherche_base("GTTGCC", 'A')`; retourne -1.
3. Écrire une fonction `proportion` permettant de calculer les proportions de présence des bases A, C, G et T dans une séquence ADN.
4. Écrire une fonction `complement` qui retourne le complémentaire d'une séquence ADN : A devient T, T devient A, C devient G, G devient C.
5. Écrire une fonction `sous-chaîne` qui prend deux séquences ADN en paramètres et retourne true si la première est une sous séquence de la deuxième.
6. Écrire une fonction `laPlusLongueSousChaîne` permettant de retourner la plus longue sous-séquence commune à deux séquences ADN passées en paramètres.