Algorithmique et programmation
« Les chaînes de caractères »

Exercice 1 :
Ecrire une fonction «**apparitions(s , c)** » qui renvoie le nombre d'apparition d'une chaîne « **c** » dans une chaîne de caractères « **s** » sans utiliser la méthode « count ».

Exercice 2 :
Ecrire une fonction « **nbremaj(s)** »qui renvoie le nombre des caractères majuscules de la chaîne de caractères « **s** ».

Exercice 3 :
Ecrire une fonction « **somme(n)** »qui renvoie la somme des chiffres d'un entier passé en paramètre.

Exercice 4 :
Ecrire une fonction « **mots(s)** »qui renvoie le nombre de mots contenus dans une chaîne de caractères « **s** ». Les mots peut être séparé par : [" " , "," , ";" , " :" , " ." , " !" , " ?"]

Exercice 5 :
Écrire une fonction **premierMot(chaine)** qui renvoie le premier mot d'une chaîne de caractère. Par exemple si ma chaîne est « Samedi, je rentre chez mois», on renverra «Samedi».

Exercice 6 :
Un palindrome est un mot ou une phrase dont l'ordre des lettres reste le même si on le lit de gauche à droite ou de droite à gauche. Par exemple, «ressasser »et «Engage le jeu que je le gagne » sont des palindromes.
Écrivez une fonction « **palindrome(s)** »qui détermine si une chaîne de caractères est un palindrome. Pensez à vous débarrasser des majuscules et des espaces.

Exercice 7 : (Codage de César simpflifié)
Écrire une fonction **codage(ch, cle)** qui prend en argument une chaîne de caractères et décale ses lettres par un nombre (cle) de crans dans l'alphabet. Par exemple, «bac» sera transformée en «edf» avec cle=3. On utilisera pour cela les fonctions ord et chr qui permettent la conversion des caractères en code ascii (ou unicode).

On écrira aussi une fonction de**codage(ch, cle)** de décodage.

Exercice 8 :
Ecrire une fonction « **bin(n)** » qui renvoie dans une chaîne de caractères le résultat de conversion en binaire de n.

Exercice 9 : (Méthode title)
Écrire une fonction majuscule_mot qui met en majuscules la première lettre de chaque mot d'une chaîne de caractères. Par exemple, «je mange du fromage» donnera «Je Mange Du

Fromage» (on suppose que chaque mot est séparé par un espace et qu'il n'y a pas de symboles de ponctuation). On utilisera pour cela la méthode upper.
Remarque : votre fonction majuscule_mot existe déjà en Python sous forme de la méthode title.

Exercice 10 (ADN, ??)

On représente un brin d'ADN par une chaîne de caractères qui peut contenir quatre caractères différents : 'A' (Adénine), 'C' (Cytosine), 'G' (Guanine) et 'T' (Thymine).

1. Écrire une fonction estADN(ch) qui prend en argument une chaîne de caractères et renvoie True si cette chaîne correspond à un brin d'ADN et qui renvoie False sinon. Par exemple, estADN("TTGAC") et estADN("GCAATAG") renvoient True mais estADN("AMOG") et estADN("CaTg") renvoient False.

2. Ecrire une fonction masseMolaire(ch) qui calcule la masse molaire d'une séquence ADN passée en argument. Chaque lettre a une masse donnée : 'A' (135 g/mol) ; 'T' (126 g/mol) ; 'G' (151 g/mol) ; 'C' (111 g/mol). La masse totale est la somme des masses des lettres de la séquence.
   Par exemple, masseMolaire("AGATC") renvoie (135 + 151 + 135 + 126 + 111) g/mol, c'est-à-dire 658 g/mol.

Chaque base possède une base complémentaire avec laquelle elle peut s'associer : "A" et "T" sont complémentaires, et "C" et "G" sont complémentaires.

3. Écrire une fonction brinComp(b) qui étant donné un brin d'ADN b calcule et renvoie son brin complémentaire, c'est à dire le brin constitué des bases complémentaires de b. Par exemple, brinComp("A") renvoie "T" et brinComp("AAGT") renvoie "TTCA".

4. Écrire une fonction sous_sequence(A, B) qui prend en argument deux chaînes de caractères représentant des brins d'ADN et renvoie True si le premier brin est une sous-séquence du deuxième, et qui renvoie False sinon.
   Par exemple, sous_sequence("ATC", "GGTATCG") renvoie True et sous_sequence("GC", "AAT") renvoie False.