Introduction

Une étudiante en biologie que vous venez de rencontrer vous demande de l'aide pour un problème de bioinformatique. Grâce aux progrès des techniques de séquençage automatiques de l'ADN, il est maintenant facile d'obtenir l'ADN d'un être vivant. L'ADN est composé de quatre types de bases : **A**dénine, **C**ytosine, **G**uanine et **T**hymine. Pour manipuler ces séquences ADN, les biologistes ont pris l'habitude de les représenter sous la forme d'une (longue) suite de caractères représentants les 4 bases qui forment l'ADN. A titre d'exemple, voici un extrait de séquence ADN provenant de la base de données GenBank:

```
LOCUS       SCU49845     5028 bp    DNA             PLN       21-JUN-1999
DEFINITION  Saccharomyces cerevisiae TCP1-beta gene, partial cds, and Axl2p
            (AXL2) and Rev7p (REV7) genes, complete cds.
...

ORIGIN
        1 gatcctccat atacaacggt atctccacct caggtttaga tctcaacaac ggaaccattg
       61 ccgacatgag acagttaggt atcgtcgaga gttacaagct Aaaacgagca gtagtcagct
      121 ctgcatctga agccgctgaa gttctactaa gggtggataa catcatccgt gcaagaccaa
      181 gaaccgccaa tagacaacat atgtaacata tttaggatat acctcgaaaa taataaaccg
      241 ccacactgtc attattataa ttagaaacag aacgcaaaaa ttatccacta tataattcaa
      301 agacgcgaaa aaaaaagaac aacgcgtcat agaacttttg gcaattcgcg tcacaaataa
...
     4861 ttctccactt cactgtcgag ttgctcgttt ttagcggaca aagatttaat ctcgttttct
     4921 ttttcagtgt tagattgctc taattctttg agctgttctc tcagctcctc atatttttct
     4981 tgccatgact cagattctaa ttttaagcta ttcaatttct ctttgatc
```

Une séquence d'ADN d'un gène peut être très longue. A titre d'exemple, on estime que l'ADN humain contient de l'ordre de 3×10^9 bases découpées en environ 20.000 gènes. L'ADN d'une bactérie contient de l'ordre de 4 millions de bases. Les biologistes ont séquencé l'ADN d'un grand nombre d'espèces qu'ils stockent dans des bases de données spécialisées comme GenBank. Le traitement de toutes ces données a mené à la création de la bio-informatique, discipline qui rassemble des biologistes et des informaticiens.
Afin d'aider cette étudiante en biologie, vous allez écrire durant cette mission vos premiers algorithmes de bio-informatique.

Travail à faire

Les séquences d'ADN sont composées d'un suite de caractères A, T, C et G. Pour traiter de telles séquences en Python, le plus simple est de les manipuler comme des chaînes de caractères.

Votre objectif durant cette mission est de développer quelques fonctions permettant de traiter des séquences d'ADN stockées sous la forme de chaînes de caractères.

1. Écrire une fonction sequenceADN(n) ayant pour paramètre d'entrée un entier naturel et renvoyant une séquence d'ADN aléatoire de longueur n. On pourra utiliser la fonction choice(range(4)) de la bibliothèque random pour obtenir un nombre aléatoire entre 0 et 3 avec la correspondance

0	1	2	3
A	T	C	G

2. La deuxième fonction à écrire est la fonction is_adn(s). Elle prend comme argument une chaîne de caractères s et retourne True si la chaîne de caractères contient uniquement les caractères a, t, c ou g (à la fois en majuscules et en minuscules) et False sinon. Une chaîne de caractères vide ("") n'est pas considérée comme étant de l'ADN.

3. La troisième fonction à écrire est la fonction positions(s, p). Elle prend comme arguments deux chaînes de caractères s et p. Elle retourne les positions des occurences de p dans s. Par exemple, pour ACGACCG (majuscules) et cg (minuscule) le résultat doit être [1,5]. Vous ne pouvez *pas* utiliser la fonction find de Python.

4. La quatrième fonction à écrire est baptisée distance_h. Elle calcule la distance de Hamming (http://fr.wikipedia.org/wiki/Distance_de_Hamming) entre deux chaînes de caractères de longueurs égales. En théorie de l'information, cette distance est définie comme étant le nombre de positions où les deux chaînes de caractères diffèrent. Voici quelques exemples qui devraient vous aider à mieux comprendre cette distance :

Chaîne 1	Chaîne 2	Distance
A	A	0
AG	GG	1
AG	AT	1
ATGAC	ATGAC	0

ATGAC	AGGAG	2
ATGAC	TGACG	5

Si les chaînes n'ont pas la même longueur, la fonction doit retourner None.

Pour des fonctions qui peuvent être exécutées sur des chaînes de caractères, regardez ici.

5. La cinquième fonction à écrire est distances_matrice(l). Étant donné une liste de chaînes de caractères, la fonction doit calculer une matrice des distances de Hamming entre toutes ces chaînes de caractères. Par exemple, pour cette liste: ["AG", "AT", "GT", "ACG", "ACT"] la fonction doit retourner:

```
[ [    0,   1,    2, None, None ],
  [    1,   0,    1, None, None ],
  [    2,   1,    0, None, None ],
  [ None, None, None, 0,    1    ],
  [ None, None, None, 1,    0    ] ]
```

6. La sixième fonction à écrire est BrinComplementaire qui a comme paramètre une liste ADN et retourne son complémentaire sous forme d'une nouvelle liste ADN. Vous pouvez utiliser les deux constantes tableaux : NUC=['A','C','G','T'] et NUCINV=['T','G','C','A']

7. Dans les séquences d'ADN, on retrouve parfois des palindromes. Un palindrome est un mot dont l'ordre des caractères reste le même qu'on le lise de gauche à droite ou de droite à gauche, comme "radar" ou "kayak". Une séquence ADN telle que CTAGGATC est un exemple de palindrome. A titre d'exemple, le plus long palindrome de la séquence ACCTGTTAGGATTC est TTAGGATT. Certains palindromes ont un rôle particulier d'un point de vue biologique et il est intéressant de pouvoir trouver dans une séquence donnée le plus long palindrome.

   Votre dernier objectif dans cette activité est d'écrire la fonction plus_long_palindrome qui prend comme argument une chaîne de caractères et retourne une chaîne de caractères contenant le plus long palindrome de la chaîne passée en argument. Dans cette phase de réalisation, nous considérons qu'un caractère unique est lui-même un palindrome. Lorsque la fonction ne trouve aucun palindrome dans une chaîne de caractères, elle devra retourner "".

   Détecter le plus long palindrome dans une chaîne de caractères est un problème compliqué. En informatique, lorsque l'on est face à un problème compliqué, la meilleure approche pour le résoudre est de le *découper en petits problèmes* plus simples. Il suffit ensuite d'écrire une fonction pour résoudre chaque petit problème et de combiner ces fonctions pour résoudre le problème compliqué.

Pour rechercher le plus long palindrome, une piste est de d'abord écrire une fonction permettant d'extraire d'une chaîne de caractères de longueur *n* les sous-chaînes de longueur *n-1*, *n-2*, ...