



Epreuve d'informatique

1^{er} semestre AU : 2022 – 2023

Date : octobre 2022 ; Durée : 1H ; section : BG2 ; Nombre de page : 2

L'utilisation des calculatrices n'est pas autorisée pour cette épreuve.

Le langage de programmation sera obligatoirement Python.

*

* *

Implémentation. Dans ce sujet, nous adopterons la syntaxe du langage Python. On rappelle qu'en Python, il est important de bien respecter les indentations car elles permettent de définir des blocs.

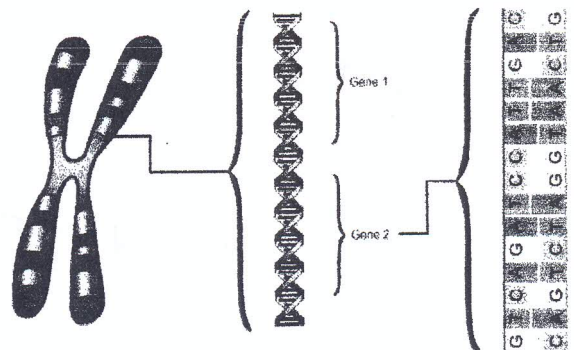
Problème

On souhaite extraire des informations sur des gènes à partir d'un fichier nommé GenBank.txt. Pour cela, il faudra récupérer les lignes décrivant la position des gènes. Voici par exemple les quatre premières lignes de ce fichier :

```
gene; complement ; 1807 ; 2169  
gene ;2707;2934  
gene ; complement ; 9016;9853  
gene ;complement ; 10568 ; 11951
```

Lorsque la ligne contient le mot complément le gène est situé sur le brin complémentaire, sinon il est situé sur le brin direct.

Un gène est formé par différentes bases. Votre code devra récupérer le premier et deuxième nombre de chaque ligne, indiquant respectivement la position du début et de la fin du gène. Attention à bien les convertir en entier afin de pouvoir calculer la longueur du gène.



Travail demandé :

1. Ecrire une fonction **proprietes_gene(gene)** qui prend en paramètre une chaîne de caractère qui présente un gène et qui retourne ses propriétés sous forme d'un tuple.

```
>>> proprietes_gene('gene ;complement ;1807 ;2169')  
( 'gene', 1, 'complement', 1807, 362)
```

2. Ecrire une fonction **est_complementaire(gene)** qui renvoie True si le gène passé en paramètre est complémentaire, False sinon.

```
>>> proprietes_gene('gene ;complement ;1807 ;2169')  
True
```

3. Ecrire une fonction **longueur(gene)** qui renvoie la longueur d'un gène passé en paramètre.

```
>>> proprietes_gene('gene ;complement ;1807 ;2169')  
362
```

4. Ecrire une fonction **lireGenes(nomF)** qui prend en paramètre le nom du fichier **nomF** et retourne une liste de tuple dont chacun représente les propriétés d'un gène lu à partir du fichier.

```
>>> lireGenes('GenBank.txt')  
[( 'gene', 'complement', 1807, 2169), ('gene', 2707, 2934), ('gene', 'complement', 9016, 9853), ... ]
```

5. Ecrire une fonction **count_gene(nomF)** qui renvoie un dictionnaire dont :
 - les clés sont : 'complémentaires' et 'direct' .
 - et les valeurs sont les nombres de gènes direct et complémentaire présents dans le fichier **nomF** passé en paramètres.

```
>>> count_gene('GenBank.txt')  
{ 'complementaire' : 145 , 'direct' : 143 }
```

6. Ecrire une fonction **mean_gene_direct(nomF)** qui renvoie la moyenne des longueurs des gènes directs présents dans le fichier **nomF** passé en paramètre.