

## 1. Exercice Python

Dans cet exercice nous allons utiliser la fonction **un\_nombre()** qui génère aléatoirement un entier entre 1 et 4 inclus.

```
def un_nombre() :  
    return int(random.uniform(1,5))
```

- 1.1. Écrivez une fonction **un\_nucleotide()** qui, en utilisant la fonction **un\_nombre()**, génère aléatoirement un nucléotide et renvoie ce nucléotide ('A', 'C', 'G' ou 'T').
- 1.2. Écrivez une fonction **uniformite(n)** qui réalise n générations aléatoires de nucléotides et affiche ensuite le nombre de fois que chaque nucléotide a été généré.
- 1.3. Écrivez une fonction **nb\_generations(nucleotide)** qui prend en paramètre un nucléotide ('A', 'C', 'G' ou 'T') et qui, en utilisant la fonction **un\_nucleotide()**, retourne le nombre de générations aléatoires avant d'obtenir ce nucléotide. Le nombre de générations correspond au nombre d'appels de la fonction **un\_nucleotide()** qu'il sera nécessaire d'effectuer pour obtenir le nucléotide donné en paramètre.
- 1.4. En utilisant la fonction **nb\_generations(nucleotide)**, écrivez une fonction **moyenne\_tentatives(nucleotide, n)** qui calcule la moyenne du nombre de tentatives avant d'obtenir le nucléotide donné, sur un échantillon de n générations de ce nucléotide. Par exemple **moyenne\_tentatives('G', 10)** va générer des nucléotides jusqu'à obtenir 10 fois le nucléotide 'G' et la fonction renvoie la moyenne du nombre de tentatives pour arriver à générer ces 10 occurrences de 'G'.
- 1.5. Écrivez une fonction **generation\_sequence(n)** qui génère de manière aléatoire une séquence d'ADN de taille n et qui renvoie la séquence générée. Aide : pour concaténer deux chaînes de caractères vous pouvez utiliser l'opérateur +.

1.6. Écrivez une fonction **encodage(sequence)** qui prend en paramètre une séquence d'ADN et qui, en considérant que chaque nucléotide a un chiffre associé (par exemple 'A'=1, 'C'=2, 'G'=3 et 'T'=4), réécrit la séquence en tant que nombre, et le renvoie. Par exemple, pour la séquence « AAGCTAG » la fonction renverra le nombre 1132413. *Attention : il s'agit du nombre 1132413 et non pas la chaîne de caractères « 1132413 » ; pour construire le nombre attendu vous pouvez réaliser des multiplications par 10 et des additions successives. Aide : pour obtenir la taille d'une chaîne de caractères vous pouvez utiliser la fonction **len()**.*

1.7. Écrivez une fonction **decodage(nombre)** qui réalise l'opération inverse par rapport à la fonction précédente, c'est à dire que à partir d'un nombre composé uniquement des chiffres 1, 2, 3 et 4 la fonction renvoie la séquence d'ADN correspondante. Par exemple, pour le nombre 1132413, la fonction renverra la séquence « AAGCTAG ». Pour cette fonction on considère la même association nucléotide chiffre que pour la question précédente.