

## Informatique en CPGE (2018-2019) TP 6 : les types composés

### Exercice 1

Le code génétique humain peut être vu comme une chaîne, simple mais longue, (environ 3 milliards de caractères), formée avec les caractères A, C, G et T. Une partie de l'analyse du code consiste en la recherche de certains mots.

1. Ecrire une fonction `chaîne` qui prend en argument un entier `n` et renvoie une chaîne composée de `n` caractères tirés successivement au hasard parmi les caractères 'A', 'C', 'G' et 'T'.
2. Définir une chaîne nommée `dna` de 1000 caractères tirés au hasard parmi les caractères 'A', 'C', 'G' et 'T'.
3. Ecrire une fonction `compteurACGT` qui prend en argument une chaîne de caractères puis calcule combien de fois chacun des caractères 'A', 'C', 'G' et 'T' apparaît dans la chaîne. La fonction renvoie un quadruplet (type tuple) constitué des quatre valeurs calculées.  
Appliquer cette fonction à la chaîne `dna`.
4. Reprendre les trois questions précédentes en remplaçant la chaîne de caractères par une liste de caractères.

### Exercice 2

En cryptographie, le chiffre de César est une méthode permettant de coder un texte. Chaque lettre du texte original est remplacée par une lettre se trouvant à une distance constante, la clé, fixée au début, et toujours du même côté dans l'alphabet. Par exemple si la clé est 3, la chaîne "bonjour" devient "erqmxu"; si la clé est 7, la chaîne "BONJOUR" devient "IVUQVBY".

Ecrire une fonction `chiffre` qui prend en arguments une chaîne de caractères et une clé (nombre entier naturel) et renvoie la chaîne codée. On pourra utiliser le code ASCII avec les fonctions `ord` et `chr`. Rappel : l'instruction `chr(65)` permet d'afficher 'A' et l'instruction `ord('A')` permet d'afficher 65. Attention : on ne code que les lettres de l'alphabet (minuscules ou majuscules); les signes de ponctuation ou les espaces ne sont pas codés.

Ecrire une fonction `dechiffre` qui prend en arguments une chaîne de caractères codée et la clé utilisée au codage et renvoie la chaîne décodée.

### Exercice 3

1. Ecrire une fonction `moyenne` prenant en argument une liste de nombres, qui calcule et renvoie la moyenne des nombres.
2. Recherche du minimum ou du maximum.
  - (a) Ecrire une fonction `minimum` prenant en argument une liste de nombres nommée `tab`, qui calcule et renvoie la valeur minimale de ces nombres. Pour cela on initialise le minimum à `tab[0]`, puis on compare avec `tab[1]` et on garde la plus petite valeur. Puis on compare cette valeur avec `tab[2]` et on garde la plus petite valeur. On continue ainsi jusqu'à la fin du tableau.
  - (b) Ecrire une fonction `maximum` qui prend en argument une liste de nombres, calcule et renvoie la valeur maximale de ces nombres.
3. Ecrire une fonction `mediane` qui prend en argument une liste de nombres, qui calcule et renvoie la médiane de cette liste. On pourra utiliser la méthode `sort()` ou la fonction `sorted` qui permettent d'ordonner une liste de nombres.
4. Ecrire une fonction `statistiques` qui prend en argument une liste de nombres et qui renvoie la moyenne de ces nombres, la valeur maximale, la valeur minimale et la médiane.

#### Exercice 4

Nous travaillons ici avec des matrices carrées (2,2), soit deux lignes et deux colonnes.

La liste  $[[a, b], [c, d]]$  représente la matrice  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ .

Nous confondons pour la suite la matrice et la liste qui la représente.

1. Ecrire une fonction `produit(m1, m2)` prenant en arguments deux matrices nommées `m1` et `m2`, qui calcule et renvoie leur produit.
2. On considère les matrices  $A = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$  et  $B = \begin{pmatrix} 4 & 5 \\ 2 & 3 \end{pmatrix}$ .

Calculer avec ce programme les produits  $AB$  puis  $BA$ . Vérifier le résultat.

3. Ecrire une fonction `permute_lignes(m)` qui prend en argument une matrice `m` et renvoie une nouvelle matrice obtenue en permutant les deux lignes de `m`. La matrice initiale `m` doit rester inchangée.
4. Ecrire une fonction `permute_colonnes(m)` qui prend en argument une matrice `m` et renvoie une nouvelle matrice obtenue en permutant les deux colonnes de `m`. La matrice initiale `m` doit rester inchangée.