

**TP 3: Feuille d'Exercices**  
*Listes et chaînes de caractères*

**Exercice 1.**

- (1) Ecrire deux fonctions de paramètre  $N$  entier naturel, et qui retournent la liste des  $N$  premiers termes de la suite arithmétique de raison 12 et de premier terme 1 :

$$\forall n \in \mathbb{N}, u_n = 12n + 1$$

par deux méthodes différentes :

- (a) A l'aide d'une boucle.
  - (b) Par compréhension de liste.
- (2) Comparer les temps d'exécution des deux méthodes à l'aide de la fonction `clock()` du module `time`, et du petit script suivant :

```
from time import clock
def duree(fonction, n=1000000) #Durée d'exécution de fonction(n)
    debut = clock()           # temps début
    fonction(n)                # appel de fonction(n)
    fin = clock()              # temps fin
    return fin - debut         # durée retournée en seconde
```

**Exercice 2.**

- (1) Ecrire une fonction `moyenne()` prenant en paramètre une liste de nombres et qui retourne sa moyenne.
- (2) Ecrire une fonction `variance()` prenant en paramètre une liste de nombres et qui retourne sa variance.
- (3) On pourra les tester sur des listes de nombres aléatoires que l'on aura créées à l'aide de la fonction `random()` du module `random` qui renvoie un décimal (`float`) aléatoire dans  $[0, 1[$ .

**Exercice 3.** Construire par compréhension de liste les listes suivantes :

- (1) La liste des logarithmes népériens ( $\ln$ ) des entiers compris entre 1 et 20.
- (2) La liste des entiers compris entre 0 et 100 qui sont pairs et qui ne sont pas des multiples de 3.
- (3) La liste `['@-@', '@--@', '@---@', '@----@', '@-----@']`.
- (4) La liste des couples d'entiers entre 0 et 10 dont la somme est un multiple de 5.

- (5) Ecrire une fonction qui prend pour paramètre un entier `n` et retourne sous forme d'une liste de listes la matrice identité de rang `n`.

**Exercice 4.** Les caractères non accentués, chiffres et autres symboles accessibles au clavier ou non, sont disponibles dans la table ASCII de caractères, numérotés de 0 à 255 (0xFF).

La fonction `chr(n)` retourne le caractère de la table de code ASCII `n`.

La fonction `ord()` retourne le code ASCII d'un caractère dans la table.

(les caractères de 'a' à 'z' sont rangés dans l'ordre alphabétique de 97 à 122, les caractères de 'A' à 'Z' de 65 à 90.)

- (1) Ecrire un script pour afficher à l'écran les caractères de la table ASCII de code allant de 33 à 126. On les présentera par ligne de 16 caractères.
- (2) Ecrire une fonction `maj()` qui prend en argument une chaîne de caractère non accentuée et qui renvoie la chaîne avec minuscules converties en majuscules.

**Exercice 5.** Le chiffrement de César.

C'est une méthode très simple de chiffrement de messages en un texte crypté pour le rendre "illisible" à qui n'en a pas la clef. Dans un message en lettres majuscules supprimer tous les espaces et symboles, puis changer chaque lettre par la N-ième lettre suivante dans l'ordre alphabétique (après z on reprend en a), où N est un entier entre 1 et 25, c'est la clé de chiffrement.

- (1) Ecrire une fonction `CesarCrypt()` qui prend deux paramètres, la clef de chiffrement N et une chaîne de caractère en majuscule sans symbole ni espace, et renvoie le chiffrement de César de la chaîne.
- (2) Ecrire une fonction `CesarDecrypt()` qui prend deux paramètres, un texte crypté et la clef, et retourne le texte décrypté.