

**SÉRIE N° 18 : LES LISTES & LES TUPLES –SUITE–****EXERCICE 8:**

Écrire une fonction **NombresPairs** qui retourne le nombre de nombres pairs dans l'intervalle [2 , 10000] inclus

EXERCICE 9: Recherche séquentielle

Écrire une fonction **RechercheIndice** qui prend en entrée une valeur quelconque **x** et une liste d'entiers **L**, et fournit en sortie :

- l'indice de l'élément **x** dans la liste **L** si **x** est dans **L** (l'indice est alors compris entre zéro et la longueur de la liste moins un)
- la longueur de la liste **L** si **x** n'est pas dans **L**.

EXERCICE 10: Recherche séquentielle

Écrire une fonction **RechercheTousIndices** qui prend en entrée une valeur quelconque **x** et une liste d'entiers **L**, et fournit en sortie un tuple de tous les indices de **x** dans **L**

EXERCICE 11: Recherche dichotomique

On suppose maintenant qu'on a une liste d'entiers triée en ordre croissant.

Écrire une fonction **RDindice** qui prend en entrée une valeur quelconque **x** et une liste d'entiers **L**, et fournit en sortie :

- l'indice de l'élément **x** dans la liste **L** si **x** est dans **L** (l'indice est alors compris entre zéro et la longueur de la liste moins un)
- la longueur de la liste **L** si **x** n'est pas dans **L**.

EXERCICE 12:

Écrire une fonction **longueurMoyenne** qui prend en entrée une liste dont les éléments sont des chaînes de caractères (par exemple des séquences d'ADN), et qui retourne en sortie la longueur moyenne des chaînes appartenant à la liste.

- On prendra soin de renvoyer un nombre réel (plus précis qu'un entier).
- Si la liste est vide, renvoyer 0.

EXERCICE 13:

Écrire une fonction récursive **Recherche(L,x)** qui prend en paramètre une liste de nombres **L** et un nombre **x** et qui permet de tester l'existence de **x** dans la liste

EXERCICE 14:

Écrire une fonction récursive **binaire(n)** qui calcule la décomposition binaire de tout nombre entier **n** et renvoie le résultat sous forme de liste de 0 et de 1

Exemple : L'appel de la fonction **binaire(13)** retourne la liste [1,1,0,1]