Listes

Exercice 1

1. Ecrire une fonction Python **distincts** qui prend en argument une liste et qui renvoie **True** si cette liste a au moins deux éléments distincts et **False** sinon.

Fonctions récursives

Exercice 2

Soit (u_n) une suite définie par :

$$\begin{cases} u_0 = 1 \\ u_1 = 1 + \sqrt{3} \\ \forall n \in \mathbb{N},\ u_{n+2} = 2u_{n+1} - 4u_n \end{cases}$$

Ecrire une fonction Python récursive **suite** qui prend en argument un entier naturel **n** et qui renvoie la valeur de u_n.

Exercice 3

*Dans tout cet exercice, on s'interdira d'utiliser la commande ** pour calculer des puissances.*

Dans tout cet exercice, x désigne un nombre réel et n désigne un entier naturel.

1. Ecrire une fonction Python **puissance1** récursive qui prend en argument x et n et qui renvoie la valeur de x^n.
2. (a) *Question mathématique.* Vérifier que :
$$\begin{cases} x^n = \left(x^{\frac{n}{2}}\right)^2 & \text{si n est pair} \\ x^n = x\left(x^{\frac{n-1}{2}}\right)^2 & \text{si n est impair} \end{cases}$$
   (b) Ecrire une fonction Python **puissance2** récursive qui prend en argument x et n et qui renvoie la valeur de x^n. Cette fonction reposera sur les formules de la question précédente.
3. Pour estimer la durée d'exécution d'une fonction Python, on peut utiliser la syntaxe suivante :

```
import time as t

def fonction(arg) :
    debut = t.time()
    ...
    ...
    fin = t.time()
    return ... , fin - debut
```

Faire des tests permettant de comparer les vitesses d'exécution des deux fonctions **puissance1** et **puissance2**.

Dichotomie

Exercice 4

1. Ecrire une fonction Python **recherche** qui prend en argument une liste de nombres **L** non vide supposée triée et un nombre **x** et qui renvoie **True** si **x** est présent dans la liste et **False** sinon.
   Cette recherche ne sera pas faite par dichotomie.
2. Ecrire une fonction Python **recherche_dichotomie** qui prend en argument une liste de nombres **L** non vide supposée triée et un nombre **x** et qui renvoie **True** si **x** est présent dans la liste et **False** sinon.
   Cette recherche sera faite par dichotomie.

Exercice 5

1. Aller dans le dossier de la classe et copier / coller les listes **test1**, **test2** et **test3** dans votre fichier Python.
2. Ecrire une fonction Python **croissante** qui prend en argument une liste de nombres **L** non vide et qui renvoie **True** si la liste **L** est triée par ordre croissant et **False** sinon.
3. Pour chacune de ces listes :
   (a) Donner sa longueur.
   (b) Dire si elle est triée par ordre croissant.
   (c) Donner son minimum et son maximum.
   (d) Dire si le nombre 789 y est présent en utilisant deux méthodes différentes et comparer les durées d'exécution.

Exercice 6

1. Ecrire une fonction Python **dichotomie** qui prend en argument une fonction $\mathtt{f}$ supposée continue strictement monotone sur un intervalle $[\mathtt{a}, \mathtt{b}]$ dont on sait qu'elle s'annule une unique fois sur $[\mathtt{a}, \mathtt{b}]$ en un réel $\mathtt{x}_0$, les deux réels $\mathtt{a}$ et $\mathtt{b}$, ainsi qu'un réel $\mathtt{e}$ strictement positif et qui renvoie un encadrement de $\mathtt{x}_0$ à $\mathtt{e}$ près.
   Cet encadrement sera obtenu par dichotomie.
2. Ecrire une fonction Python **approx** qui renvoie la liste des encadrements à 10^{-5} près de $\sqrt{\mathtt{k}}$ pour $\mathtt{k}$ compris entre 0 et 100.
3. Estimer la durée d'exécution de la fonction précédente.