DS n° 04

— Faire tous les exercices dans un même fichier NomPrenom.py à sauvegarder,
— mettre en commentaire l'exercice et la question traités (ex : # Exercice 1),
— ne pas oublier pas de commenter ce qui est fait dans votre code (ex : # Je crée une fonction pour calculer la racine d'un nombre),
— il est possible de demander un déblocage pour une question mais uniquement celles avec une $\star$. Celle-ci sera notée 0,
— il faut vérifier avant de partir que le code peut s'exécuter et qu'il affiche les résultats que vous attendez.

Exercice 1

Pour tout entier naturel n, on considère la fonction f_n définie sur $\mathbb{R}$ par :

$$f_n(x) = x^{n+1} - x^n - 1.$$

On admet les résultats suivants sur f_n :

- le tableau de variations de f_n est :

x	1		α_n		$+\infty$
f_n	-1	$\nearrow$	0	$\nearrow$	$+\infty$

- $f_n(2) > 0$ pour $n \geq 1$.

Donc il existe un unique $\alpha_n \in [1, 2]$ tel que f_n s'annule en α_n. L'objectif est d'étudier la suite $(\alpha_n)_{n\in\mathbb{N}}$.

1. $\star$ Soit n fixé en amont. Ecrire une fonction `f` qui prend comme argument x et renvoie la valeur de $f_n(x)$.
2. Afficher la valeur de $f_{10}(1)$.
3. Tracer les courbes de f_n, pour n entier de 1 à 10. La plage d'affichage sera le rectangle $[1, 2] \times [-1, 1]$ (fonctions `xlim` et `ylim` du module `matplotlib.pyplot` en *Python*).
4. Conjecturer le comportement de la suite $(\alpha_n)_{n\geq 0}$: monotonie, limite (réponse en commentaire).
5. $\star$ Écrire une fonction `dichotomie` qui prend comme entrée une fonction f, les bornes initiales a et b, la précision ε et qui renvoie une approximation d'une solution à ε près de l'équation $f(x) = 0$ de l'intervalle $[a, b]$.
6. En utilisant l'algorithme précédent, déterminer des valeurs approchées de α_n à 10^{-6} près pour n variant de 2 à 200. Quelle conjecture pouvez-vous faire ? (réponse en commentaire).

Exercice 2

1. Préliminaires :
   Ecrire une fonction `maxi` qui prend comme argument une liste et renvoie la valeur maximale ainsi que sa position dans la liste. Si il y a plusieurs maxima, la fonction ne renvoie que la première position.
   Dans cette question l'utilisation des fonctions Python `max` *et* `index` *sont interdites.*

```
>>>L=[1,8,1,-2,-7,8]
>>>maxi(L)
(8,1)
```

TOURNEZ LA PAGE.

Dans cet exercice, on manipule des suites (finies) d'entiers sous la forme de listes d'entiers. Ainsi la suite $(0, 1, 3, 8, 8)$ sera représentée par la liste `[0,1,3,8,8]`.
La liste est croissante (respectivement décroissante, monotone) si la suite est croissante (respectivement décroissante, monotone).

2. Monotonie :
   (a) Ecrire une fonction `estCroissante` qui teste si une liste d'entiers est croissante. La fonction renvoie un booléen `True` ou `False`.
   (b) Afficher le résultat de la fonction `estCroissante` pour la liste `L=[0,1,3,8,8]`. (La réponse doit être `True`).
   (c) Écrire de même une fonction `estDecroissante` qui teste si une liste d'entiers est décroissante.
   (d) Écrire de même une fonction `estMonotone` qui teste si une liste d'entiers est monotone.
3. Soit la liste $L = [u_0, u_1, \cdots, u_{n-1}]$ de longueur n. On appelle tranche de L une liste de la forme $[u_i, u_{i+1}, \cdots, u_j]$ où $0 \leq i \leq j < n$.
   On cherche une tranche de L croissante et de longueur maximale.
   Par exemple, une tranche croissante de longueur maximale de $[0, 1, 0, 1, 3, 3, 5, 0, 1, 7]$ est $[0, 1, 3, 3, 5]$, correspondant aux indices $i = 2$ et $j = 6$.
   (a) $\star$ Écrire une fonction `LC` de deux arguments, une liste `L` et un entier `p`, qui renvoie l'entier `d` tel que la liste $[\mathtt{u_p}, \cdots, \mathtt{u_d}]$ est la tranche croissante de L la plus longue possible en partant de la `p`ième position. Autrement dit, $[\mathtt{u_p}, \cdots, \mathtt{u_d}]$ est croissante avec ou bien `d=n-1` ou bien $\mathtt{u_d} > \mathtt{u_{d+1}}$.
   (b) Afficher le résultat de `LC` pour la liste `L=[0,1,0,1,3,3,5,0,1,7]` et l'entier 2.
   (c) Écrire une fonction `maxCroissante` d'argument une liste `L` qui renvoie la plus longue tranche croissante de `L`. S'il n'y a pas unicité, on renvoie la première trouvée.
4. Soit la liste $L = [u_0, u_1, \cdots, u_{n-1}]$ de longueur n. Une monotonie de L est un couple d'indices (i, j) tel que $0 \leq i < j < n$, que la sous-liste $[u_i, u_{i+1}, \cdots, u_j]$ est monotone et qu'elle ne l'est plus si on l'étend, à droite ou à gauche, d'un élément supplémentaire (lorsque c'est possible). La monotonie est dite "banale" lorsque $j = i + 1$.
   (a) Proposez une liste d'entiers de longueur 5 qui ne présente que des monotonies banales. (réponse en commentaire)
   (b) Écrire une fonction `cahots`, de complexité linéaire, qui teste si une liste ne comporte que des monotonies banales.
   (c) Après avoir créé une liste arbitraire `L` de valeurs toutes distinctes, on peut l'ordonner par `L.sort()`. Imaginer ensuite une méthode pour réordonner de manière à ce qu'elle ne comporte que des monotonies banales.