DEVOIR DE SYNTHESE SEMESTRE 2

Matière : *INFORMATIQUE*

Filières : MP1, PC1, PT1, PB1 - Durée : 2 h

Toutes les réponses aux questions doivent être rédigées en ***Python*** *et non pas en syntaxe algorithmique.*

Exercice 1 : Méthode de la sécante (4 points)

La méthode de la sécante est une technique de résolution d'équations algébriques par approximations successives.

Soit f: $[a, b] \longrightarrow \mathbb{R}$ une fonction continue et strictement croissante telle que $f(a)<0$, $f(b)>0$. Alors, la suite définie par :

$$x_0 = a,\ x_1 = b \quad et \quad x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \qquad (n \text{ étant un entier} \geq 1)$$

est croissante et converge vers la solution de l'équation $f(x) = 0$.

Pour évaluer numériquement la solution de l'équation $f(x)= 0$ sur l'intervalle $[a, b]$, on calcule la suite des valeurs x_n. On arrête le calcul lorsque la différence absolue entre deux valeurs successives devient inférieure à une certaine précision donnée : epsilon = 10^{-6}.

La fonction $f(x)=xe^x-(x+1)$ admet une racine unique dans l'intervalle $[0, \pi/2]$.

Définir :

1) Une fonction **f** qui prend en argument un réel ***x*** et renvoie : $xe^x-(x+1)$.

2) Une fonction **Secante** qui prend en argument deux réels ***a*** et ***b*** et renvoie la solution approximative de l'équation $f(x) = 0$ en appelant la fonction f.

Exercice 2 : Tracé des données d'un fichier (4 points)

Des données expérimentales sont disponibles sous forme d'un fichier texte nommé «data.txt » dans lequel les informations sont présentées sous forme de colonnes. La première colonne du fichier correspond au temps (unité : s), la deuxième au débit molaire d'un gaz A (unité : mol/s) enregistré au cours d'une première expérience, et la troisième colonne au débit molaire de A (unité : mol/s) enregistré au cours d'une seconde expérience.

```
00.0  1.9231E-04  1.9230E-04
10.0  1.8293E-04  1.9108E-04
20.0  1.7401E-04  1.8991E-04
30.0  1.6552E-04  1.8878E-04
40.0  1.5745E-04  1.8771E-04
50.0  1.4977E-04  1.8667E-04
...
```

1) Donner le code permettant de :
- ouvrir le fichier «data.txt » qui contient les données expérimentales.
- créer trois listes T, A1 et A2 correspondant respectivement aux données de la première, deuxième et troisième colonne.
- déterminer le nombre de points expérimentaux n.

2) Donner le code permettant de tracer simultanément sur un graphe les deux courbes représentant les débits molaires de A en fonction du temps au cours des deux expériences.

Exercice 3 : Génération d'un tableau (6 points)

Soit un tableau t de taille n dont les éléments valent 0 ou 1. Le tableau est initialisé comme suit. On définit la suite récurrente d'entiers $(u_i)_{0\leq i\leq n-1}$ par :

– u_0 est un entier positif pour initialiser la suite u_i ;
– pour tout i compris entre 1 et $n-1$, $u_i = (16365 \times u_{i-1}) \bmod 65521$.
Puis, pour $0 \leq i \leq n-1$, on pose $t[i] = u_i \bmod 2$.

1) Ecrire une fonction **calcul** qui prend en argument un entier u_0 et la taille n et retourne le tableau t.

Une coupe $t[i : j]$ est un *palindrome* si elle est égale à la coupe obtenue en prenant ces éléments à l'envers, c'est-à-dire si $t[i] = t[j-1]$, $t[i+1] = t[j-2]$, $t[i+2] = t[j-3]$, etc.

2) Ecrire une fonction **est_palindrome** qui prend en argument une coupe s et renvoie *True* si s est palindrome et *False* sinon.

3) Ecrire une fonction **nbr_palindrome** qui prend en argument un tableau t et un entier p et renvoie le nombre de palindromes de longueur p présents dans le tableau t.

Exercice 4 : Tri cocktail (6 points)

Le tri cocktail ou tri à bulles bidirectionnel est une méthode de tri qui améliore légèrement le tri à bulles dans la mesure où il permet non seulement aux plus grands éléments de migrer vers la fin de la liste mais aussi aux plus petits éléments de se déplacer au début de la liste.

Etude d'un exemple :
L est une liste de cardinalité 8 initialisée avec la série d'entiers [9, 1, 8, 2, 7, 3, 5, 4].
La méthode de tri consiste tout d'abord à faire migrer l'élément le plus grand vers la fin de la liste à l'aide de permutations successives.

Lors de la <u>première passe</u>, le premier parcours vers la <u>droite</u> déplace les éléments plus grands que leur voisin immédiat vers la droite, et en particulier, va déplacer de proche en proche le plus grand élément de la liste à son emplacement <u>définitif</u> en fin de liste. Il ne doit plus être pris en compte.

Première passe :

parcours à droite →

[**9**, 1, 8, 2, 7, 3, 5, 4]
[1, **9**, 8, 2, 7, 3, 5, 4]
[1, 8, **9**, 2, 7, 3, 5, 4]
[1, 8, 2, **9**, 7, 3, 5, 4]
[1, 8, 2, 7, **9**, 3, 5, 4]
[1, 8, 2, 7, 3, **9**, 5, 4]
[1, 8, 2, 7, 3, 5, **9**, 4]
[1, 8, 2, 7, 3, 5, 4, **9**]

Dans cette phase, les éléments du tableau sont parcourus de gauche à droite : dès que l'on rencontre deux éléments consécutifs qui ne sont pas ordonnés (l'élément (i) doit être inférieur à l'élément (i+1)), on permute leurs positions.

Ensuite, le second parcours vers la gauche va déplacer les éléments plus petits que leur voisin immédiat vers la gauche, et en particulier, déplacera l'élément le plus petit de la liste à son emplacement définitif en tête de liste. Il ne doit plus être pris en compte.

parcours à gauche ←

[1, 8, 2, 7, 3, 5, **4**, 9]
[1, 8, 2, 7, 3, **4**, 5, 9]
[1, 8, 2, 7, **3**, 4, 5, 9]
[1, 8, 2, **3**, 7, 4, 5, 9]
[1, 8, **2**, 3, 7, 4, 5, 9]
[1, **2**, 8, 3, 7, 4, 5, 9]
[**1**, 2, 8, 3, 7, 4, 5, 9]

Dans cette deuxième phase, les éléments du tableau sont parcourus de droite à gauche : dès que l'on rencontre deux éléments consécutifs qui ne sont pas ordonnés (l'élément (i-1) doit être inférieur à l'élément (i)), on permute leurs positions.

De même, lors de la seconde passe, on relance la remontée de la plus grande valeur vers la fin de la liste, sans prendre en compte la première et la dernière valeur. On travaille alors sur l'extrait de liste [2, 8, 3, 7, 4, 5] :

Deuxième passe :

parcours à droite →

[1, **2**, 8, 3, 7, 4, 5, 9]
[1, 2, **8**, 3, 7, 4, 5, 9]
[1, 2, 3, **8**, 7, 4, 5, 9]
[1, 2, 3, 7, **8**, 4, 5, 9]
[1, 2, 3, 7, 4, **8**, 5, 9]
[1, 2, 3, 7, 4, 5, **8**, 9]

Puis on relance la descente de plus petite valeur vers le début de la liste en travaillant uniquement sur l'extrait [2, 3, 7, 4, 5]. Et ainsi de suite…

Question :

Ecrire une fonction itérative **Tri_cocktail** qui prend en entrée une liste **L** et qui renvoie une liste triée dans le sens croissant.