

TP 6 : Listes

- △ • Penser à ouvrir un nouveau fichier pour chaque exercice et à l'enregistrer immédiatement en lui donnant un nom approprié.
- On commentera chaque fonction écrite en expliquant en une ligne ce qu'elle calcule.
- Penser à tester les fonctions sur des exemples simples.

1 Applications directes

Exercice 1 *Calculs sur des listes.* On pensera à tester les fonctions suivantes sur des exemples simples.

- Écrire une fonction **somme (L)** qui calcule la somme des éléments de la liste L (si cette liste est vide, le résultat renvoyé devra être 0).
- Écrire une fonction **moyenne (L)** qui calcule la moyenne des éléments de la liste L (cette liste ne devra pas être vide).
- Écrire une fonction **variance (L)** qui calcule la variance de la liste L définie par :

$$\text{variance}(L) = \frac{1}{n} \sum_{k=0}^{n-1} (L[k] - m)^2$$

où n est la longueur de la liste L .

Exercice 2 *Sélection des éléments d'une liste.* Écrire une fonction **selectionner (L, a, b)** qui construit une nouvelle liste obtenue en ne gardant que les éléments de L qui sont compris (au sens large) entre a et b (on supposera $a \leq b$).

Exercice 3 *Liste des termes d'une suite.* On considère la suite (u_n) définie par :

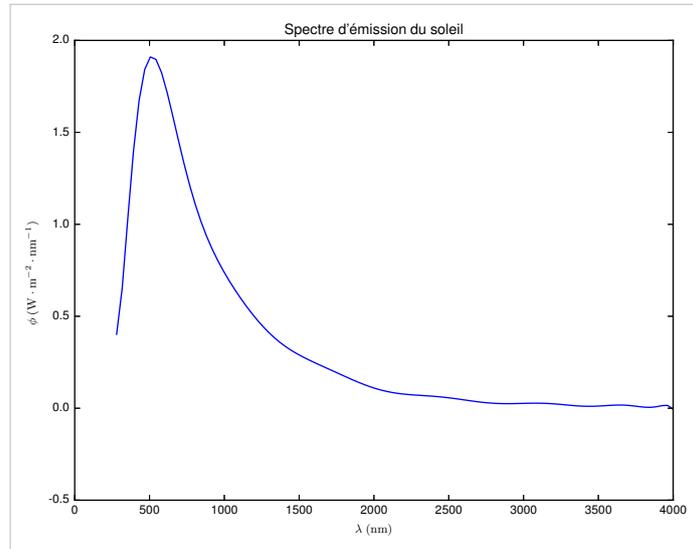
$$u_0 = 1 \\ \forall n \in \mathbb{N}, u_{n+1} = \sqrt{1 + u_n}$$

- Écrire une fonction **liste_termes (n)** qui construit la liste contenant u_0, u_1, \dots, u_n .
- Modifier la fonction précédente pour l'écrire sous la forme **liste_termes (u0, n)** afin de pouvoir changer la valeur de u_0 .

2 Mise en œuvre

Exercice 4 *Indice du maximum.* On considère une liste X de taille n . Écrire une fonction **indice_max (X)** qui retourne un entier k tel que $X[k]$ soit le maximum de $X[0], \dots, X[n-1]$. Tester cette fonction sur quelques exemples.

Exercice 5 Application : loi du déplacement de Wien. On donne ci-dessous le spectre d'émission du soleil¹



Les données numériques correspondant à ces points se trouvent dans deux listes, une liste **Lambda** pour les abscisses et une liste **Phi** pour les ordonnées. Ces deux listes se trouvent dans le fichier **SpectreSolaire.py** qui vous a été envoyé.

- Écrire les instructions permettant de déterminer l'indice **kmax** correspondant à la valeur maximale dans la liste **Phi** (on pourra utiliser la fonction **indice_max** définie dans l'exercice précédent).
- Définir **Lambda_max = Lambda[kmax]** et faire afficher **kmax** et **Lambda_max**.
- On peut alors obtenir une valeur approchée de la température T de la surface du soleil en utilisant la loi du déplacement de Wien :

$$T = \frac{2.898 \cdot 10^6}{\lambda_{\max}}$$

(λ_{\max} est en nanomètres et T en kelvin).

1. Il ne s'agit pas des données brutes, on a éliminé quelques irrégularités.

Corrections

Ex 1.

```
def somme(L):
    # Calcule la somme des termes de la liste L
    # On suppose que L est une liste de nombres
    s = 0
    for x in L:
        s = s+x
    return s
def moyenne(L):
    # Calcule la moyenne des termes de la liste L
    # On suppose que L est une liste de nombres
    return float(somme(L))/len(L)
def variance(L):
    # Calcule la variance de la liste L
    # On suppose que L est une liste de nombres
    m = moyenne(L)
    s = 0
    for x in L:
        s = s+(x-m)**2
    return float(s)/len(L)
T = [1,2]
print(somme(T), moyenne(T), variance(T))
```

3 1.5 0.25

Ex 2.

```
def selection(L, a, b):
    # Construit une nouvelle liste constituée des termes L[k]
    # tels que a<=L[k]<=b
    # On suppose a<=b
    M = []
    for x in L:
        if a<=x<=b:
            M.append(x)
    return M
```

Ex 3. Directement sous la forme `liste_termes(u0, n)` :

```

def liste_termes(u0,n):
    # Construit la liste des termes [u0,...,un]
    # Avec u0 donné et  $u(n+1)=\sqrt{1+u(n)}$ 
    u = u0
    L = [u]
    for i in range(n):
        u = (1+u)**0.5
        L.append(u)
    return L
print(liste_termes(1,10))

```

[1, 1.4142135623730951, 1.5537739740300374, 1.5980531824786175, 1.6118477541252516, 1.616121206508117, 1.6174427985273905, 1.617851290609675, 1.6179775309347393, 1.6180165422314876, 1.6180285974702324]

Autre possibilité :

```

def liste_termes(u0,n):
    L = [u0]
    for i in range(n):
        L.append((1+L[-1])**0.5)
        # Rappel : L[-1] est le dernier terme de la liste L
    return L
print(liste_termes(1,10))

```

[1, 1.4142135623730951, 1.5537739740300374, 1.5980531824786175, 1.6118477541252516, 1.616121206508117, 1.6174427985273905, 1.617851290609675, 1.6179775309347393, 1.6180165422314876, 1.6180285974702324]

Ex 4.

```

def indice_max(X):
    # Calcule un indice k tel que X[k] soit maximal
    # X est une liste non vide de nombres
    k = 0
    for i in range(1,len(X)):
        if X[i]>X[k]:
            k = i
    return k

```

Un exemple :

```
print(indice_max([0,2,7,3,2,9,1]))
```

5

Ex 5.

```
Lambda = [...]
Phi = [...]
```

```
kmax = indice_max(Phi)
Lambda_max = Lambda[kmax]
print(kmax, Lambda_max)
```

6 505.4545454545455

Calcul de la température :

```
print("Temp. :", 2.898e6/Lambda_max, "K")
```

Temp. : 5733.453237410072 K