TRAVAUX PRATIQUES 1

AUTOUR DES LISTES

Il est souvent utile de collecter plusieurs données sous un même nom, et d'y accéder à travers un indice numérique. Sous Python on dispose d'une telle structure appelée **liste** (dans d'autres langages on dira **tableau**).

La liste vide s'écrit [ ].

On peut affecter une liste à une variable `L = [22,35,56,29]` et ainsi récupérer une valeur seule par son indice, déterminer la longueur de la liste, ajouter une valeur...

```
L=[22,35,56,29]
L[1]
35
L[3]
29
L[0]
22
L[-1]
29
L[-3]
35
L[4]
IndexError : list index out of range

len(L)
4

L.append(44)
L
[22,35,56,29,44]
L.insert(3,64)
L
[22,35,56,64,29,44]
del(L[3])
L
[22,35,56,29,44]
L.index(56)
2
L.remove(44)
L
[22,35,56,29]
L.reverse()
L
[29,56,35,22]
L.sort()
L
[22,29,35,56]

a=[1,2,3]
b=[2,2,1]
c=a+b
c
[1,2,3,2,2,1]
3 in b
False
3 in c
True
```

Voici les instructions éventuellement utiles dans ce TP :

`len(L)`	donne la longueur de `L`
`L=[]`	`L` est la liste vide
`L[i]`	donne la valeur de l'élément de la liste d'indice `i`
`L+M`	concatène (juxtapose) les listes `L` et `M`
`L.append(a)`	ajoute `a` à la fin de la liste
`L.insert(i,a)`	ajoute l'élément `a` à l'indice `i`
`del(L[i])`	supprime l'élément d'indice `i`
`L.remove(a)`	supprime l'élément `a` s'il est présent dans la liste
`L.index(a)`	donne l'indice de l'élément `a`
`for k in range(i,j)`	boucle **pour** `k` allant **de** `i` (inclus) **à** `j` (exclus)
`while`	boucle **tant que**

Le **tableau unidimensionnel** ou **liste** a l'avantage de regrouper un grand nombre de valeurs en une seule variable. Chaque valeur est alors est alors repérée par un numéro appelé **indice**.

Par exemple, dans une liste `L` comportant 12 valeurs, la première valeur sera désignée par `L[0]`, la deuxième par `L[1]`,..., la dernière par `L[11]` ou `L[-1]`.

1. **Recherches dans une liste**

   (a) Quel est le rôle de la fonction suivante ?

```
def fonction1(L): #L est une liste de nombres
    a=L[0]
    for k in range (len(L)):
        if L[k]>=a:
            a=L[k]
    return a
```

   (b) Modifier la fonction précédente pour obtenir le minimum de la liste L.

   (c) Dorénavant on suppose que L est une liste contenant des nombres deux à deux distincts.
   Ecrire une fonction `MAXmax(L:list)->tuple` qui renvoie le maximum et le second maximum de la liste L. Par exemple : `MAXmax([2,6,7,1,9,4]→ (9,7)`

2. **Calcul de la moyenne arithmétique**

   (a) Quel est le rôle des programmes suivants ?

```
def fonction2(n):
    '''
    * entrée : int
    * sortie : list    '''
    L=[]
    for k in range(n):
        L.append(k)
    return L
```

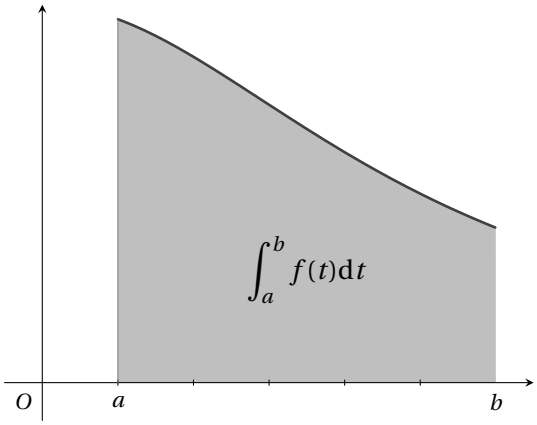
```
def fonction3(n:int)->int:
    S=0
    for k in range(n):
        S=S+k
    return S
```

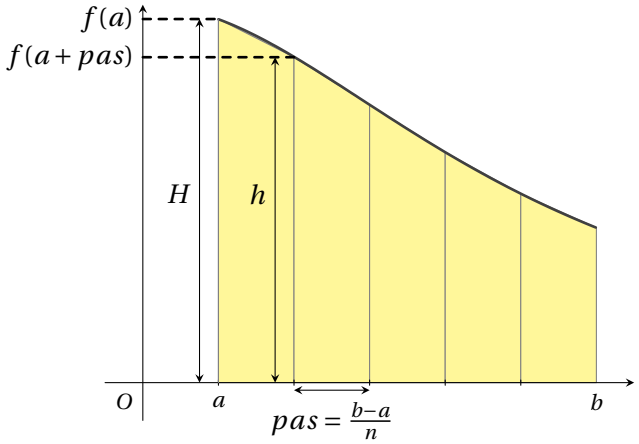
   (b) Soit L une liste contenant des nombres. Ecrire une fonction `moyenne(L:list)-> float` qui renvoie la moyenne des valeurs de la liste L.

3. **Calcul de la valeur moyenne par la méthode des trapèzes**

Méthode des trapèzes

Par définition, l'intégrale d'une fonction continue et positive sur $[a,b]$ est l'aire (grise ci-dessous) de la surface délimitée par la courbe, l'axe des abscisses et le droites verticales d'équation $x=a$ et $x=b$. La **méthode des trapèzes** consiste à approcher l'aire sous la courbe par la somme des aires de trapèzes.
Pour cela on définit une **subdivision** à pas constant :
Soit $n \in \mathbb{N}^*$, on définit le **pas** : $pas = \frac{b-a}{n}$ et $a_k = a + k \times pas$





L'aire du premier trapèze jaune est obtenue par la formule $\frac{H+h}{2} \times pas$, où $H = f(a)$ et $h = f(a+pas)$.
La formule qui donne une approximation de l'intégrale par la somme d'aires de trapèzes est donc :

$$\sum_{k=0}^{n-1} \frac{f(a_k)+f(a_{k+1})}{2} \times pas$$

(a) Ecrire une fonction `f(x)` qui pour la variable x retourne la valeur $\frac{4}{1+x^2}$.

(b) Ecrire une fonction `trapezes(f,a,b,n)` qui prend en argument une fonction f, deux réels a et b, et un entier n et qui renvoie la valeur approchée de l'intégrale obtenue par la méthode des trapèzes.

(c) Quelle valeur approchée obtient-on avec l'appel `trapezes(f,0,1,1000)` ?

Calcul de la valeur moyenne

On effectue une série de mesures sur une certaine quantité Q, durant une période donnée T.
Ces mesures sont stockées dans une liste `mesures`.
La valeur moyenne de ces mesures est définie par :

$$Q_{moy} = \frac{1}{T}\int_0^T Q(t)\mathrm{d}t$$

Définir une fonction `valeur_moyenne(mesures,T)`, prenant en arguments la liste `mesures` et la période T et qui calcule la valeur moyenne de ces mesures obtenue par la méthode des trapèzes.

Pour comparer les deux calculs de moyenne, on prend `L=[k for k in range(1,101)]` ; comparer les appels `moyenne(L)` et `valeur_moyenne(L,100)`.