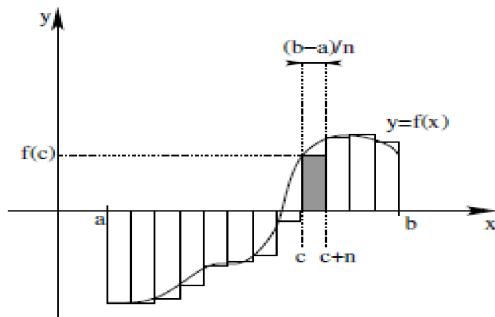


## Exercice 1 :

On se donne  $a < b$  deux réels, une fonction  $f$  sur  $[a, b]$  et  $n$  un entier non nul. On découpe  $[a, b]$  en  $n$  intervalles de même longueur. Le découpage définit une suite  $x_0=a, x_1, \dots, x_n=b$  de  $n+1$  points. On considère les  $n$  pavés du plan de base  $x_{k+1}-x_k$  de hauteur  $f(x_k)$ , pour  $k \in [0..n-1]$ .

La somme des surfaces de ces pavés est une approximation de  $\int_a^b f(x)dx$



Écrire un programme python qui détermine le réel qui approche l'intégrale de  $f$  sur l'intervalle  $[a, b]$ .

On prend  $f(x) = x^2 + \frac{1}{x}$

## Exercice 2 :

1. Donner trois méthodes différentes permettant de créer le tuple T tel que  $T= (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$   
NB : l'instruction  $T= (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$  n'est pas considérée
2. Soit le tuple T suivant :  $T= ([1, 2, 3], [4, 5, 6])$ . Parmi les instructions ci-dessous, dites si l'instruction est correcte ou non.
  - (a)  $T[0] = 'A'$
  - (b)  $T[0][0] = 'A'$
  - (c)  $T[1].append(25)$
  - (d)  $T[1][1].append(25)$
  - (e)  $T.len()$
  - (f)  $T[len(T)][0] = 12$
  - (g)  $T[len(T)-1] = 12$
  - (h)  $T[-1][-1] = 12$
3. Donner le code PYTHON permettant de créer une liste L composée de  $n$  entiers distincts choisis aléatoirement entre 1 et 500.
4. Soit  $c1$  une chaîne de caractères. Donner le code PYTHON permettant de créer une nouvelle chaîne de caractères  $c2$  à partir de  $c1$  telle qu'on insère le caractère ":" entre les blocs de 3 caractères de  $c1$  (sauf le dernier bloc)

Exemples : si  $c1 = "123456789"$ , alors  $c2 = "123 : 456 : 789"$   
Si  $c1 = "123456"$ , alors  $c2 = "123 : 456"$   
Si  $c1 = "1234567"$ , alors  $c2 = "123 : 456 : 7"$

5. Donner le code PYTHON permettant de créer la liste L à partir de c2 de la question précédente telle que L est composée de chaînes de caractères ou chaque chaîne est construite à partir d'un bloc de c2 et où chaque caractère du bloc est répété deux fois.

Exemples : si  $c2 = "123 : 456 : 789"$ , alors  $L = ["112233", "445566", "778899"]$

Si  $c2 = "123 : 456"$ , alors  $L = ["112233", "445566"]$

Si  $c2 = "123 : 456 : 7"$ , alors  $L = ["112233", "445566", "77"]$

Exercice 3 :

Les nombres **Zigzags** sont les nombres tels que leurs chiffres sont **alternativement** plus grands et plus petits que le précédent.

Exemples :

- ✓ Le nombre 1234567 est un nombre zigzag puisque ses chiffres vérifient les inégalités :  $1 < 3 > 2 < 5 > 4 < 7 > 6$ .
- ✓ Le nombre 745231 est un nombre zigzag puisque ses chiffres vérifient les inégalités :  $7 > 4 < 5 > 2 < 3 > 1$ .
- ✓ Le nombre 1325674 n'est pas un nombre zigzag puisque ses chiffres 5 et 6 ne vérifient pas les inégalités imposées :  $1 < 3 > 2 < 5 < 6 < 7 > 4$ .

1. Ecrire une fonction PYTHON intitulée **Chiffres(N)** qui retourne une liste L contenant les chiffres du nombre N donné en paramètre.  
Toute succession de chiffres identiques dans le nombre N, sera remplacée par un seul chiffre.  
Attention : les chiffres doivent garder le même ordre d'apparition que dans le nombre initial N.  
Exemple : pour  $N=155454$  la fonction retourne la liste  $L= [1, 5, 4, 5, 4]$
2. Ecrire une fonction PYTHON intitulée **Convert (L)** qui pour une liste L donnée retourne un nombre obtenu à partir des chiffres de L en gardant le même ordre des chiffres dans la liste.  
Exemple : pour  $L = [1, 5, 4, 5, 4]$  la fonction doit retourner le nombre 15454
3. Ecrire une fonction PYTHON intitulée **Saisie ()** qui saisit au clavier et retourne un nombre positif composé d'au minimum 3 chiffres et ne contenant pas de chiffres successifs identiques.

4. Ecrire une fonction PYTHON intitulée **Zigzag (N)** qui permet de vérifier si un nombre N est un nombre zigzag ou non.
5. Ecrire un programme PYTHON qui permet de :
  - ¶ Saisir un entier N composé d'au minimum 3 chiffres et ne contenant pas de chiffres successifs identiques.
  - ¶ Afficher si le nombre N est un nombre zigzag ou non