



**FACULTÉ des Sciences et  
Techniques de Tanger**  
كلية العلوم و التقنيات بطنجة

**Département Génie Informatique**  
**LST GI S5**  
**Algorithmique et POO C++**

Réalisé par : souhaila laazibi

CNE :P133375914

Encadré par M<sup>me</sup> : Ikram Ben Abdel Ouahab

## C'est quoi le python :

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels, Python est multiplateforme et open source. Python est un langage polyvalent et multiplateforme.

## Pourquoi le python :

Python est facile à apprendre, un langage de choix, c'est-à-dire à usage général : un langage de programmation interprété, interactif, orienté objet et de haut niveau à usage général.

## Le but de ce travail :

Se familiariser avec le langage python, développer notre compétence en programmation. On a utilisé comme PyCharm comme logiciel pour interpréter des résultats.

## Atelier 1 Python

- **Exercice 1 :** la somme des séries  $1! / 1 + 2! / 2 + 3! / 3 + 4! / 4 + 5! / 5$  en utilisant la fonction.

```
def factorial(x):  
    if x == 0:  
        return 1  
    else:  
        return(x*factorial(x-1))  
  
somme = 0  
n = int(input("entrer n :"))  
for i in range(1, n+1):  
    somme += factorial(i)/i  
print("la somme de série est :", somme)
```

Dans ce programme on a utilisé la récursivité avec la fonction factorial(n)

Puis on a parcouru l'ensemble de nombre à l'aide de la boucle for

```
entrer n :5  
la somme de série est : 34.0  
  
Process finished with exit code 0
```

- **Exercice 2 :** convertir un nombre décimal en nombre binaire en utilisant une fonction.

```
def conv(n):
    if n == 1:
        return 1 # si le nmb egale a 1
    else:
        conv(n//2)
        rest = n % 2 # si le nombre est different de 1 il va fire une division sur 2 et enregistrer le rest des 1 ou des 0 et
        print(rest, end='')

n = int(input("veuillez entrer un nombre"))
conv(n)
```

La fonction conv(n) va faire des divisions entières successives par 2 jusqu'à le quotient devient nul et afficher l'ensemble des 1 et des 0 a l'aide de print().

```
veuillez entrer un nombre10
010
Process finished with exit code 0
```

- **Exercice 3 :** Ecrire un programme en Python pour calculer la somme des nombres de 1 à n en utilisant la récursivité.

```
n = int(input("veuillez entrer un nombre:"))
def somme(n):
    if n == 1:
        return 1
    else:
        return(n+somme(n-1))#récursivité
print(somme(n)) #afficher le resultat
somme(n) #appelé la fonction
```

L'utilisation de la Récursivité avec la fonction somme(n)

Le resultat est comme l'exemple 1+2+3=6

```
veuillez entrer un nombre10
010
Process finished with exit code 0
```

- **Exercice 4 :** calcule de la série Fibonacci en utilisant la récursivité.

```

#le série de fibonacci :
def fibonacci(n):
    if n==0:
        return 0
    elif n==1:
        return 1
    else:
        return fibonacci(n-1)+fibonacci(n-2)
x=int(input("enter un nombre :"))
#afficher le série de fibonacci:
print("le série de fibonacci de",x,"est",fibonacci(x))

```

Fibonacci est une suite d'entiers dans laquelle chaque terme est la somme des deux termes qui le précèdent.

On la définit par : si  $n=0 \Rightarrow F_n=0$

Sinon si  $n=1 \Rightarrow 1$

Si  $n \neq 1$  et  $0$  alors :  $F_n=f(n-1)+F_n(n-2)$

Comme on voit dans le résultat :

```

enter un nombre :12
le série de fibonacci de 12 est 144

Process finished with exit code 0

```

- **Exercice 5 :** un programme en Python pour compter les chiffres d'un nombre donné en utilisant la récursivité.

```

def longueur(ch):
    if not ch:
        return 0 #si on tape ch=""
    else:
        return 1+longueur(ch[1:]) #[1:] est utilisé pour afficher la longueur de tout le rest de le liste

ch = "20211912" #le chiffre
print("la longueur du chaîne est : ",longueur(ch))

```

A l'aide de la fonction longueur : on a calculer le nombre des chiffre d'un caractere

```
la longueur du chaine est : 8  
Process finished with exit code 0
```

- **Exercice 6 :**

- Tri\_bulle :

```
def tri_bulle(tab):  
    n = len(tab)  
    for i in range(n):  
        for j in range(0, n-i-1):  
            if tab[j] > tab[j+1]:  
                tab[j], tab[j+1] = tab[j+1], tab[j]  
tab = [4, 5, 3, 6, 8, 0, 1, 2]  
tri_bulle(tab)  
print("le tableau trié en bulle est :")  
for i in range(len(tab)):  
    print("%d" % tab[i], " ", end='')  
    print("\n", end='')  
    tab = tri_bulle(tab)
```

Le tableau est parcouru du premier au dernier élément. l'élément courant est comparé à l'élément suivant. Si l'élément en cours est supérieur à l'élément suivant, il est échangé

```
le tableau trié en bulle est :  
0 1 2 3 4 5 6 8  
Process finished with exit code 0
```

- Tri par selection :

```
def selection(tab):  
    for i in range(len(tab)):  
        #trouver le min  
        min = i  
        for j in range(i + 1, len(tab)):  
            if tab[min] > tab[j]:  
                min = j  
            mintab = tab[i]  
            tab[i] = tab[min]  
            tab[min] = mintab  
        return tab  
tab = [2, 1, 19, 8, 12, 23, 11, 14]  
selection(tab)  
#pour afficher la liste trié !  
print("Le tableau trié par sélection est:")  
for i in range(len(tab)):  
    print("%d" % tab[i], " ", end='')  
    print("\n", end='')  
    tab = selection(tab)
```



- **Exercice 8 :**

```
#trouver la fréquence d'un caractère dans une chaîne:
from collections import Counter
str='hello python'
frequence=Counter(str)
for (key,value) in frequence.items(): #
    print("le nombre de fréquence de " ,key,"est:" ,value)
```

```
le nombre de fréquence de h est: 2
le nombre de fréquence de e est: 1
le nombre de fréquence de l est: 2
le nombre de fréquence de o est: 2
le nombre de fréquence de   est: 1
le nombre de fréquence de p est: 1
le nombre de fréquence de y est: 1
le nombre de fréquence de t est: 1
le nombre de fréquence de n est: 1
```

- **Exercice 9 :**

```
A = input("entrer A :")
matrice = [3, 5, 6], [9, 7, 3]
def find(n):
    for i in range(len(matrice)):
        for j in range(len(matrice)+1):
            if matrice[i][j] == n:
                return i,j
print(find(A))
```

## Atelier 2

- **Exercice 1:**

```
l1 = [2, 5, 7, 8, 16, 20]
l2 = [3, 4, 6, 10, 12, 14]
for indice, element in enumerate(l1):#parcourir tout les elements
    if indice % 2 != 0:
        l11 = element
        print(l11)
for indice, element in enumerate(l2):#parcourir tout les elements
    if indice % 2 == 0:
        l22 = element
        print(l22)
l3 = [x for x in l11 if x in l22]
print(l3)
```

Ce programme va choisir des éléments d'index impair dans L1 et des éléments d'index pair dans L2. Étant donné deux listes l1 et l2, à l'aide de la boucle for il va créer une intersection des deux listes dans L3

```
5
8
20
3
```

- **Exercice 2 :**

```
lis = [17, 12, 19, 3, 5, 7, 10, 45, 2]
print("notre liste:", lis)
n = 3
output = [lis[i:i + n] for i in range(0, len(lis), n)]
print("liste divisée:", output)
output.reverse()
print("liste inversé", output)
```

Diviser la liste en 3 morceaux égaux et inverser chaque morceau

```
notre liste: [17, 12, 19, 3, 5, 7, 10, 45, 2]
liste divisée: [[17, 12, 19], [3, 5, 7], [10, 45, 2]]
liste inversé [[10, 45, 2], [3, 5, 7], [17, 12, 19]]

Process finished with exit code 0
```

- **Exercice 3 :**

```
L=[12,4,56,7,8,92,2,3,2,4,3]
for n in L: #parcourir la liste
    print(n, " ", end='') #affichage de tout les elements de la liste
from collections import Counter
Counter(L)
print("les occurrences de chaque element :", Counter(L))
```

itérer une liste donnée et compter l'occurrence de chaque élément et créer un dictionnaire pour montrer le nombre de chaque élément comme suivant :

```
12 4 56 7 8 92 2 3 2 4 3 les occurrences de chaque element ; Counter({4: 2, 2: 2, 3: 2, 12: 1, 56: 1, 7: 1, 8: 1, 92: 1})
```



- **Exercice 4 :**

```
liste1 = [1, 2, [12, 5], [6, 7], 9, 10]
liste2 = [2, [6, 7], 15, 17]
intersection2 = [x for x in liste2 if x in liste1]#selectionné les elements des 2 liste
print(intersection2)
for x in liste2:
    if x in liste1:
        liste1.remove(x)#supprimer les elements de l'intersection
print(liste1)
```

Trouver l'intersection (commune) al'aide de la boucle for de deux liste et supprimez ces éléments du premier liste

```
[2, [6, 7]]
[1, [12, 5], 9, 10]

Process finished with exit code 0
```

- **Exercice 5 :**

itérer une liste donnée et vérifier si **un élément donné existe a l'aide de la boucle for** en tant que valeur de clé dans un dictionnaire.

```
L = [3, 24, 85, 19, 12, 14, 6]
dic = {24: "mohamed", 12: "asmae", 6: "khadija", 14: "lamiae"}
for x in dic:
    if x in L:
        L = [x for x in dic]
print(L)
```

```
[24, 12, 6, 14]
```