

COMPTE RENDU DE TP1

« PYTHON »

-Atelier 1 :

Ex1

o Solution du exercice :

```
#declaration de la fonction factorielle
def factorielle(x) :
    if x==0:
        return 1
    else:
        return x*factorielle(x - 1)

#demande de l'utilisateur pour saisir la valeur de n
n=int(input("entrer la valeur de n:"))

#initialisation du variable sum par 0
sum=0

#utilisation de boucle for afin de parcourir tout les elements
for i in range(1,n+1):
    sum = sum + factorielle(i)/i

#affichage de la somme final
print(sum)
```

o Code source + Commentaires

```

1  #EX 1
2  #declaration de la fonction factorielle
3  def factorielle(x) :
4      if x==0:
5          return 1
6      else:
7          return x*factorielle(x - 1)
8  #demande de l'utilisateur pour saisir la valeur de n
9  n=int(input("entrer la valeur de n:"))
10 #initialisation du variable sum par 0
11 sum=0
12 #utilisation de boucle for afin de parcourir tout les elements
13 for i in range(1,n+1):
14     sum = sum + factorielle(i)/i
15 #affichage de la somme final
16 print(sum)
17

```

⇒ Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code

o Sortie du programme + Commentaires :

```

Run: atl1 ex 8 x  ATELIER 1.PY x  atelier1 ex1 x
C:\Users\yousra\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/yousra/PycharmProjects/pythonProject/atelier1 ex1.py"
entrer la valeur de n:
1.0
2.0
4.0
10.0
34.0
154.0
Process finished with exit code 0

```

⇒ Cet exercice permet de calculer le factorielle des nbrs n puis calculer la somme de $n! / n$

EX2

o Solution du exercice :

#declaration de la fonction qui converti un nbr decimal en binaire

def decimal_to_binary(n):

binary = ""

#la condition d arret

while(n != 0) :

binary += str(n % 2)

n = n // 2

#pour inverser le code binaire

return binary[::-1]

n = int(input("entrer un nbr"))

#affichage de resultat

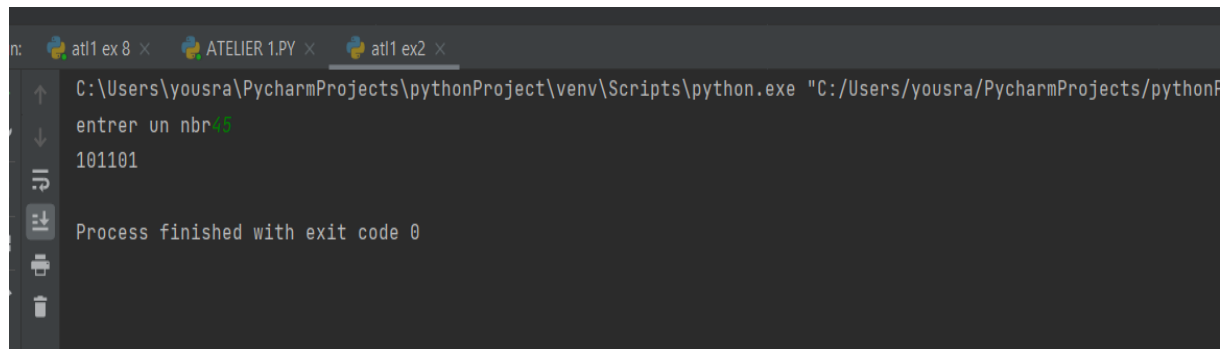
print(decimal_to_binary(n))

o Code source + Commentaires

```
1      #declaration de la fonction qui converti un nbr decimal en binaire
2      def decimal_to_binary(n):
3          binary = ""
4          #la condition d arret
5          while(n != 0 ) :
6              binary += str(n % 2)
7              n = n // 2
8          #pour inverser le code binaire
9          return binary[::-1]
10     n = int(input("entrer un nbr"))
11     #affichage de resultat
12     print(decimal_to_binary(n))
```

⇒ **Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code**

o Sortie du programme + Commentaires :



```
atl1 ex 8 x  ATELIER 1.PY x  atl1 ex2 x
C:\Users\yousra\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/yousra/PycharmProjects/pythonProject/atl1 ex2.py"
entrer un nbr
101101
Process finished with exit code 0
```

⇒ Cet exer consiste a convertir un nbr de base decimal en sa base binaire.

EX3

o Solution du exercice :

#declaration d une fct qui fait la somme en utilisant la recursivité

def somme(n):

 if n == 0:

 return 0

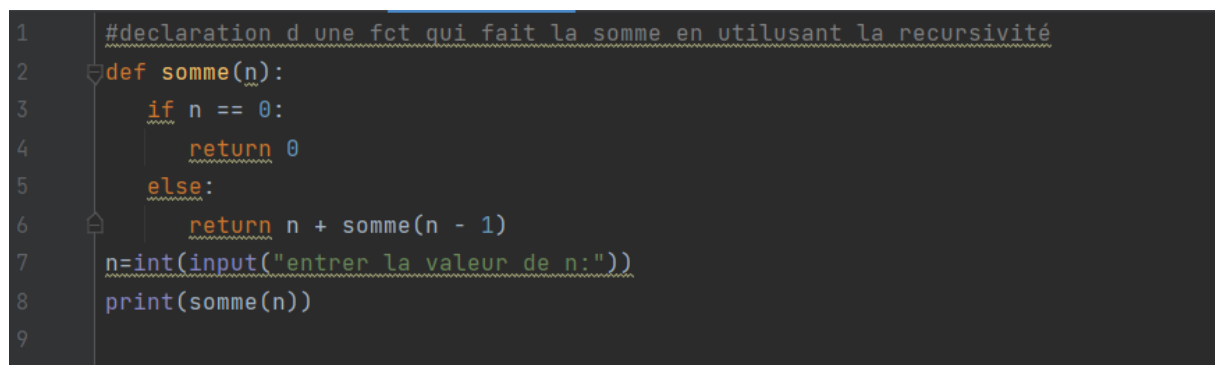
 else:

 return n + somme(n - 1)

n=int(input("entrer la valeur de n:"))

print(somme(n))

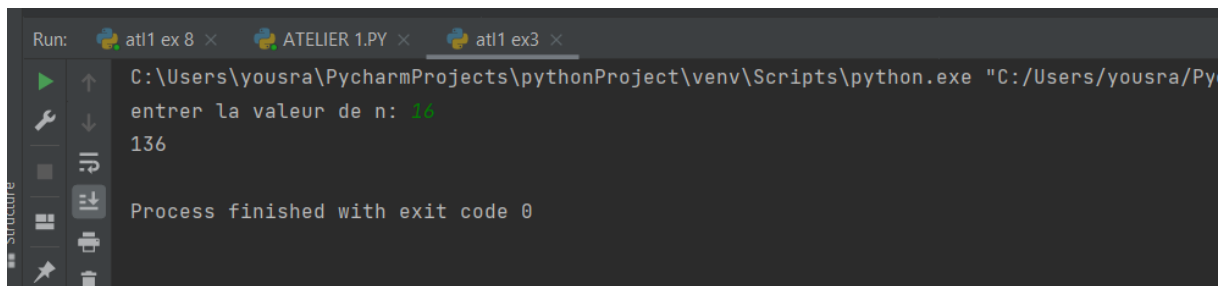
o Code source + Commentaires



```
1  #declaration d une fct qui fait la somme en utilisant la recursivité
2  def somme(n):
3      if n == 0:
4          return 0
5      else:
6          return n + somme(n - 1)
7  n=int(input("entrer la valeur de n:"))
8  print(somme(n))
9
```

⇒ Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code

o Sortie du programme + Commentaires :



```
Run: atl1 ex 8 x ATELIER 1.PY x atl1 ex 3 x
C:\Users\yousra\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/yousra/Py
entrer la valeur de n: 16
136
Process finished with exit code 0
```

⇒ Cet exr a pour but de calculer la somme des nbr de 1 à n (donnée par l'utilisateur)

EX4

o Solution du exercice :

#declaration de la fct

def fibonacci(n):

if(n <= 1):

return n

else:

return (fibonacci(n-1) + fibonacci(n-2))

#demande de l'utilisateur pour taper le nbr de termes voulu

n = int(input("Entrez le nombre de termes:"))

print("Suite de Fibonacci en utilisant la recursion :")

for i in range(n):

#affichage de resultat

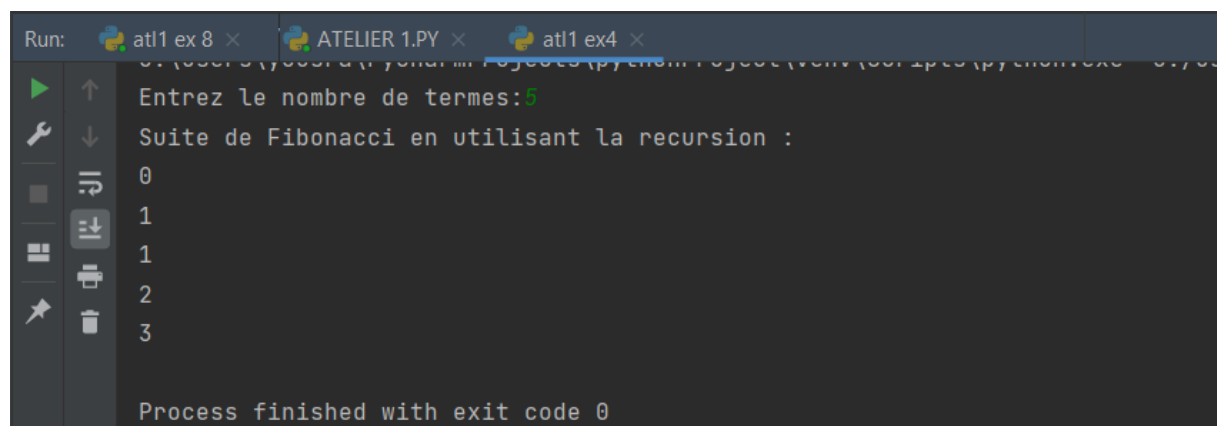
```
print(fibonacci(i))
```

o Code source + Commentaires

```
1  #declaration de la fct
2  def fibonacci(n):
3      if(n <= 1):
4          return n
5      else:
6          return (fibonacci(n-1) + fibonacci(n-2))
7  #demande de l'utilisateur pour taper le nbr de termes voulu
8  n = int(input("Entrez le nombre de termes:"))
9  print("Suite de Fibonacci en utilisant la recursion :")
10 for i in range(n):
11     #affichage de resultat
12     print(fibonacci(i))
13
14
```

⇒ **Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code**

o Sortie du programme + Commentaires :



```
Run: atl1 ex 8 x ATELIER 1.PY x atl1 ex4 x
Entrez le nombre de termes:5
Suite de Fibonacci en utilisant la recursion :
0
1
1
2
3
Process finished with exit code 0
```

⇒ Cet exr permet de imprimer la série de fibonacci.

EX5 :

o Solution du exercice :

#declaration d une fct qui compte les chiffres

def compte(N):

#declaration d un variable a stoké et l'initialiser par 0

nbr = 0

while N != 0:

N=int(N/10)

nbr += 1

return nbr

#demande de l utilisateur a taper le nbr voulu

N = int(input("saisir un nbr : "))

#affichage de resultat

print("le nbr total de chiffre dans le nombre ",N,"est",compte(N))

o Code source + Commentaires

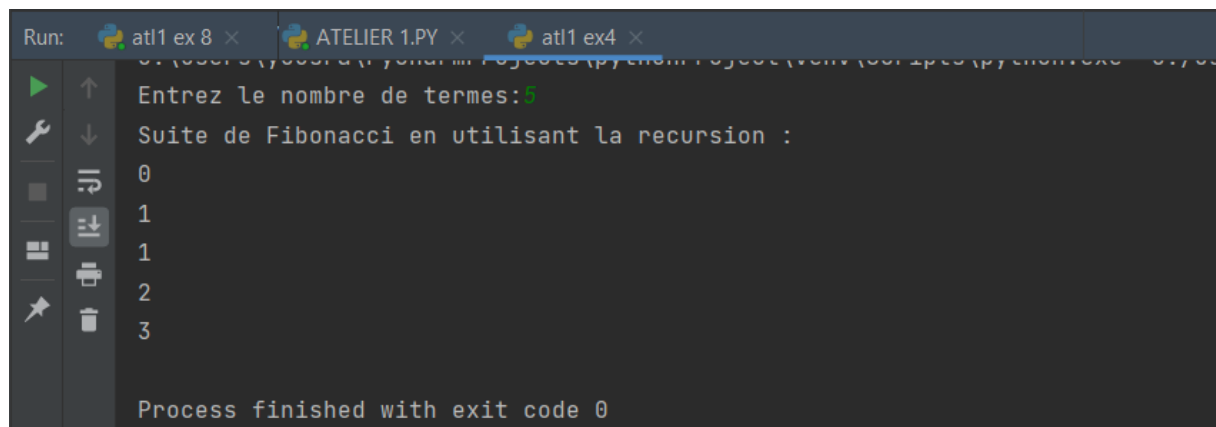
```

1  #declaration d une fct qui compte les chiffres
2  def compte(N):
3      #declaration d un variable a stoké et l'initialiser par 0
4      nbr = 0
5      while N != 0:
6          N=int(N/10)
7          nbr += 1
8      return nbr
9
10 #demande de l'utilisateur a taper le nbr voulu
11 N = int(input("saisir un nbr : "))
12 #affichage de resultat
13 print("le nbr total de chiffre dans le nombre ",N,"est",compte(N))

```

⇒ Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code

o Sortie du programme + Commentaires :



```

Run: atl1 ex 8 x ATELIER 1.PY x atl1 ex4 x
Entrez le nombre de termes: 5
Suite de Fibonacci en utilisant la recursion :
0
1
1
2
3
Process finished with exit code 0

```

EX6

o Solution du exercice :

#declaration de la premiere fct triante

def tri_bull(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]
    return tab
```

#declaration de la deusieme fct triante

```
def tri_selection(tab):
    for i in range (len(tab)):
        min = i
        for j in range (i+1 , len(tab)):
            if tab[min] > tab[j]:
                min = j
        tmp = tab[i]
        tab[i] = tab[min]
        tab[min] = tmp
    return tab
```

#declaration de la troisieme fct triante

```
def tri_insertion(tab):
    for i in range(1,len(tab)):
        k = tab[i]
        j = i - 1
        while j>=0 and k<tab[j]:
            tab[j+1] = tab[j]
            j=-1
```

```
        tab[j+1] = k
    return tab

#declaration du tableau a trier
tab = [60,22,35,9,100,230,514,24,66,10,15,67,669]

#affichage de resultat selon les algorithmes

#premiere resultat:
print("EN UTILISANT TRI A BULL VOICI LE RESULTAT :
",tri_bull(tab))

#seconde resultat:
print("EN UTILISANT TRI PAR SELECTION VOICI LE
RESULTAT: ",tri_selection(tab))

#tertiaire resultat
print("EN UTILISANT TRI PAR INSERTION VOICI LE
RESULTAT: ",tri_insertion(tab))

#et finalement on obtient le meme resultat
```

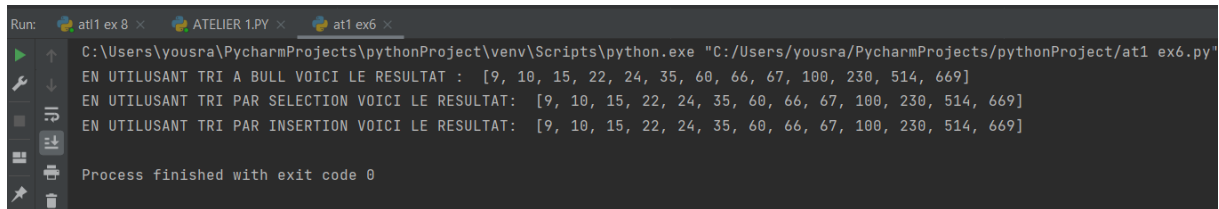
[o Code source + Commentaires](#)

```
Project
1      #declaration de la premiere fct triante
2      def tri_bull(tab):
3          n = len(tab)
4          for i in range(n):
5              for j in range(0, n-i-1):
6                  if tab[j] > tab[j+1]:
7                      tab[j], tab[j+1] = tab[j+1], tab[j]
8          return tab
9      #declaration de la deusieme fct triante
10     def tri_selection(tab):
11         for i in range(len(tab)):
12             min = i
13             for j in range(i+1, len(tab)):
14                 if tab[min] > tab[j]:
15                     min = j
16             tmp = tab[i]
17             tab[i] = tab[min]
18             tab[min] = tmp
19         return tab
20     #declaration de la troisieme fct triante
21     def tri_insertion(tab):
22         for i in range(1, len(tab)):
23             k = tab[i]
24             j = i - 1
25             while j >= 0 and k < tab[j]:
26                 tab[j+1] = tab[j]
27                 j = j - 1
28             tab[j+1] = k
29         return tab
30     #declaration du tableau a trier
```

```
29         return tab
30     #declaration du tableau a trier
31     tab = [60, 22, 35, 9, 100, 230, 514, 24, 66, 10, 15, 67, 669]
32     #affichage de resultat selon les algorithmes
33     #premiere resultat:
34     print("EN UTILISANT TRI A BULL VOICI LE RESULTAT : ", tri_bull(tab))
35     #seconde resultat:
36     print("EN UTILISANT TRI PAR SELECTION VOICI LE RESULTAT: ", tri_selection(tab))
37     #tertiaire resultat
38     print("EN UTILISANT TRI PAR INSERTION VOICI LE RESULTAT: ", tri_insertion(tab))
39     #et finalement on obtient le meme resultat
Favorites
```

⇒ Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code

o Sortie du programme + Commentaires :



```
Run: at1 ex 8 x ATELIER 1.PY x at1 ex 6 x
C:\Users\yousra\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/yousra/PycharmProjects/pythonProject/at1 ex6.py"
EN UTILISANT TRI A BULL VOICI LE RESULTAT : [9, 10, 15, 22, 24, 35, 60, 66, 67, 100, 230, 514, 669]
EN UTILISANT TRI PAR SELECTION VOICI LE RESULTAT: [9, 10, 15, 22, 24, 35, 60, 66, 67, 100, 230, 514, 669]
EN UTILISANT TRI PAR INSERTION VOICI LE RESULTAT: [9, 10, 15, 22, 24, 35, 60, 66, 67, 100, 230, 514, 669]
Process finished with exit code 0
```

Cet exer a pour but de trier un tableau donnée en utilisant les trois types de tri :(Insertion/Selection/Bull)

EX7

o Solution du exercice :

#declaration de la chaine a inverser

str="yousra"

string_reversed=[]

i = len(str)

while i > 0:

 string_reversed += str[i-1]

 i = i -1

#affichage du chaine inversée

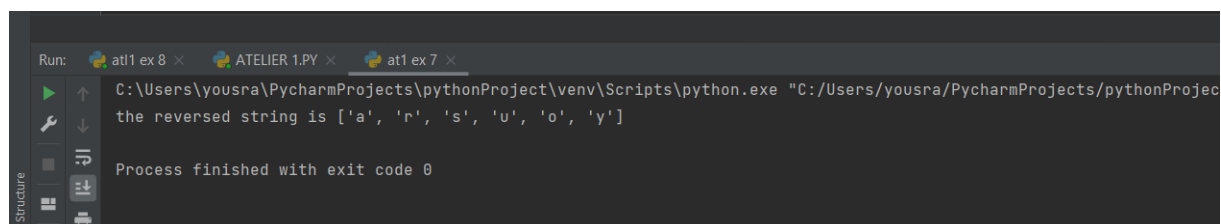
print("the reversed string is",string_reversed)

o Code source + Commentaires

```
1  #declaration de la chaine a inverser
2  str="yousra"
3  string_reversed=[]
4  i = len(str)
5  while i > 0:
6      string_reversed += str[i-1]
7      i = i - 1
8  #affichage du chaine inversée
9  print("the reversed string is",string_reversed)
10
```

⇒ Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code

o Sortie du programme + Commentaires :



```
Run: at1 ex 8 x ATELIER 1.PY x at1 ex 7 x
C:\Users\yousra\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/yousra/PycharmProjects/pythonProject/
the reversed string is ['a', 'r', 's', 'u', 'o', 'y']
Process finished with exit code 0
```

⇒ Cet exer contient une fct qui consiste à inverser les lettres de la chaine suivante « yousra »

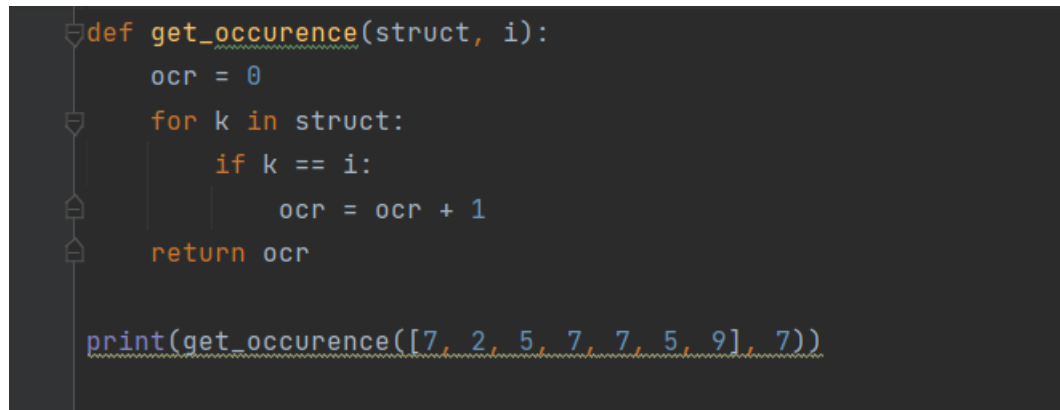
EX8

o Solution du exercice :

```
def get_occurence(struct, i):
    ocr = 0
    for k in struct:
        if k == i:
```

```
    ocr = ocr + 1  
    return ocr  
print(get_occurence([7, 2, 5, 7, 7, 5, 9], 7))
```

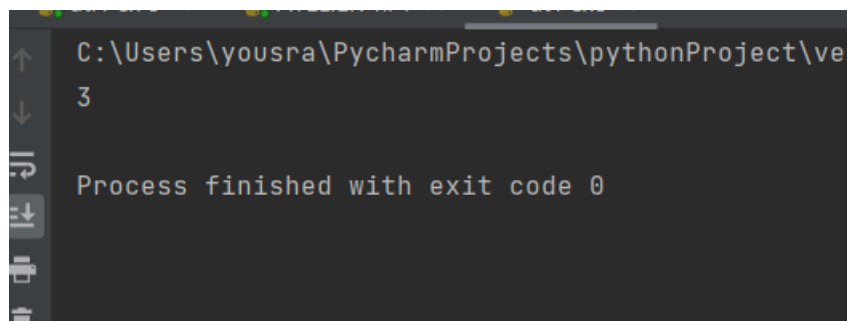
o Code source + Commentaires



```
def get_occurence(struct, i):  
    ocr = 0  
    for k in struct:  
        if k == i:  
            ocr = ocr + 1  
    return ocr  
  
print(get_occurence([7, 2, 5, 7, 7, 5, 9], 7))
```

⇒ Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code

o Sortie du programme + Commentaires :



```
C:\Users\yousra\PycharmProjects\pythonProject\ven  
3  
  
Process finished with exit code 0
```

⇒ Cet exer contient une fct qui permet de trouver la fréquence d un caractère dans une chaine

-Atelier 2:

EX1

o Solution du exercice :

#declaration de la fonction factorielle

def factorielle(x) :

if x==0:

return 1

else:

return x*factorielle(x - 1)

#demande de l'utilisateur pour saisir la valeur de n

n=int(input("entrer la valeur de n:"))

#initialisation du variable sum par 0

sum=0

#utilisation de boucle for afin de parcourir tout les elements

for i in range(1,n+1):

sum = sum + factorielle(i)/i

#affichage de la somme final

print(sum)

o Code source + Commentaires

```
#declaration de la fonction factorielle
def factorielle(x) :
    if x==0:
        return 1
    else:
        return x*factorielle(x - 1)
#demande de l'utilisateur pour saisir la valeur de n
n=int(input("entrer la valeur de n:"))
#initialisation du variable sum par 0
sum=0
#utilisation de boucle for afin de parcourir tout les elements
for i in range(1,n+1):
    sum = sum + factorielle(i)/i
#affichage de la somme final
print(sum)
```

⇒ Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code

o Sortie du programme + Commentaires :

```
C:\Users\yousra\PycharmProjects\pythonProject\venv\Scripts\python.exe
[2, 4, 1, 3]
Process finished with exit code 0
```

➔ **Cet exer a pour but** de Créer une liste en choisissant des éléments d'index impair dans la première liste et des éléments d'index pair dans la seconde. Étant donné deux

listes, l1 et l2, écrivez un programme pour créer une troisième liste l3 en choisissant un élément d'indice impair dans la liste l1 et des éléments d'indice pair dans la liste l2.

EX2

o Solution du exercice :

#declaration de la fct

```
def divide_conquere(data):
```

```
    i = 0
```

```
    export = []
```

```
    for i in range(0, 9, 3):
```

```
        export.append(data[i:i + 3])
```

```
    return export
```

```
print(divide_conquere([0, 8, 5, 6, 6, 4, 7, 3, 9]))
```

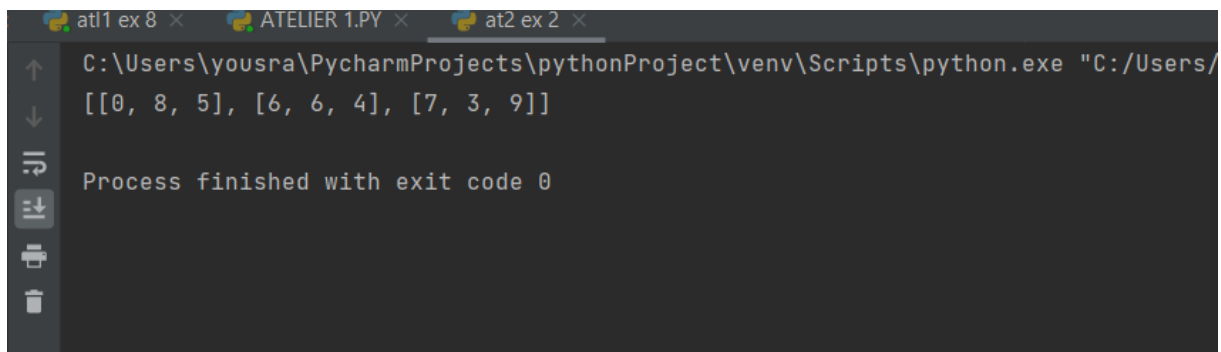
o Code source + Commentaires

```
#declaration de la fct
def divide_conquere(data):
    i = 0
    export = []
    for i in range(0, 9, 3):
        export.append(data[i:i + 3])
    return export

print(divide_conquere([0, 8, 5, 6, 6, 4, 7, 3, 9]))
```

⇒ Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code

o Sortie du programme + Commentaires :



```
atl1 ex 8 x ATELIER 1.PY x at2 ex 2 x
C:\Users\yousra\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/
[[0, 8, 5], [6, 6, 4], [7, 3, 9]]
Process finished with exit code 0
```

Cet exer a pour bute de Deviser la liste en 3 morceaux égaux et inverser chaque morceau

EX3

o Solution du exercice :

#declaration de la fct occurence

def occurence(list):

 a = {}

 for j in list:

 if str(j) in a:

 a[str(j)] = a.get(str(j)) + 1

 else:

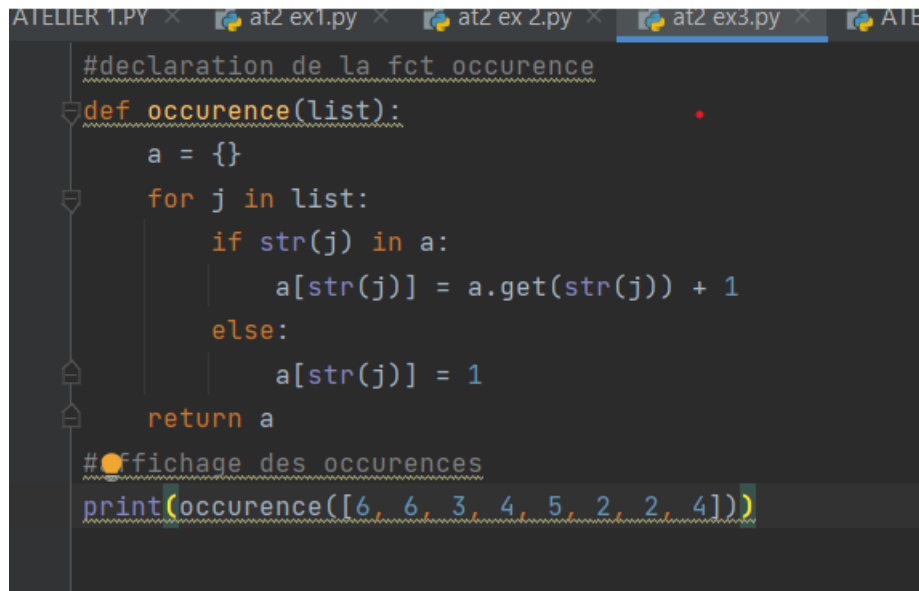
 a[str(j)] = 1

 return a

#affichage des occurences

```
print(occurence([6, 6, 3, 4, 5, 2, 2, 4]))
```

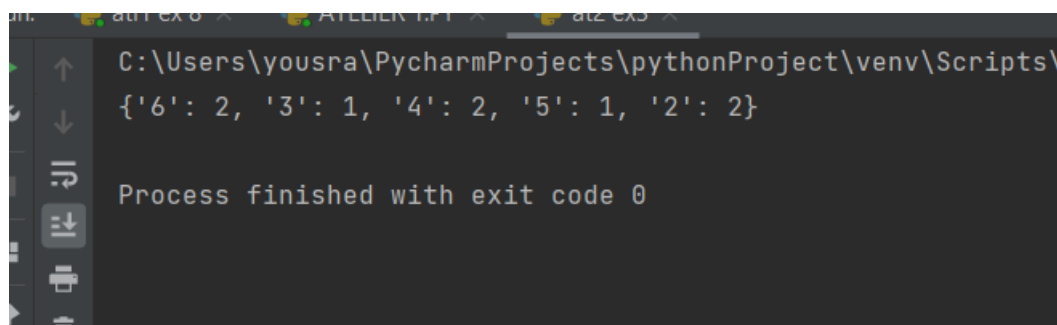
o Code source + Commentaires



```
#declaration de la fct occurence
def occurence(list):
    a = {}
    for j in list:
        if str(j) in a:
            a[str(j)] = a.get(str(j)) + 1
        else:
            a[str(j)] = 1
    return a
#affichage des occurrences
print(occurence([6, 6, 3, 4, 5, 2, 2, 4]))
```

⇒ Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code

o Sortie du programme + Commentaires :



```
C:\Users\yousra\PycharmProjects\pythonProject\venv\Scripts\
{'6': 2, '3': 1, '4': 2, '5': 1, '2': 2}

Process finished with exit code 0
```

⇒ **Cet un programme qui permet d'itérer une liste donnée et compter l'occurrence de chaque élément et créer un dictionnaire pour montrer le nombre de chaque élément.**

EX4

o Solution du exercice :

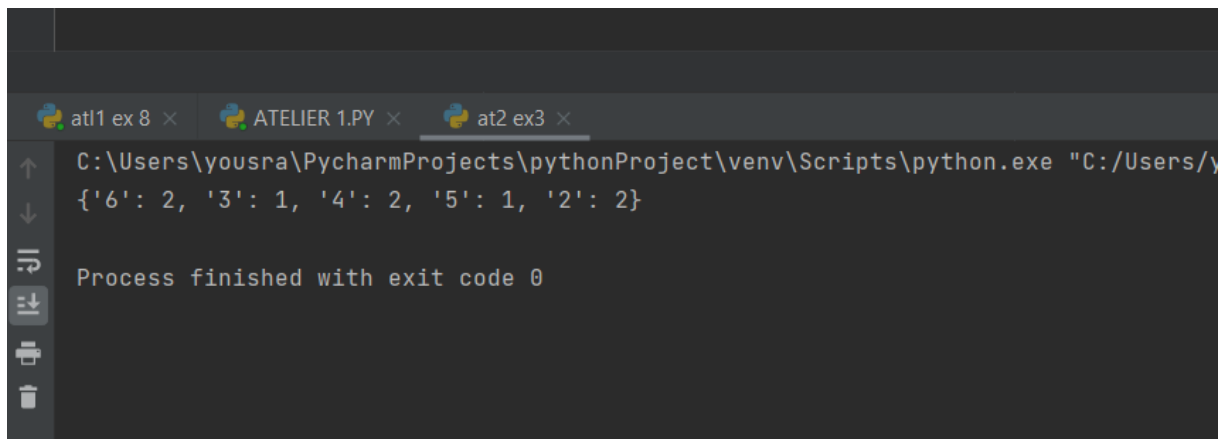
```
def difference(set1, set2):  
    a = set(("  
    for i in set1:  
        for j in set2:  
            if i == j: a.add(i)  
    b = set1.copy()  
    for i in set1:  
        if i in a: b.remove(i)  
    print("this is the difference => ", a, "this is the remainder of set1", b)  
print(difference({23, 42, 65, 57, 78, 83, 29}, {57, 83, 29, 67, 73, 43, 48}))
```

o Code source + Commentaires

```
def difference(set1, set2):  
    a = set(("  
    for i in set1:  
        for j in set2:  
            if i == j: a.add(i)  
    b = set1.copy()  
    for i in set1:  
        if i in a: b.remove(i)  
  
    print("this is the difference => ", a, "this is the remainder of set1", b)  
  
print(difference({23, 42, 65, 57, 78, 83, 29}, {57, 83, 29, 67, 73, 43, 48}))
```

⇒ Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code

o Sortie du programme + Commentaires :



```
at1 ex 8 x  ATELIER 1.PY x  at2 ex3 x
C:\Users\yousra\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/y
{'6': 2, '3': 1, '4': 2, '5': 1, '2': 2}
Process finished with exit code 0
```

⇒ **Ce programme consiste à** Trouver l'intersection (commune) de deux Sets et supprimez ces éléments du premier Set

EX5

o Solution du exercice :

#declaration du fct

def get(list1,dict1):

#affectation des elements clés au dictionnair

keys = list(dict1.keys())

result = []

**#utiliser la boucle for pour verifier l existances des éléments
assemblés**

for i in keys:

for j in list1:

```

        if dict1[str(i)] == j :
            result.append(j)

    return result

#declaration de la liste et du dictionnaire en appelant la fct get
print(get([47,64,69,37,76,83,95,97],
{'Yassine':47, 'Imane':69, 'Mohammed':76, 'Abir':97}))

```

o Code source + Commentaires

```

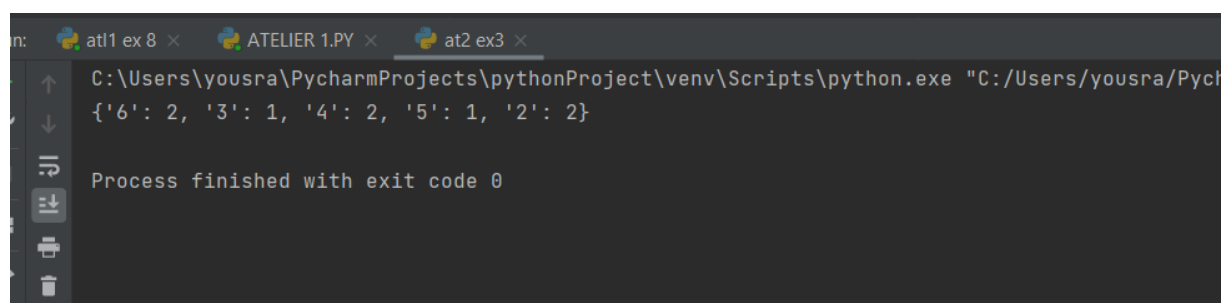
#declaration du fct
def get(list1,dict1):
    #affectation des elements clés au dictionnaire
    keys = list(dict1.keys())
    result = []
    #utiliser la boucle for pour verifier l existences des éléments assemblés
    for i in keys:
        for j in list1:
            if dict1[str(i)] == j:
                result.append(j)
    return result

#declaration de la liste et du dictionnaire en appelant la fct get
print(get([47,64,69,37,76,83,95,97],
{'Yassine':47, 'Imane':69, 'Mohammed':76, 'Abir':97}))

```

⇒ Les commentaires de chaque partie sont déclarée au dessus tout au long de la programmation du code

o Sortie du programme + Commentaires :



```

C:\Users\yousra\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/yousra/Pyct
{'6': 2, '3': 1, '4': 2, '5': 1, '2': 2}

Process finished with exit code 0

```

- ⇒ Ce programme a pour but d'itérer une liste donnée et vérifier si un élément donné existe en tant que valeur de clé dans un dictionnaire. Sinon, supprimez-le de la liste.

Préparé par : Yousra Msaouri Charroud

Final