



# ***TP PYTHON***

## **COMPTE RENDU**

ENCADRE PAR :

**Pr. Lotfi ELAACHAK**

**Mme. Ikramben abdel ouahab**

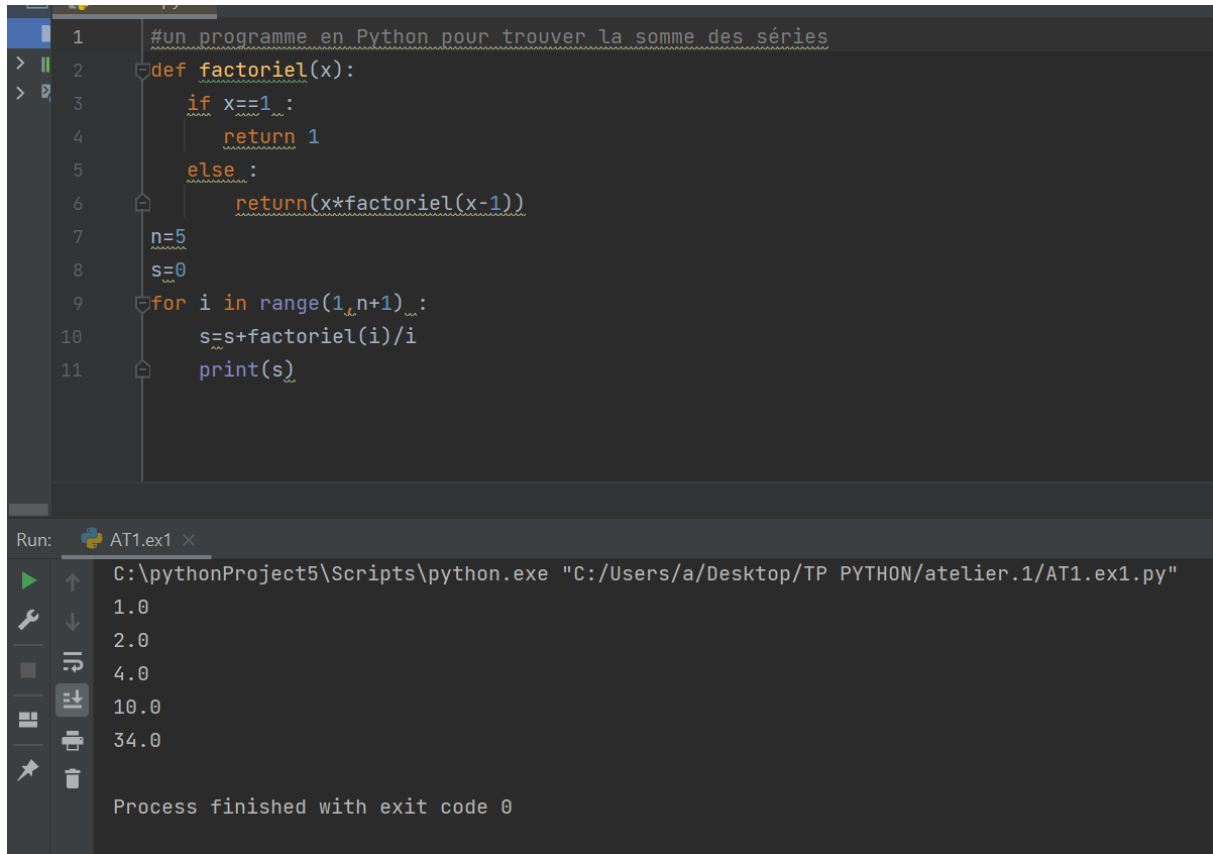
Réalisé par :

**Annaimi yassine**

**grp :02**

# ATELIER 1

## Exercice1 ATELIER 1 :



```
1 #un programme en Python pour trouver la somme des séries
2 def factoriel(x):
3     if x==1:
4         return 1
5     else:
6         return(x*factoriel(x-1))
7 n=5
8 s=0
9 for i in range(1,n+1):
10     s=s+factoriel(i)/i
11 print(s)
```

Run: AT1.ex1 x

C:\pythonProject5\Scripts\python.exe "C:/Users/a/Desktop/TP PYTHON/atelier.1/AT1.ex1.py"

1.0  
2.0  
4.0  
10.0  
34.0

Process finished with exit code 0

## Code source exercice 1:

```
#un programme en Python pour trouver la somme des séries
def factoriel(x):
    if x==1:
        return 1
    else:
        return(x*factoriel(x-1))
n=5
s=0
for i in range(1,n+1):
    s=s+factoriel(i)/i
print(s)
```

## Exercice 2 atelier 1 :

```
1  #un programme convert les nombres de types a vers type b (decimal >> binaire )
2  def convert(x):
3      if x==0:
4          return 0
5      else:
6          return (x%2)+(10*convert(x//2))
7  n = int(input("saisir le nbr : "))
8  print(convert(n))
9
```

AT1.ex2 x

C:\Python\Python37\python.exe "C:/Users/a/Desktop/TP PYTHON/atelier.1/AT1.ex2.py"

saisir le nbr : 23

10111

Process finished with exit code 0

## Code source exercice 2 :

```
#un programme convert les nombres de types a vers type b (decimal >>
binaire )
def convert(x):
    if x==0:
        return 0
    else:
        return (x%2)+(10*convert(x//2))
n = int(input("saisir le nbr : "))
print(convert(n))
```

## Exercice 3 atelier 1 :

```
1 def somme(n):
2     if n==0:
3         return 0
4     else:
5         return n + somme(n-1)
6 n = int(input("saisir le nbr de sommet : "))
7 print(somme(n))
8 # un script qui calcule la somme par la récursivité
9
```

somme() > if n==0

AT1.ex3 x

C:\Python\Python37\python.exe "C:/Users/a/Desktop/TP PYTHON/atelier.1/AT1.ex3.py"

saisir le nbr de sommet : 34

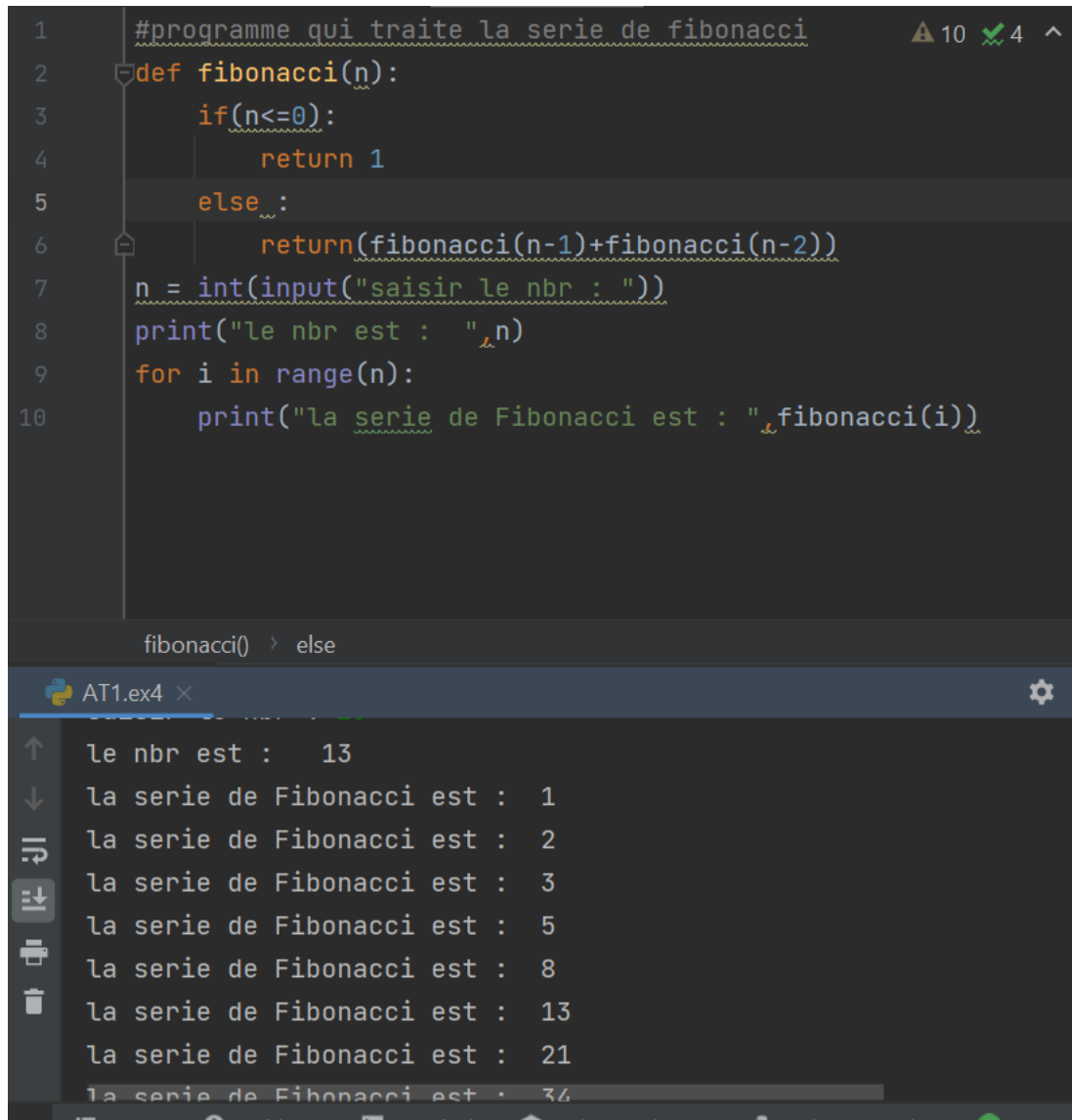
595

Process finished with exit code 0

## Code source exercice 3 :

```
def somme(n):
    if n==0:
        return 0
    else:
        return n + somme(n-1)
n = int(input("saisir le nbr de sommet : "))
print(somme(n))
# un script qui calcule la somme par la récursivité
```

## Exercice 4 atelier 1 :



The screenshot shows a Python IDE with a dark theme. The top pane displays the source code for a Fibonacci program. The code defines a recursive function `fibonacci(n)` that returns 1 for `n <= 0` and the sum of the two preceding numbers otherwise. It then prompts the user for a number `n` and prints the Fibonacci sequence up to `n`. The bottom pane shows the execution of the program, where the user has entered 13, and the output displays the sequence: 1, 2, 3, 5, 8, 13, 21, 34.

```
1 #programme qui traite la serie de fibonacci
2 def fibonacci(n):
3     if(n<=0):
4         return 1
5     else:
6         return(fibonacci(n-1)+fibonacci(n-2))
7 n = int(input("saisir le nbr : "))
8 print("le nbr est : ",n)
9 for i in range(n):
10    print("la serie de Fibonacci est : ",fibonacci(i))
```

fibonacci() > else

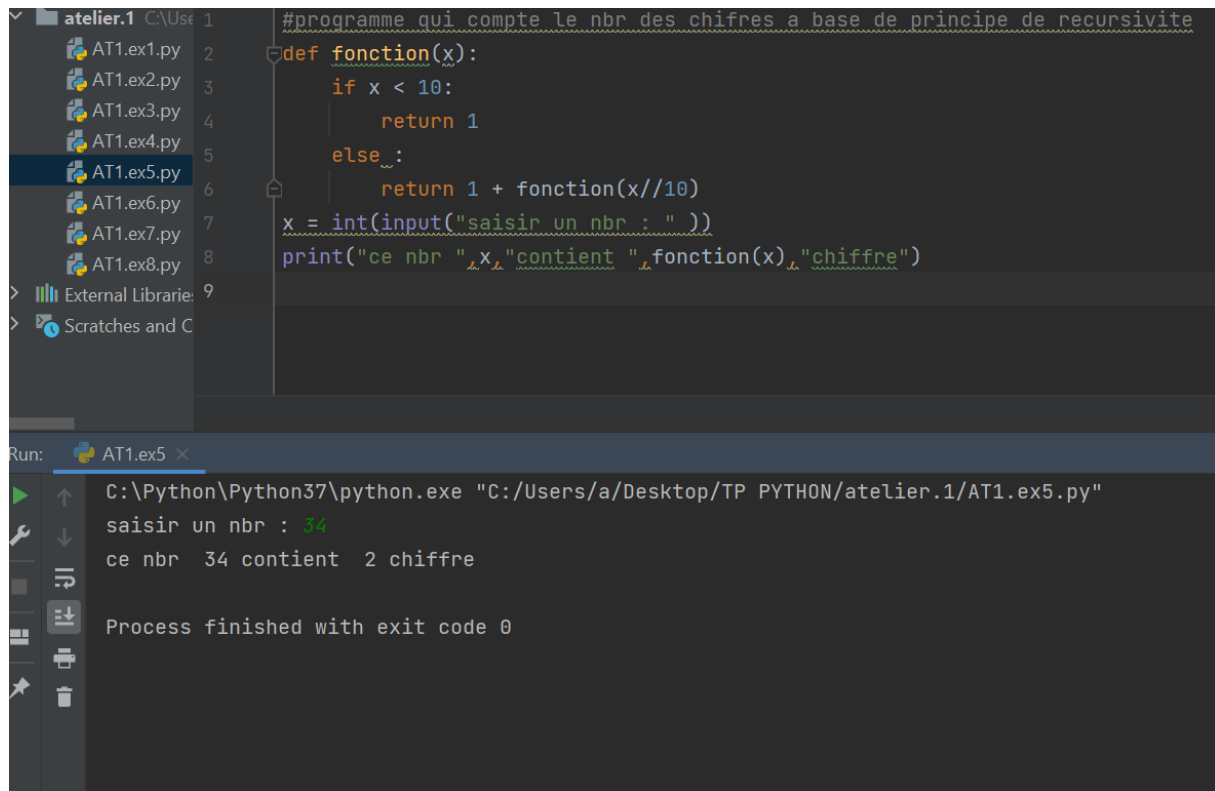
AT1.ex4 ×

↑ le nbr est : 13  
↓ la serie de Fibonacci est : 1  
↺ la serie de Fibonacci est : 2  
↻ la serie de Fibonacci est : 3  
⇓ la serie de Fibonacci est : 5  
⏏ la serie de Fibonacci est : 8  
🗑 la serie de Fibonacci est : 13  
la serie de Fibonacci est : 21  
la serie de Fibonacci est : 34

## Code source exercice 4:

```
#programme qui traite la serie de fibonacci
def fibonacci(n):
    if(n<=0):
        return 1
    else :
        return(fibonacci(n-1)+fibonacci(n-2))
n = int(input("saisir le nbr : "))
print("le nbr est : ",n)
for i in range(n):
    print("la serie de Fibonacci est : ",fibonacci(i))
```

## Exercice 5 atelier 1 :



The screenshot shows a Python IDE with a file explorer on the left containing files AT1.ex1.py through AT1.ex8.py. The main editor displays the following code:

```
#programme qui compte le nbr des chiffres a base de principe de recursivite
def fonction(x):
    if x < 10:
        return 1
    else:
        return 1 + fonction(x//10)
x = int(input("saisir un nbr : "))
print("ce nbr ",x,"contient ",fonction(x),"chiffre")
```

The Run console at the bottom shows the execution of AT1.ex5.py with the following output:

```
C:\Python\Python37\python.exe "C:/Users/a/Desktop/TP PYTHON/atelier.1/AT1.ex5.py"
saisir un nbr : 34
ce nbr 34 contient 2 chiffre
Process finished with exit code 0
```

## Code source exercice 5 :

```
#programme qui compte le nbr des chiffres a base de principe de recursivite
def fonction(x):
    if x < 10:
        return 1
    else :
        return 1 + fonction(x//10)
x = int(input("saisir un nbr : "))
print("ce nbr ",x,"contient ",fonction(x),"chiffre")
```

## Exercice 6 atelier 1 :

```
1 def tri_bulle(tab):
2     n = len(tab)
3     for i in range(n):
4         for j in range(0, n-i-1):
5             if tab[j] > tab[j+1]:
6                 tab[j], tab[j+1] = tab[j+1], tab[j]
7     return tab
8 def tri_selection(tab):
9     for i in range(len(tab)):
10        min = i
11        for j in range(i+1, len(tab)):
12            if tab[min] > tab[j]:
13                min = j
14        tmp = tab[i]
15        tab[i] = tab[min]
16        tab[min] = tmp
17    return tab
```

```
18 def tri_insertion(tab):
19     for i in range(1, len(tab)):
20         k = tab[i]
21         j = i - 1
22         while j >= 0 and k < tab[j]:
23             tab[j+1] = tab[j]
24             j -= 1
25         tab[j+1] = k
26     return tab
27 tab = [20, 12, 45, 89, 1, 23, 54, 34, 60, 99]
28 print("TRI A BULL : ", tri_bulle(tab))
29 print("TRI PAR SELECTION : ", tri_selection(tab))
30 print("TRI PAR INSERTION : ", tri_insertion(tab))
```

tri\_insertion() > for i in range(1, len(tab))

AT1.ex6 ×

C:\Python\Python37\python.exe "C:/Users/a/Desktop/TP PYTHON/atelier.1/AT1.ex6.py"

TRI A BULL : [1, 12, 20, 23, 34, 45, 54, 60, 89, 99]

TRI PAR SELECTION : [1, 12, 20, 23, 34, 45, 54, 60, 89, 99]

TRI PAR INSERTION : [1, 12, 20, 23, 34, 45, 54, 60, 89, 99]

Process finished with exit code 0

## Code source exercice 6 :

```
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
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
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
tab = [20,12,45,89,1,23,54,34,60,99]
print("TRI A BULL : ",tri_bull(tab))
print("TRI PAR SELECTION : ",tri_selection(tab))
print("TRI PAR INSERTION : ",tri_insertion(tab))
```



## Exercice 7 atelier 1 :

```
1 def inverser(str):
2     return str[::-1]
3
4 str1 = input("saisir une string : ")
5 print(inverser(str1))
#programme inverseur des chiffres
```

AT1.ex7 ×

C:\Python\Python37\python.exe "C:/Users/a/Desktop/TP PYTHON/atelier.1/AT1.ex7.py"

saisir une string : 1233

3321

Process finished with exit code 0

## Code source exercice 7 :

```
def inverser(str):
    return str[::-1]
str1 = input("saisir une string : ")
print(inverser(str1))
#programme inverseur des chiffres
```

## Exercice 8 atelier 1 :

```
1 def trouve(str, char):
2     return str.count(char)
3
4 str1 = input("saisir le string : ")
5 char1 = input("saisir le char : ")
6 print("le char ", char1, "repeter ", trouve(str1, char1), "fois")
#la fct trouve la fréquence d'un caractère dans une chaîne
```

AT1.ex8 ×

C:\Python\Python37\python.exe "C:/Users/a/Desktop/TP PYTHON/atelier.1/AT1.ex8.py"

saisir le string : 1234

saisir le char : 1256

le char 1256 repeter 0 fois

Process finished with exit code 0

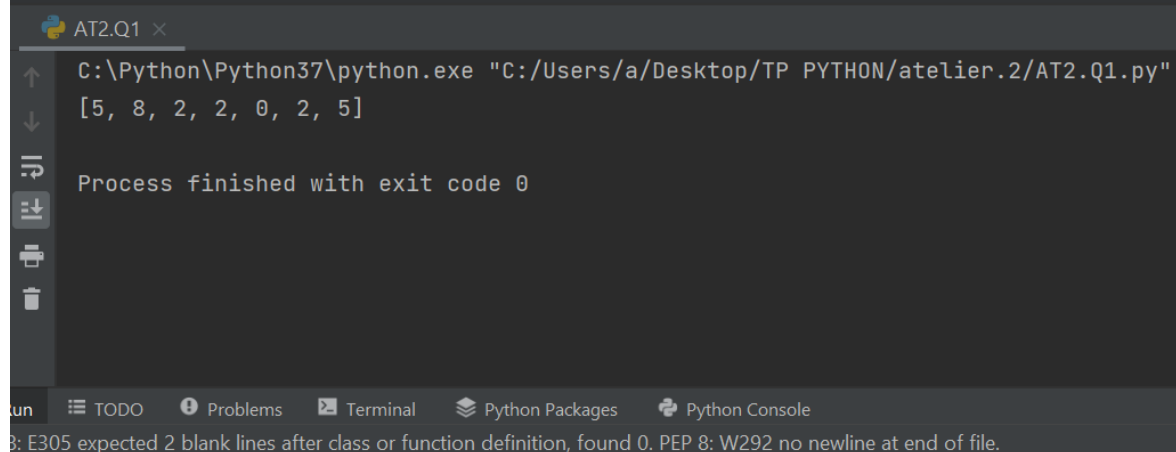
## Code source exercice 8 :

```
def trouve(str, char):
    return str.count(char)
str1 = input("saisir le string : ")
char1 = input("saisir le char : ")
print("le char ", char1, "repeter ", trouve(str1, char1), "fois")
#la fct trouve la fréquence d'un caractère dans une chaîne
```

# ATELIER 2

## QUESTIONE 1 atelier 2 :

```
1 def indice(list1, list2):
2     a, b = [], []
3     for i in range(0, len(list1)):
4         if i % 2 != 0: a.append(list1[i])
5     for j in range(0, len(list2)):
6         if j % 2 == 0: b.append(list2[j])
7     return a + b
8 print(indice([1,5,3,8,7,2,6],[2,6,0,9,2,3,5]))
```



AT2.Q1 x

```
C:\Python\Python37\python.exe "C:/Users/a/Desktop/TP PYTHON/atelier.2/AT2.Q1.py"
```

[5, 8, 2, 2, 0, 2, 5]

Process finished with exit code 0

run | TODO | Problems | Terminal | Python Packages | Python Console

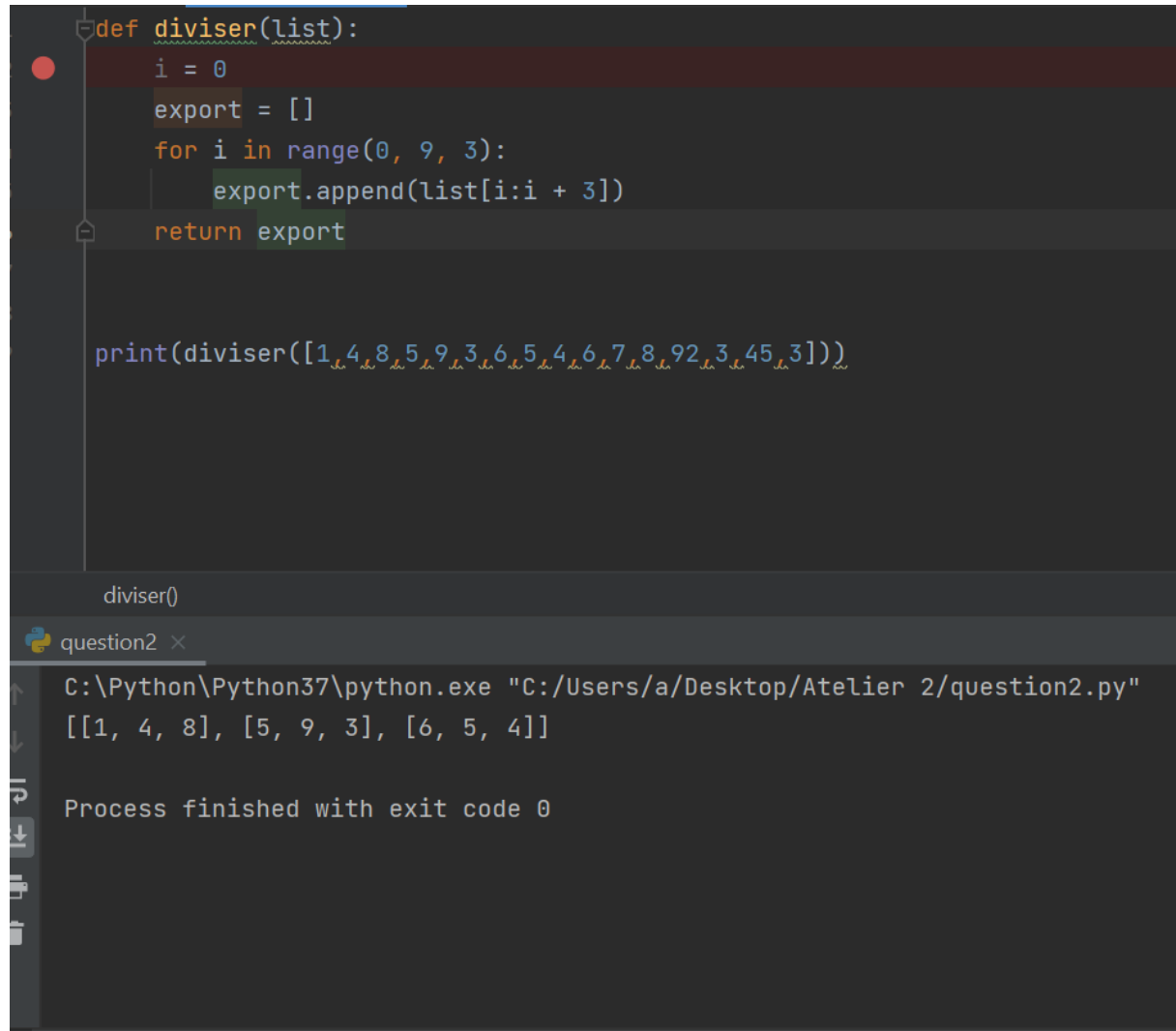
B: E305 expected 2 blank lines after class or function definition, found 0. PEP 8: W292 no newline at end of file.

## Code source QUESTIONE 1 :

```
def indice(list1, list2):
    a, b = [], []
    for i in range(0, len(list1)):
        if i % 2 != 0: a.append(list1[i])
    for j in range(0, len(list2)):
        if j % 2 == 0: b.append(list2[j])
```

```
    return a + b
print(indice([1,5,3,8,7,2,6,], [2,6,0,9,2,3,5]))
```

## QUESTIONE 2 atelier 2 :

A screenshot of a Python IDE. The top pane shows a function definition: 

```
def diviser(list):
    i = 0
    export = []
    for i in range(0, 9, 3):
        export.append(list[i:i + 3])
    return export
```

 Below the function is a call to the function: 

```
print(diviser([1,4,8,5,9,3,6,5,4,6,7,8,92,3,45,3]))
```

 The bottom pane shows the execution of the function, with the output: 

```
diviser()
[[1, 4, 8], [5, 9, 3], [6, 5, 4]]
```

 Below the output, it says "Process finished with exit code 0".

```
def diviser(list):
    i = 0
    export = []
    for i in range(0, 9, 3):
        export.append(list[i:i + 3])
    return export

print(diviser([1,4,8,5,9,3,6,5,4,6,7,8,92,3,45,3]))
```

## Code source QUESTIONE 2 :

```
def diviser(list):
    i = 0
    export = []
    for i in range(0, 9, 3):
        export.append(list[i:i + 3])
    return export

print(diviser([1,4,8,5,9,3,6,5,4,6,7,8,92,3,45,3]))
```

## QUESTIONE 3 atelier 2 :

```
1  # programme compte l'occurrence de chaque élément et montrer le nombre de chaque élément
2  def occur(list1):
3      x = {}
4      for i in list1:
5          if str(i) in x:
6              x[str(i)] = x.get(str(i)) + 1
7          else:
8              x[str(i)] = 1
9      return x
10
11 print(occur([1,2,2,2,3,4,5,77,77,65,43]))
```

AT2.Q3 x

```
C:\Python\Python37\python.exe "C:/Users/a/Desktop/TP PYTHON/atelier.2/AT2.Q3.py"
{'1': 1, '2': 3, '3': 1, '4': 1, '5': 1, '77': 2, '65': 1, '43': 1}
```

Process finished with exit code 0

TODO Problems Terminal Python Packages Python Console

## Code source QUESTIONE 3 :

```
# programme compte l'occurrence de chaque élément et montrer le nombre de
chaque élément
def occur(list1):
    x = {}
    for i in list1:
        if str(i) in x:
            x[str(i)] = x.get(str(i)) + 1
        else:
            x[str(i)] = 1
    return x

print(occur([1,2,2,2,3,4,5,77,77,65,43]))
```

## QUESTIONE 4 atelier 2 :

```
1  '''la fct commun est pour afficher les nbrs repeter dans les
2  deux set et de supprimer les nombre non repeter au 1er set '''
3  def commun(set1, set2):
4      x = set(())
5      for i in set1:
6          for j in set2:
7              if i == j: x.add(i)
8      y = set1.copy()
9      for i in set1:
10         if i in x: y.remove(i)
11
12     print("les nbrs commun sont : ", x)
13     print("les nombres supprime sont :", y)
14
15     print(commun({21,11,56,87,44,90}, {11,33,87,90}))
16
```

commun() > for i in set1 > for j in set2

AT2.Q4 x

C:\Python\Python37\python.exe "C:/Users/a/Desktop/TP PYTHON/atelier.2/AT2.Q4.py"

les nbrs commun sont : {90, 11, 87}

les nombres supprime sont : {21, 56, 44}

None

Process finished with exit code 0

## Code source QUESTIONE 4 :

```
'''la fct commun est pour afficher les nbrs repeter dans les
deux set et de supprimer les nombre non repeter au 1er set '''
def commun(set1, set2):
    x = set(())
    for i in set1:
        for j in set2:
            if i == j: x.add(i)
    y = set1.copy()
    for i in set1:
        if i in x: y.remove(i)

    print("les nbrs commun sont : ", x)
    print("les nombres supprime sont :", y)
```

```
print(commun({21,11,56,87,44,90}, {11,33,87,90}))
```

## QUESTIONE 5 atelier 2 :

```
1 def select(list1,dictA):
2     keys = list(dictA.keys())
3     resultat = []
4     for i in keys:
5         for j in list1:
6             if dictA[str(i)] == j:
7                 resultat.append(j)
8     return resultat
9 print(select([47,64,69,37,76,83,95,97],
10 {'Yassine':47, 'Imane':69, 'Mohammed':76, 'Abir':97}))
```

AT2.Q5 x

C:\Python\Python37\python.exe "C:/Users/a/Desktop/TP PYTHON/atelier.2/AT2.Q5.py"

[47, 69, 76, 97]

Process finished with exit code 0

## Code source QUESTIONE 5 :

```
def select(list1,dictA):
    keys = list(dictA.keys())
    resultat = []
    for i in keys:
        for j in list1:
            if dictA[str(i)] == j :
                resultat.append(j)
    return resultat
print(select([47,64,69,37,76,83,95,97],
{'Yassine':47, 'Imane':69, 'Mohammed':76, 'Abir':97}))
```

