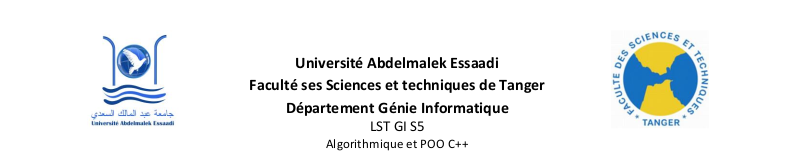
**

*RAPPORT DE TP PYTHON*

**ATELIER 1 ET 2**

Contenu de rapport

§ introduction :

§ ATELIER 1:

§ solution des exercices:

§ sortie du programme et commentaires:

§ ATELIER 2 :

§ solution des exercices:

§ sortie du programme et commentaires:

Présenté Par:

Encadré par :

**° Pr. Ikram Ben Abdel Ouahab**

**° Pr. Lotfi ELAACHAK**

**° Riyadi Adnane.**

**ATELIER 1 :**

**1. INTRODUCTION:**

Le langage de programmation Python a été créé en 1989 par Guido van Rossum, aux Pays-Bas. Le nom Python vient d'un hommage à la série télévisée Monty Python's Flying Circus dont G. van Rossum est fan. La première version publique de ce langage a été publiée en 1991.

La dernière version de Python est la version 3. Plus précisément, la version 3.7 a été publiée en juin 2018.

La [*Python Software Foundation*](https://www.python.org/psf/) est l'association qui organise le développement de Python et anime la communauté de développeurs et d'utilisateurs.

**2.** **solution des exercices:**

**Exercice 1 :**

**SOLUTION :**

def factoriel (x):

if x==1 :

return 1

else :

return( x \* factoriel(x-1))

m=5

s=0

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

s= s + factoriel(i)/i

print(s)

**Exercice 2 :**

**SOLUTION :**

def convert(x):

if x==0:

return 0

else:

return ((x%2)+ (10\*convert(x//2)))

n= int(input("donner le nombre:"))

print(convert(n))

**Exercice 3 :**

**SOLUTION :**

def recursiv(x):

if x==0:

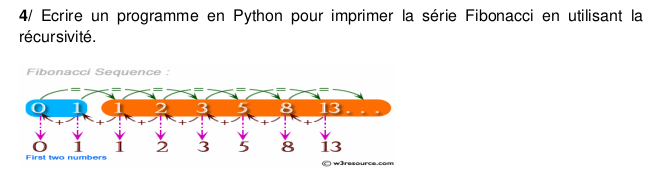
return 0

else:

return( x+ recursiv(x-1))

n= int(input("give me a number of your choice"))

print( recursiv( n ))

**Exercice 4 :**

**SOLUTION :**

def fibonn(n):

if n==0:

return 0

elif n==1:

return 1

else:

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

var = int(input("donner le nombre de fois"))

for i in range (0, var):

print(fibonn(i), end=' ')

**Exercice 5 :**

**SOLUTION :**

def calcul(x):

if x<10:

return 1

else:

return(1 + calcul(x//10))

n=int(input("give ne a number"))

print( calcul(n ))

**Exercice 6 :**

**SOLUTION :**

soit le tableau suivant comme exemple :

from array import\*

tab = ([98, 22, 15, 32, 2, 74, 63, 70, 34, 45, 754, 1235])

# Programme Python pour faire du Tri à bulle :

def tri\_bulle(T):

n=len(T)

tri=True

while tri == True:

tri=False

for i in range(0, n-1):

if T[i] >= T[i+1] :

tmp = T[i]

T[i]= T[i+1]

T[i+1]=tmp

tri= True

n=n-1

print("le tableau avant le tri a bulle:\n\n", tab)

print("---------------------------------------------------------------------------------\n\n")

tri\_bulle(tab)

print ("Le tableau apres le tri a bulle est:\n\n", tab)

# Programme Python pour faire du Tri par selection :

def tri\_selct(T):

for i in range(0, len(T)-1):

min= T[i]

val\_min= i

for j in range(i+1, len(T)):

if T[j]< min:

min = T[j]

val\_min=j

tmp= T[i]

T[i]=T[val\_min]

T[val\_min] = tmp

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n")

print("le tableau avant le tri par selction:\n\n", tab)

print("---------------------------------------------------------------------------------\n\n")

tri\_selct(tab)

print("le tableau apres le tri par selection est :\n\n", tab)

# Programme Python pour faire du Tri par insertion:

def tri\_inser(T):

for i in range (1, len(T)):

tmp = T[i]

j= i-1

while j>=0 and T[j]> tmp:

T[j+1] = T[j]

j= j-1

T[j+1] = tmp

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n")

print("le tableau avant le tri par insertion:\n\n", tab)

print("---------------------------------------------------------------------------------\n\n")

tri\_inser(tab)

print("le tableau apres le tri par insertion est :\n\n", tab)

**Exercice 7 :**

**SOLUTION :**

def inverse(list2):

newList = list (reversed(list2))

print(newList)

str1 = str(input("entrer la phrase a inverser:\n"))

print("voici votre phrase avant l'inverser:\n",str1)

n= len(str1)

list1=[]

for i in range(0,n):

list1.append(str1[i])

print(list1)

print(inverse(list1))

**Exercice 8 :**

**SOLUTION :**

methode 1 :

from collections import Counter

def freq(char1):

Val = Counter(str1)

print("le nombre d'occurence de ", char, "et:", Val[char1])

str1 = str(input("entrer votre phrase:\n"))

char=(input("entrer le caracter a chercher son nombre d'occurence:\n"))

freq(char)

methode 2 :

def nombre\_occurence(l, char):

n=0

for i in l:

if( i == char):

n=n+1

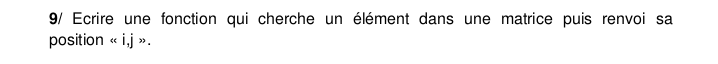
return n

str1 = str(input("entrer votre phrase:\n"))

char=(input("entrer le caracter a chercher son nombre d'occurence:\n"))

print("le nombre d'occurence de ", char, "est", nombre\_occurence(str1, char))

**Exercice 9 :**

**SOLUTION :**

def position\_nombre(l, char):

lig=0

col=0

for i in range(len(l)) :

for j in range(len(l[i])):

if( l[i][j]== char):

lig= i

col=j

print("la position de ", char, "est", (lig, col))

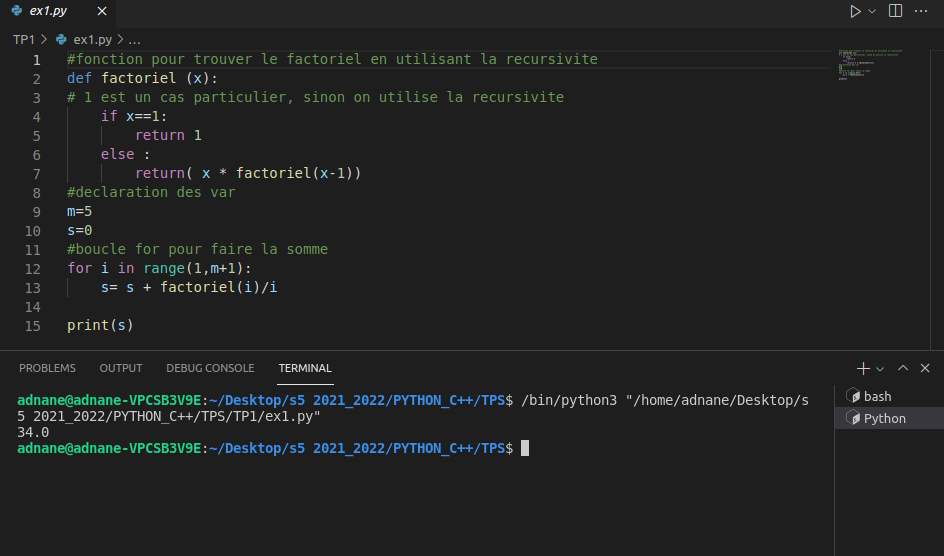
lst=[[1,2,3],[3,5,6],[7,8,9]]

char=int(input("entrer le caracter a chercher son position:\n"))

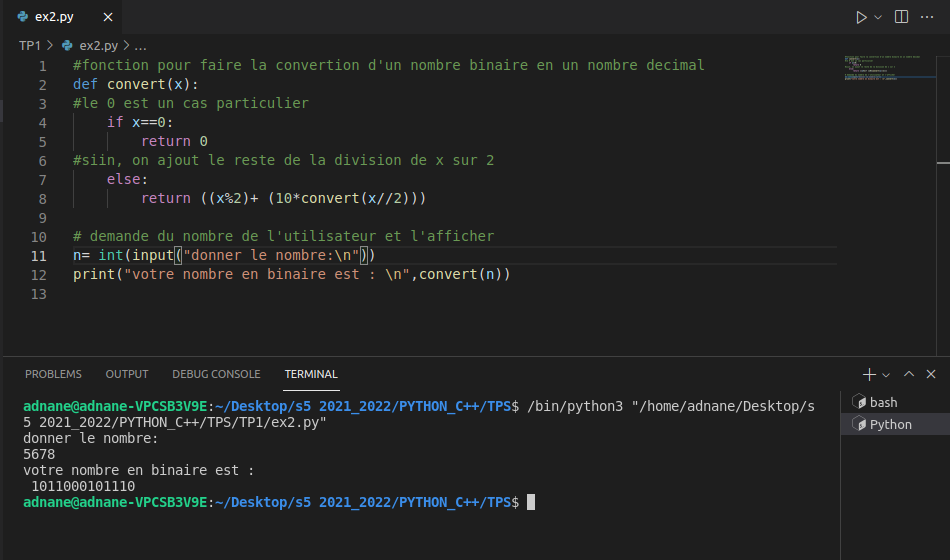
position\_nombre(lst, char)

**3. code source et comentaires**

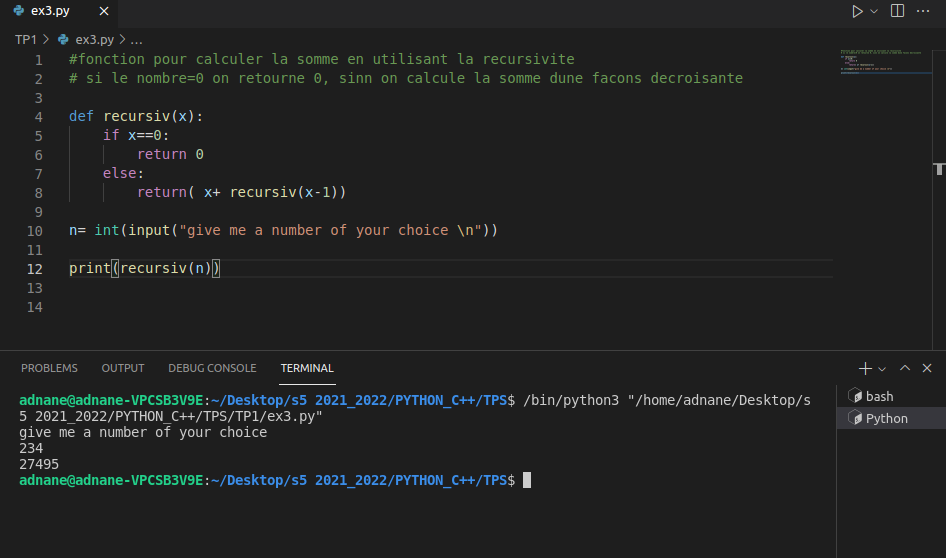
**et sortie du programme:**

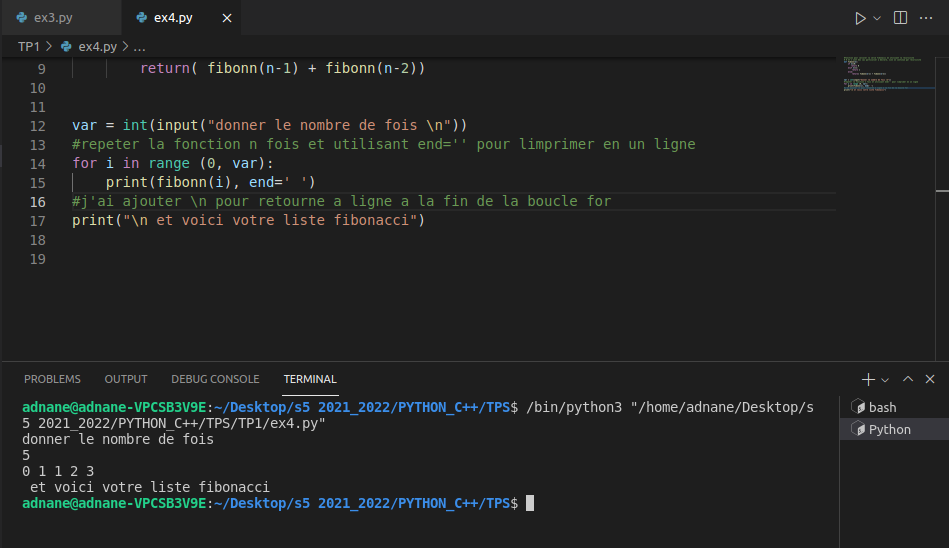
**Exercice 1 :**

**Exercice 2 :**

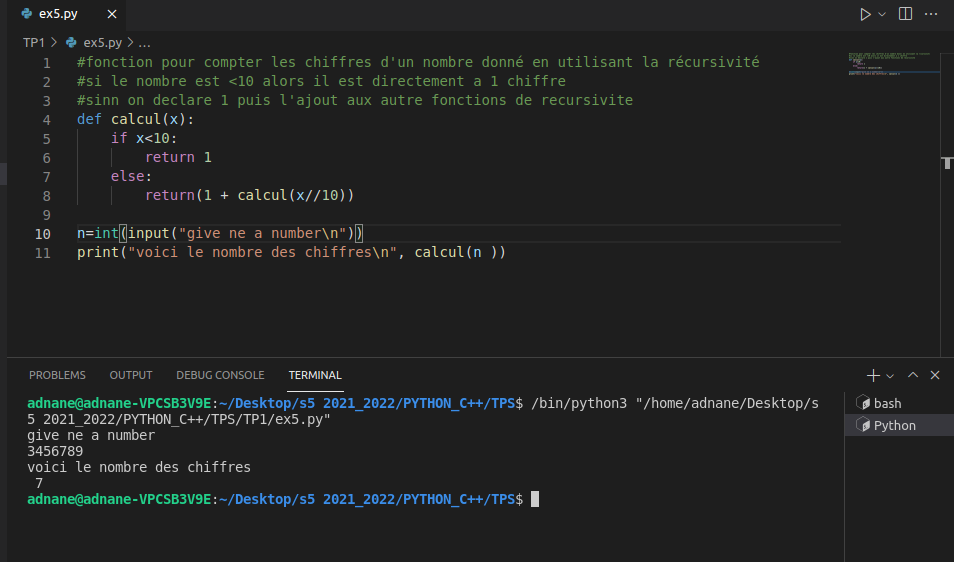
****

**Exercice 3 :**

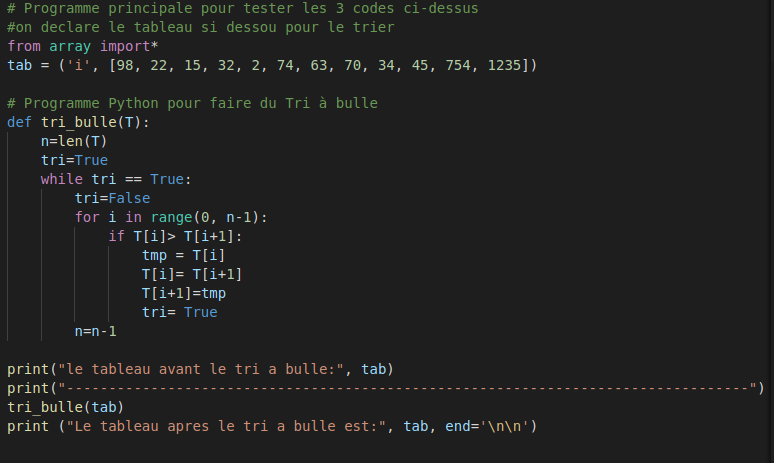
**Exercice 4 :**

****

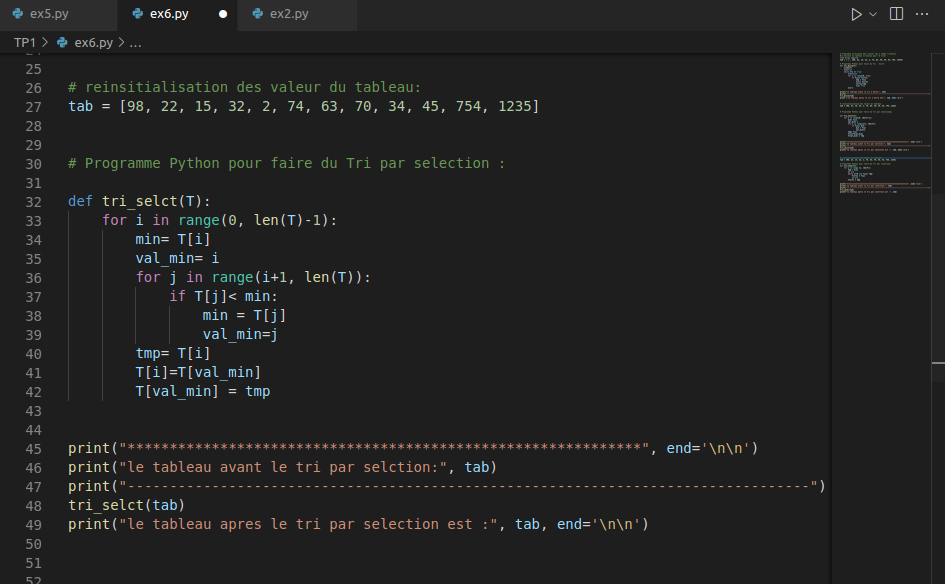
**Exercice 5 :**

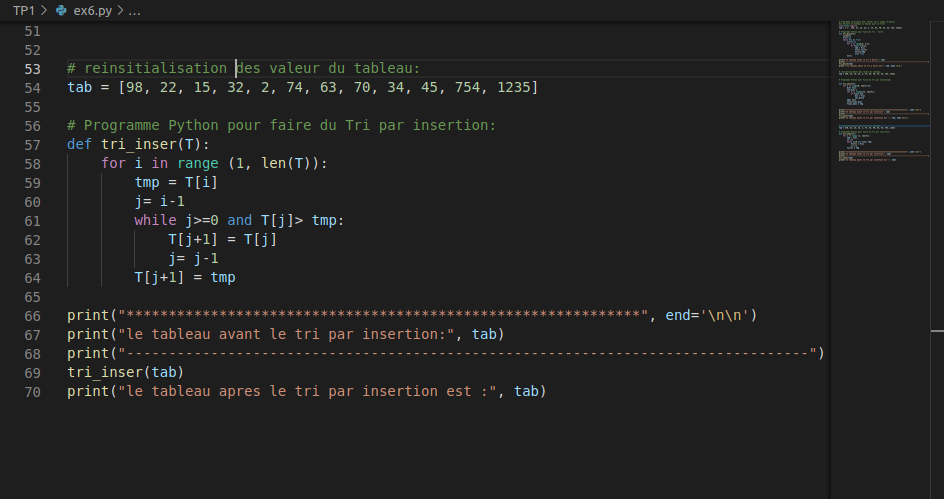
**Exercice 6 :**

**tri a bulle :**

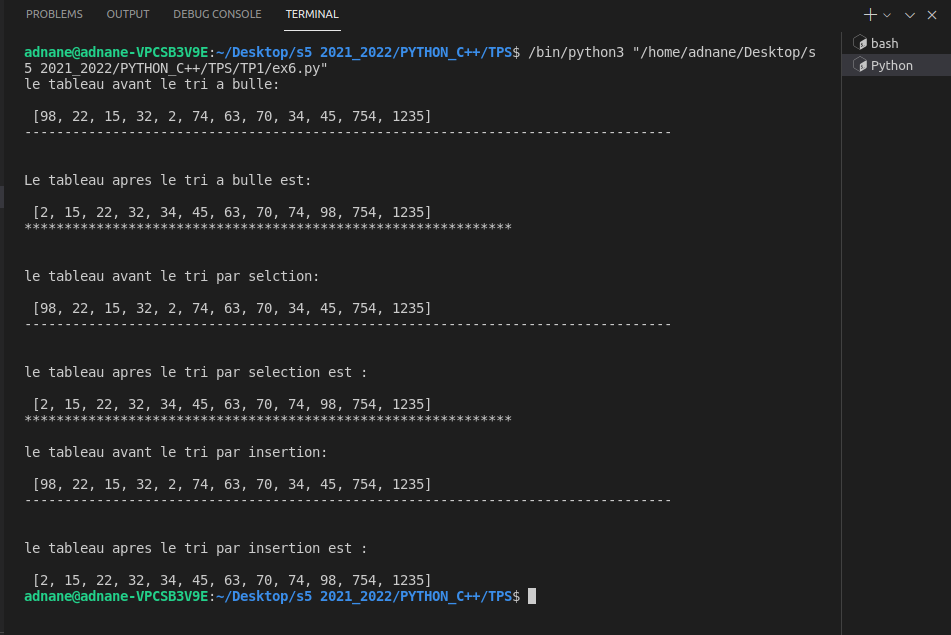
****

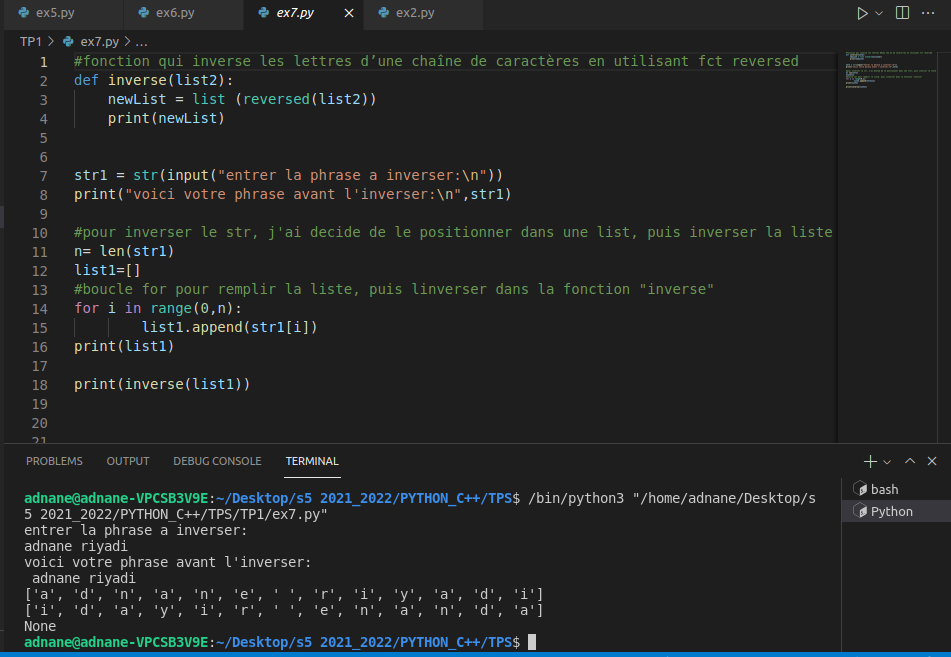
**tri par selection:**

**tri par insertion:**

****

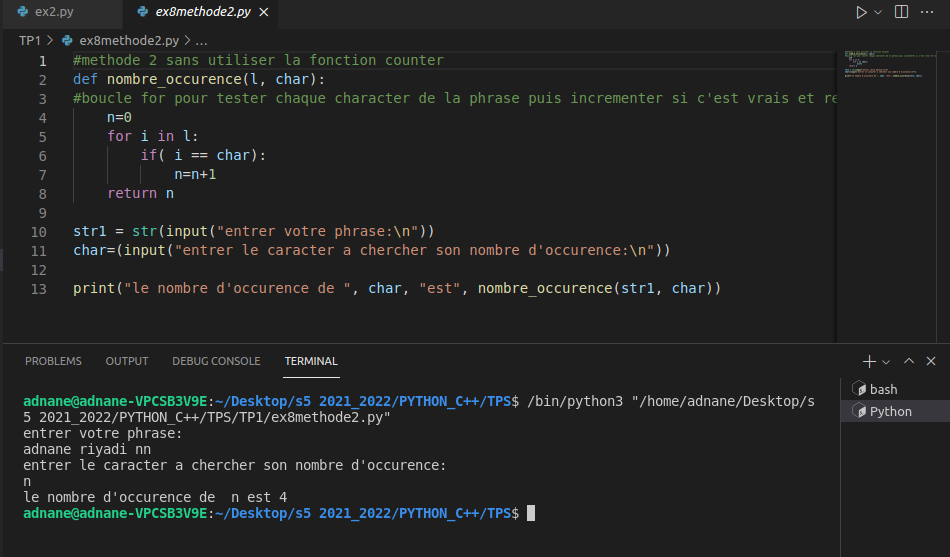
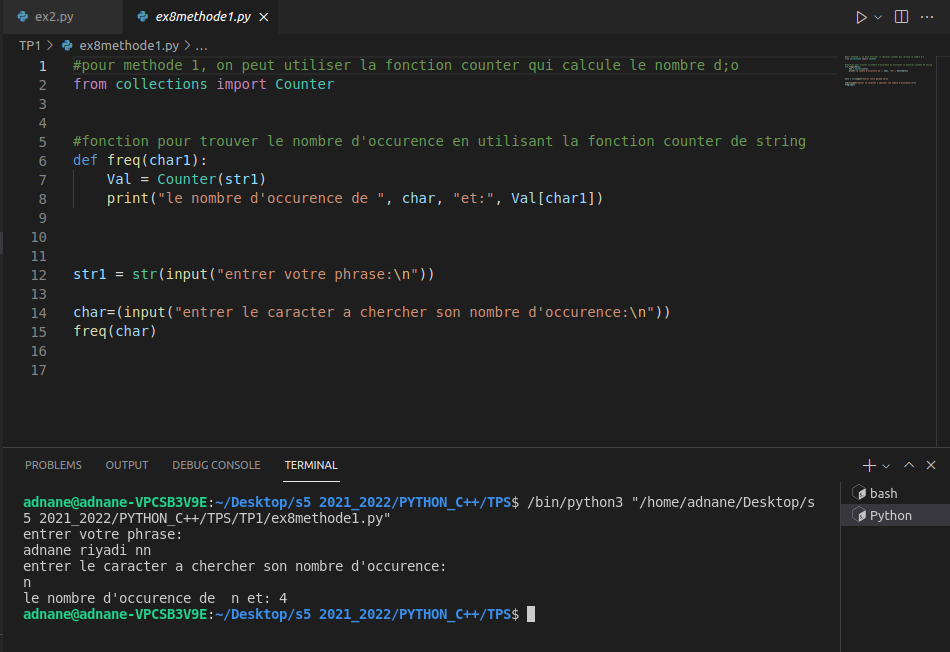
**sortie du program :**

**exercice 7 :**

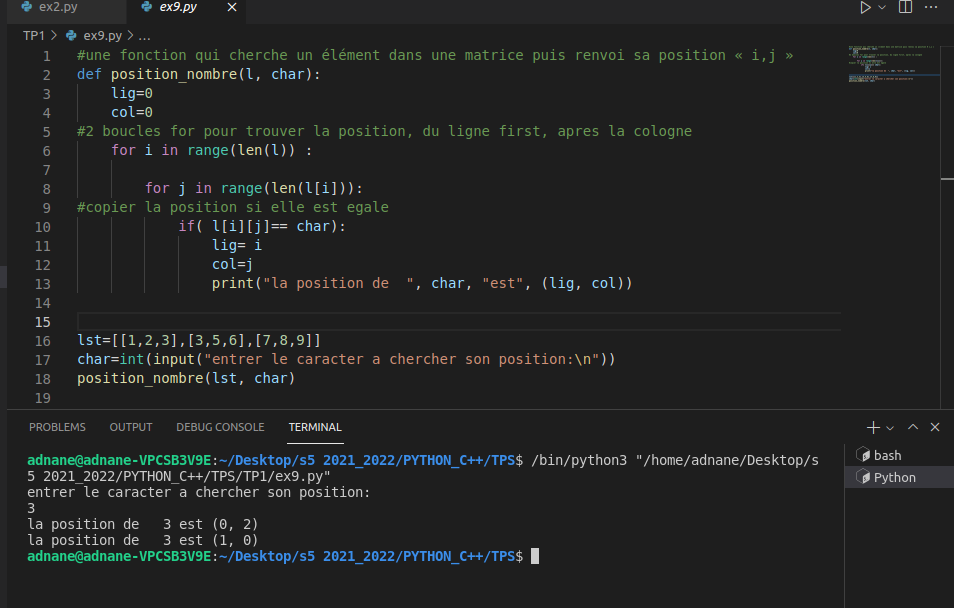
****

**exercice 8 :**

**methode 1 :**

**methode 2 :**

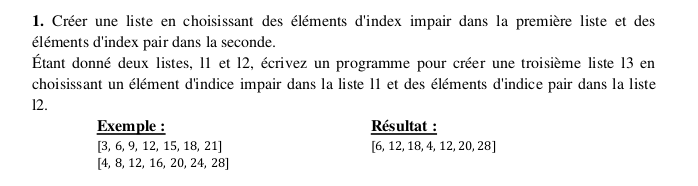
**exercice 9 :**

****

**ATELIER 2 :**

**1. solution des exercices:**

**exercice 1 :**

**SOLUTION :**

def new\_list (listA, listB):

N\_list = []

for i in range (1, len(listA), 2):

N\_list.append(listA[i])

for j in range( 0, len(listB), 2):

N\_list.append(listB[j])

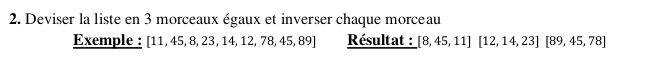
print(N\_list)

list1= [3, 6, 9, 12, 15, 18, 21]

list2 = [4, 8, 12, 16, 20, 24, 28]

new\_list(list1,list2)

**exercice 2 :**

**SOLUTION :**

def divise\_inverse(listA):

n= len(listA)

n=int(n/3)

list1=[]

list1= listA[0:n]

list1.reverse()

print(list1)

list2=[]

list2= listA[n:-n]

list2.reverse()

print(list2)

list3=[]

list3= listA[-n:]

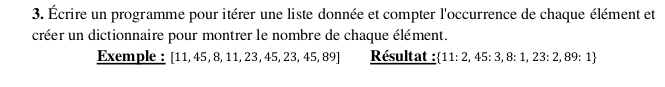
list3.reverse()

print(list3)

list\_a= [11, 45, 8, 23, 14, 12, 78, 45, 89,45,34,56]

divise\_inverse(list\_a)

**exercice 3 :**

**SOLUTION :**

def iterer(list):

my\_dict={}

for i in range(len(list)):

if list[i] in my\_dict:

my\_dict[list[i]]+= 1

else:

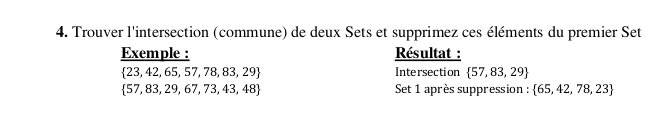
my\_dict[list[i]]= 1

print(my\_dict)

list1= [11, 45, 8, 11, 23, 45, 23, 45, 89]

iterer(list1)

**exercice 4 :**

****

**SOLUTION :**

def inter( listA, listB):

intersection= {n for n in listA if n in listB}

print( "lintersection de notre 2 listes est : \n",intersection)

listA= { n for n in listA if n not in listB }

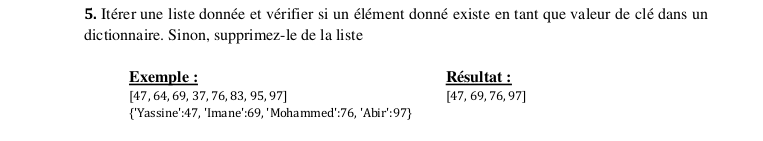
print("la nouvelle liste 1 est: \n",listA)

list1=[23, 42, 65, 57, 78, 83, 29]

list2= [57, 83, 29, 67, 73, 43, 48]

inter(list1,list2)

**exercice 5 :**

**SOLUTION :**

def test\_exictence (dict, listA):

listB=[]

for i in listA:

if i in my\_dict.values():

listB.append(i)

print(listB)

list1=[47, 64, 69, 37, 76, 83, 95, 97]

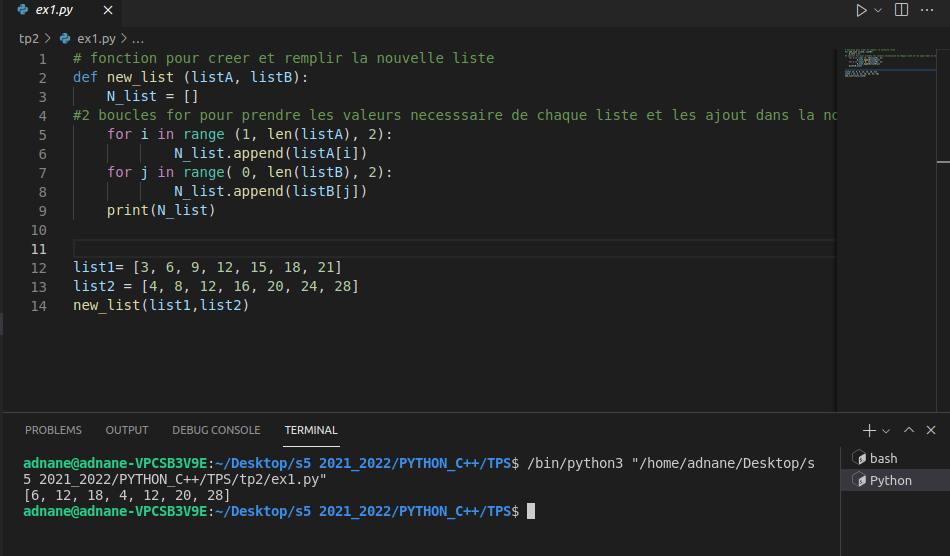
my\_dict= {'Yassine':47, 'Imane':69, 'Mohammed':76, 'Abir':97}

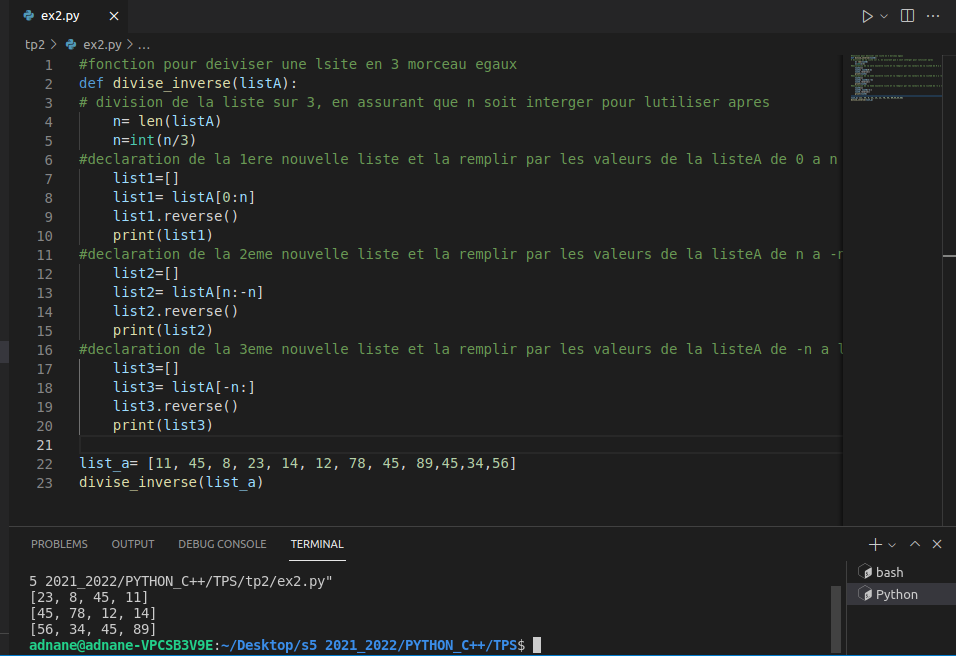
print(test\_exictence(my\_dict, list1))

**2. code source et comentaires**

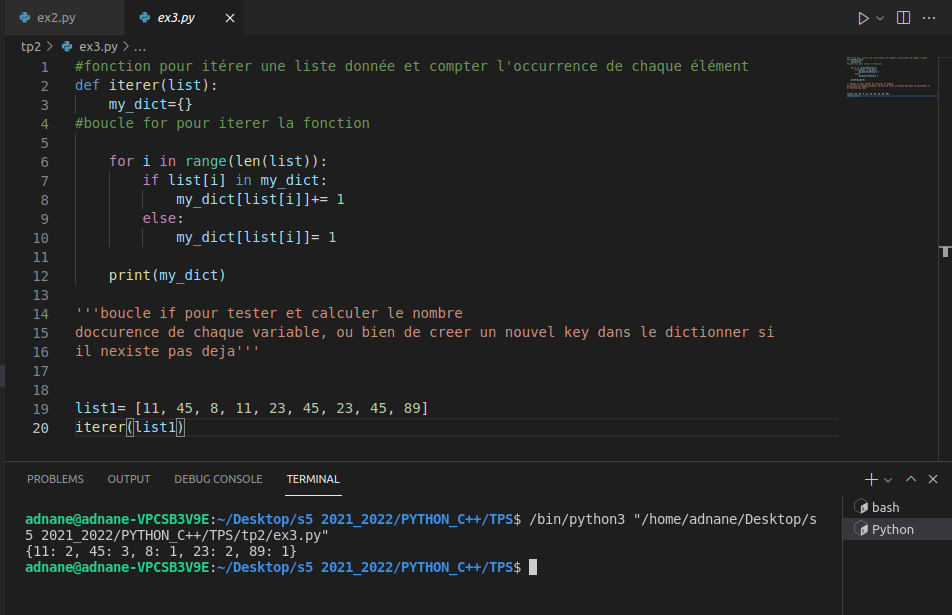
**et sortie du programme:**

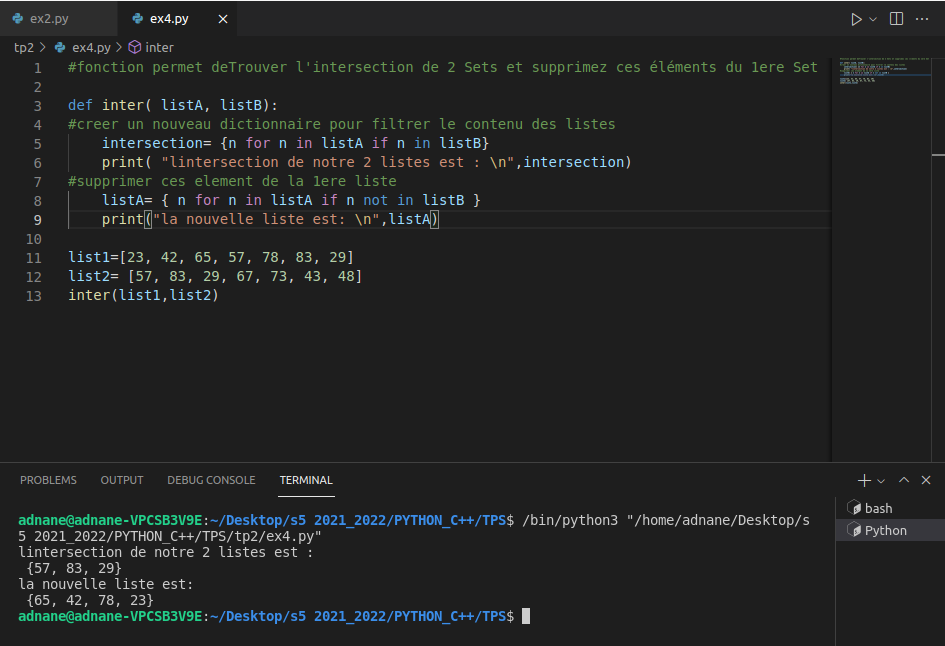
**exercice 1 :**

**exercice 2 :**

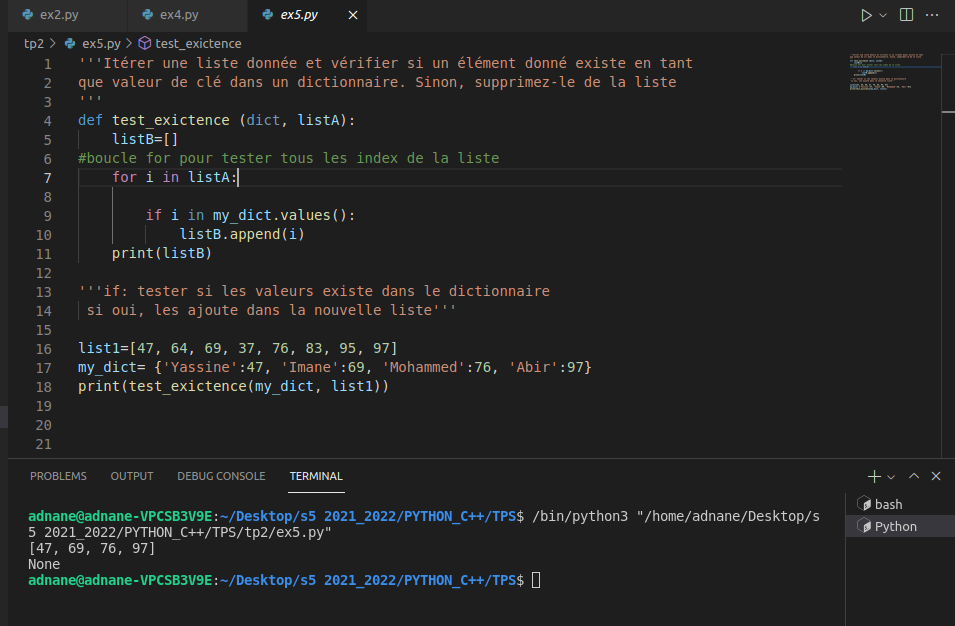
****

**exercice 3 :**

**exercice 4 :**



**exercice 5 :**

****