



Université Abdelmalek Essaadi
Faculté des Sciences et techniques de Tanger
Département Génie Informatique
LST GI S5
Algorithmique et POO C++



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:

Encadré par :

° Pr. Ikram Ben Abdel Ouahab

° Pr. Lotfi ELAACHAK

Présenté Par:

° 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://python.org) est l'association qui organise le développement de Python et anime la communauté de développeurs et d'utilisateurs.

2. solution des exercices:

Exercice 1 :

1/ Ecrire un programme en Python pour trouver la somme des séries $1! / 1 + 2! / 2 + 3! / 3 + 4! / 4 + 5! / 5$ en utilisant la fonction.

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 :

2/ Ecrire un programme en Python pour convertir le nombre décimal en nombre binaire en utilisant la fonction.

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))

```

3/ Ecrire un programme en Python pour calculer la somme des nombres de 1 à n en utilisant la récursivité.

Exercice 3 :

SOLUTION :

```

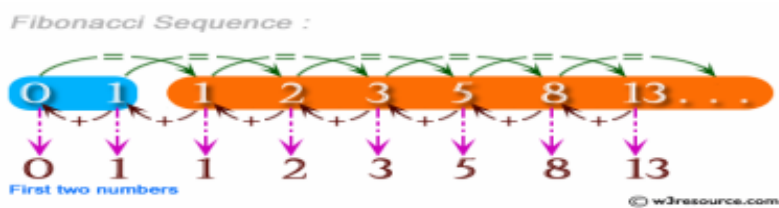
def recursiv(x):
    if x==0:
        return 0
    else:
        return( x+ recursiv(x-1))

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

print( recursiv( n ))

```

4/ Ecrire un programme en Python pour imprimer la série Fibonacci en utilisant la récursivité.



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 :

5/ Ecrire un programme en Python pour compter les chiffres d'un nombre donné en utilisant la récursivité.

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 :

6/ Ecrire trois fonctions python pour trier un tableau selon les algorithmes : tri a bull, tri par sélection et tri par insertion.

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)

```

7/ Ecrire un fonction Python qui inverse les lettres d'une chaîne de caractères.

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))

```

8/ Ecrire une fonction Python pour trouver la fréquence d'un caractère dans une chaîne.

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))
```

9/ Ecrire une fonction qui cherche un élément dans une matrice puis renvoi sa position « i,j ».

Exercice 9 :

SOLUTION :

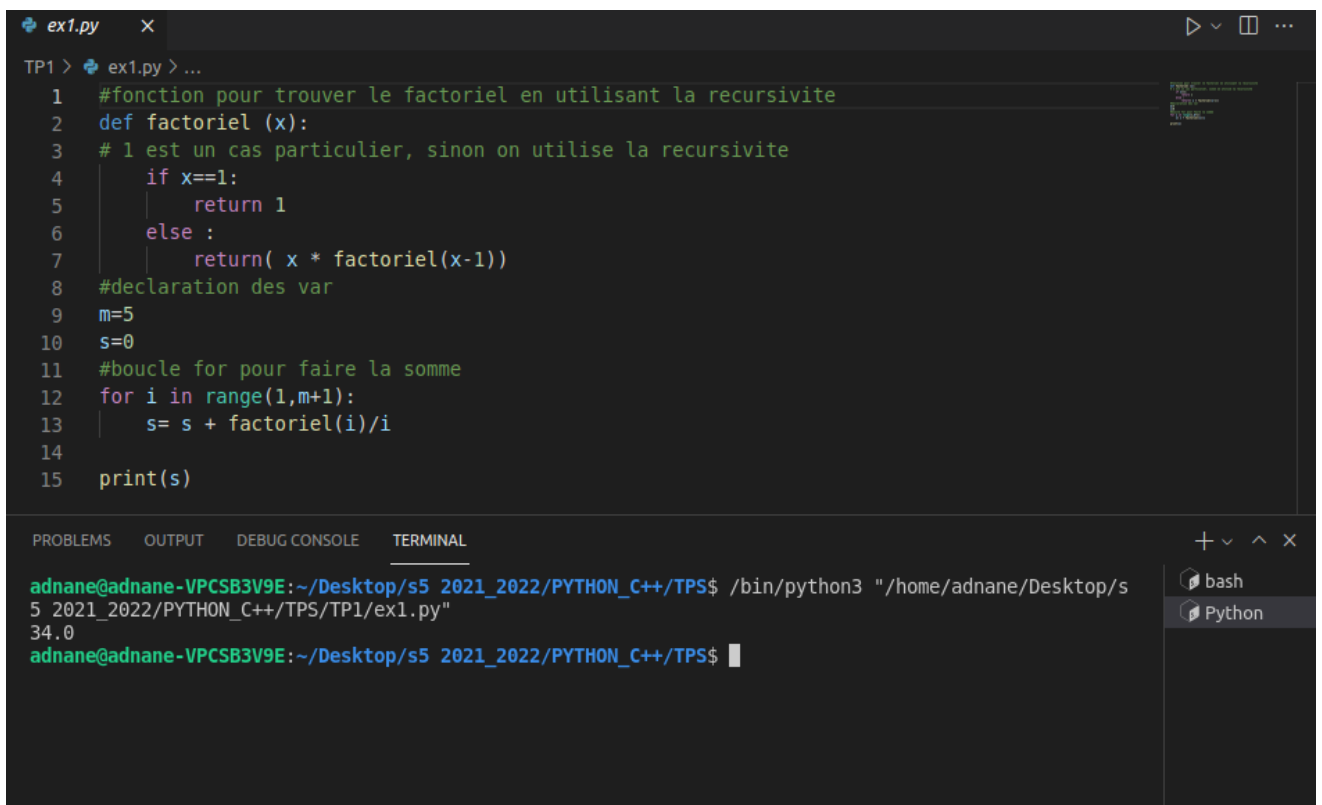
```
def position_nombre(l, char):  
    lig=0  
    col=0  
  
    for i in range(len(l)) :  
        for j in range(len(l[i])):  
            if( l[i][j]== char):  
                lig= i  
                col=j  
    print("la position de ", char, "est", (lig, col))
```

```
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 commentaires et sortie du programme:

Exercice 1 :



The screenshot shows a code editor with a file named `ex1.py` and a terminal window below it. The code in `ex1.py` defines a recursive function `factoriel` to calculate the factorial of a number `x`. It then calculates the sum of factorials from 1 to 5 and prints the result.

```
1 #fonction pour trouver le factoriel en utilisant la recursivite
2 def factoriel (x):
3     # 1 est un cas particulier, sinon on utilise la recursivite
4     if x==1:
5         return 1
6     else :
7         return( x * factoriel(x-1))
8 #declaration des var
9 m=5
10 s=0
11 #boucle for pour faire la somme
12 for i in range(1,m+1):
13     s= s + factoriel(i)/i
14
15 print(s)
```

The terminal window shows the command to run the script and the output:

```
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$ /bin/python3 "/home/adnane/Desktop/s5 2021_2022/PYTHON_C++/TPS/TP1/ex1.py"
34.0
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$
```

Exercice 2 :


```
ex2.py x
TP1 > ex2.py > ...
1 #fonction pour faire la conversion d'un nombre binaire en un nombre decimal
2 def convert(x):
3     #le 0 est un cas particulier
4     if x==0:
5         return 0
6     #siin, on ajout le reste de la division de x sur 2
7     else:
8         return ((x%2)+ (10*convert(x//2)))
9
10 # demande du nombre de l'utilisateur et l'afficher
11 n= int(input("donner le nombre:\n"))
12 print("votre nombre en binaire est : \n",convert(n))
13
```

adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS\$ /bin/python3 "/home/adnane/Desktop/s5 2021_2022/PYTHON_C++/TPS/TP1/ex2.py"

donner le nombre:
5678
votre nombre en binaire est :
1011000101110
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS\$

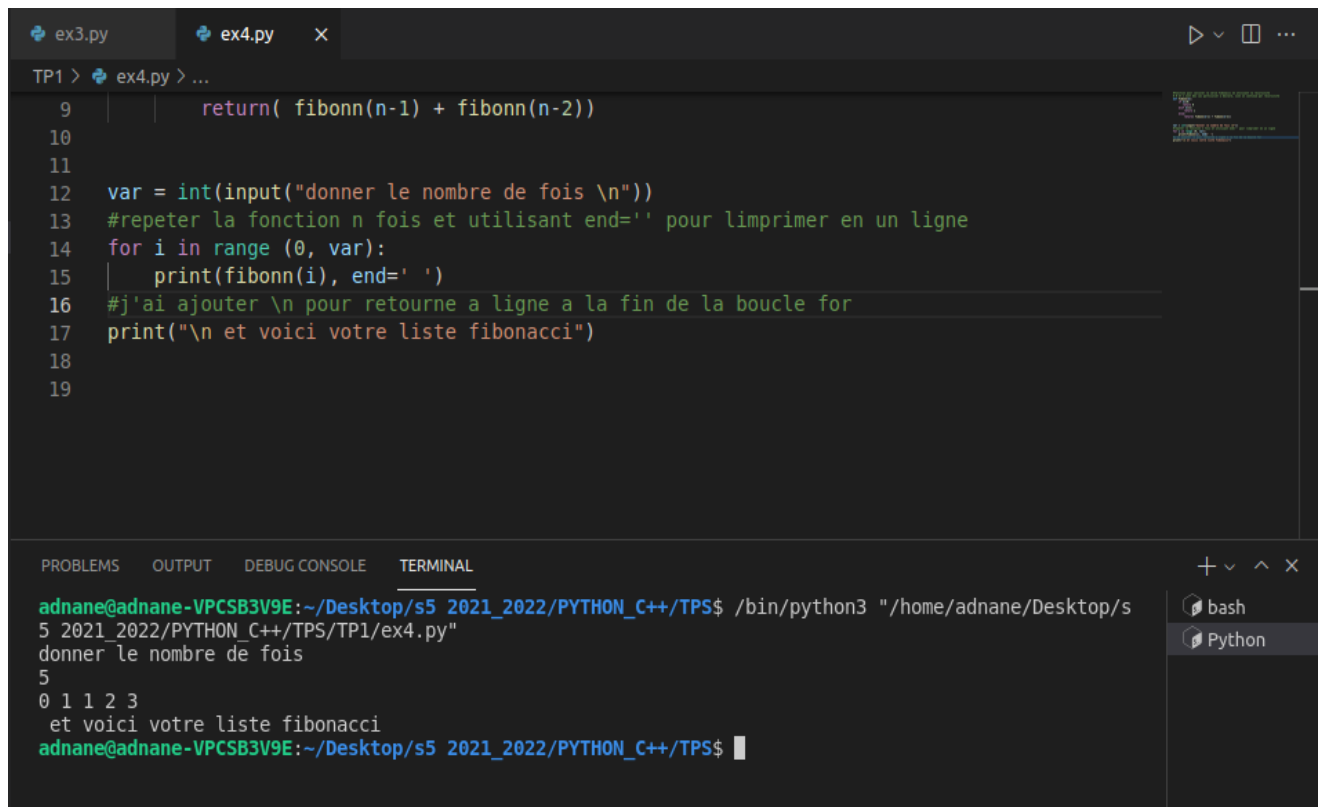
Exercise 3 :

```
ex3.py x
TP1 > ex3.py > ...
1 #fonction pour calculer la somme en utilisant la recursivite
2 # si le nombre=0 on retourne 0, sinn on calcule la somme dune facons decroisante
3
4 def recursiv(x):
5     if x==0:
6         return 0
7     else:
8         return( x+ recursiv(x-1))
9
10 n= int(input("give me a number of your choice \n"))
11
12 print(recursiv(n))
13
14
```

adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS\$ /bin/python3 "/home/adnane/Desktop/s5 2021_2022/PYTHON_C++/TPS/TP1/ex3.py"

give me a number of your choice
234
27495
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS\$

Exercise 4 :



The screenshot shows a code editor with two tabs: 'ex3.py' and 'ex4.py'. The 'ex4.py' tab is active, displaying a Python script. The script defines a recursive function 'fibonn' and uses it to calculate the first 'n' Fibonacci numbers. The terminal output shows the execution of the script, where the user enters '5' and the program prints the first 5 Fibonacci numbers: '0 1 1 2 3'.

```
9 |         return( fibonn(n-1) + fibonn(n-2))
10 |
11 |
12 |     var = int(input("donner le nombre de fois \n"))
13 |     #repete la fonction n fois et utilisant end='' pour l'imprimer en une ligne
14 |     for i in range(0, var):
15 |         print(fibonn(i), end=' ')
16 |     #j'ai ajouter \n pour retourner a ligne a la fin de la boucle for
17 |     print("\n et voici votre liste fibonacci")
18 |
19 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS\$ /bin/python3 "/home/adnane/Desktop/s5 2021_2022/PYTHON_C++/TPS/TP1/ex4.py"

donner le nombre de fois

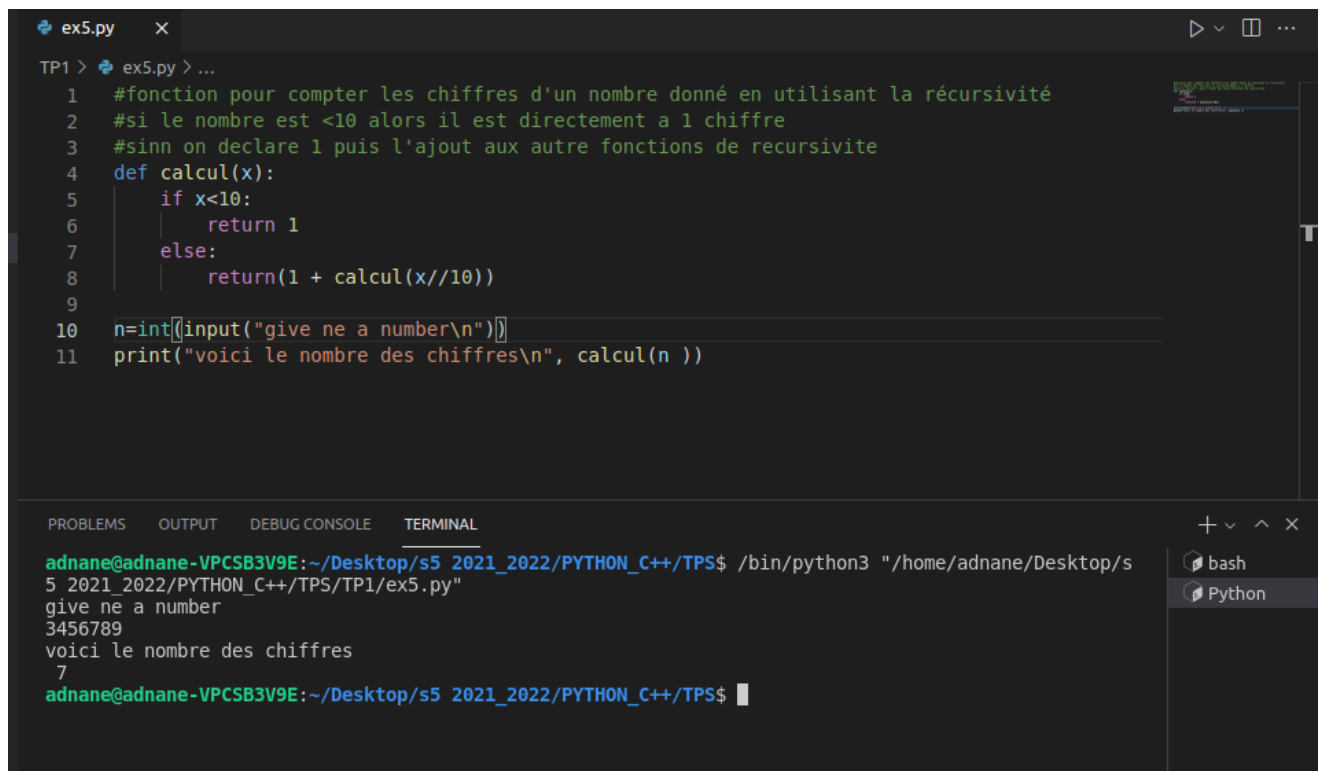
5

0 1 1 2 3

et voici votre liste fibonacci

adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS\$

Exercise 5 :



The screenshot shows a code editor with a tab 'ex5.py'. The script defines a recursive function 'calcul' to count the number of digits in a given number. The terminal output shows the execution of the script, where the user enters '3456789' and the program prints '7', which is the number of digits in the input.

```
1 | #fonction pour compter les chiffres d'un nombre donné en utilisant la récursivité
2 | #si le nombre est <10 alors il est directement a 1 chiffre
3 | #sinn on declare 1 puis l'ajout aux autres fonctions de récursivité
4 | def calcul(x):
5 |     if x<10:
6 |         return 1
7 |     else:
8 |         return(1 + calcul(x//10))
9 |
10 | n=int(input("give me a number\n"))
11 | print("voici le nombre des chiffres\n", calcul(n))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS\$ /bin/python3 "/home/adnane/Desktop/s5 2021_2022/PYTHON_C++/TPS/TP1/ex5.py"

give me a number

3456789

voici le nombre des chiffres

7

adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS\$

Exercise 6 :

tri a bulle :

```

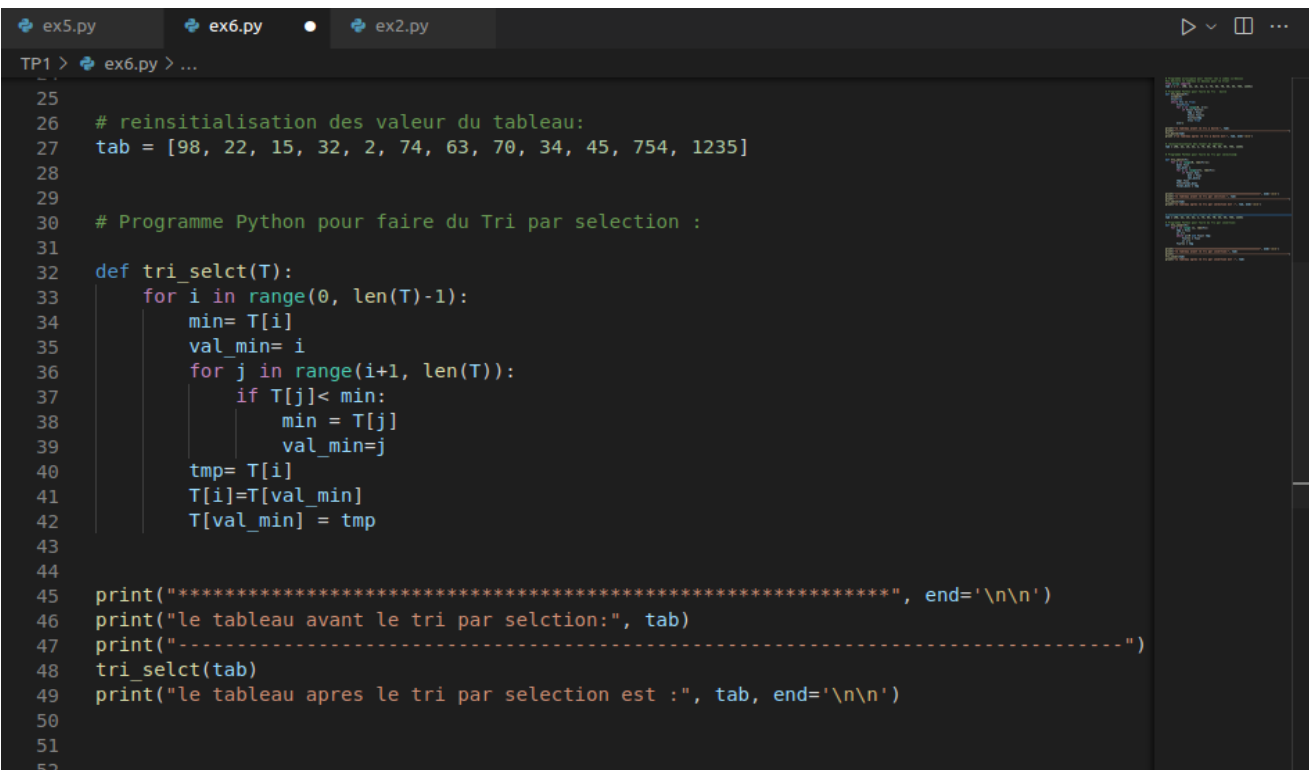
# Programme principale pour tester les 3 codes ci-dessus
#on declare le tableau si dessous pour le trier
from array import*
tab = ('i', [98, 22, 15, 32, 2, 74, 63, 70, 34, 45, 754, 1235])

# 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:", tab)
print("-----")
tri_bulle(tab)
print ("Le tableau apres le tri a bulle est:", tab, end='\n\n')

```

tri par selection:



```

ex5.py  ex6.py  ex2.py
TP1 > ex6.py > ...
25
26 # reinitialisation des valeur du tableau:
27 tab = [98, 22, 15, 32, 2, 74, 63, 70, 34, 45, 754, 1235]
28
29
30 # Programme Python pour faire du Tri par selection :
31
32 def tri_selct(T):
33     for i in range(0, len(T)-1):
34         min= T[i]
35         val_min= i
36         for j in range(i+1, len(T)):
37             if T[j]< min:
38                 min = T[j]
39                 val_min=j
40         tmp= T[i]
41         T[i]=T[val_min]
42         T[val_min] = tmp
43
44
45 print("*****", end='\n\n')
46 print("le tableau avant le tri par selction:", tab)
47 print("-----")
48 tri_selct(tab)
49 print("le tableau apres le tri par selection est :", tab, end='\n\n')
50
51
52

```

tri par insertion:

```
TP1 > ex6.py > ...
51
52
53 # reinitialisation des valeur du tableau:
54 tab = [98, 22, 15, 32, 2, 74, 63, 70, 34, 45, 754, 1235]
55
56 # Programme Python pour faire du Tri par insertion:
57 def tri_inser(T):
58     for i in range (1, len(T)):
59         tmp = T[i]
60         j= i-1
61         while j>=0 and T[j]> tmp:
62             T[j+1] = T[j]
63             j= j-1
64         T[j+1] = tmp
65
66 print("*****", end='\n\n')
67 print("le tableau avant le tri par insertion:", tab)
68 print("-----")
69 tri_inser(tab)
70 print("le tableau apres le tri par insertion est :", tab)
```

sortie du program :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$ /bin/python3 "/home/adnane/Desktop/s5 2021_2022/PYTHON_C++/TPS/ex6.py"
le tableau avant le tri a bulle:

[98, 22, 15, 32, 2, 74, 63, 70, 34, 45, 754, 1235]
-----

Le tableau apres le tri a bulle est:

[2, 15, 22, 32, 34, 45, 63, 70, 74, 98, 754, 1235]
*****

le tableau avant le tri par selction:

[98, 22, 15, 32, 2, 74, 63, 70, 34, 45, 754, 1235]
-----

le tableau apres le tri par selection est :

[2, 15, 22, 32, 34, 45, 63, 70, 74, 98, 754, 1235]
*****

le tableau avant le tri par insertion:

[98, 22, 15, 32, 2, 74, 63, 70, 34, 45, 754, 1235]
-----

le tableau apres le tri par insertion est :

[2, 15, 22, 32, 34, 45, 63, 70, 74, 98, 754, 1235]
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$
```

exercice 7 :

The image shows a Visual Studio Code editor window with a Python file named `ex7.py` open. The script defines a function `inverse` that takes a list and returns a new list with its elements reversed. It then prompts the user to enter a phrase, converts it to a list of characters, and prints the original list, the reversed list, and the reversed string.

```
1 #fonction qui inverse les lettres d'une chaîne de caractères en utilisant fct reversed
2 def inverse(list2):
3     newList = list (reversed(list2))
4     print(newList)
5
6
7 str1 = str(input("entrer la phrase a inverser:\n"))
8 print("voici votre phrase avant l'inverser:\n",str1)
9
10 #pour inverser le str, j'ai decide de le positionner dans une list, puis inverser la liste
11 n= len(str1)
12 list1=[]
13 #boucle for pour remplir la liste, puis linverser dans la fonction "inverse"
14 for i in range(0,n):
15     list1.append(str1[i])
16 print(list1)
17
18 print(inverse(list1))
19
20
21
```

The terminal output shows the execution of the script. The user enters the phrase "adnane riyadi". The program prints the list of characters, the reversed list, and the reversed string.

```
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$ /bin/python3 "/home/adnane/Desktop/s
5 2021_2022/PYTHON_C++/TPS/TP1/ex7.py"
entrer la phrase a inverser:
adnane riyadi
voici votre phrase avant l'inverser:
adnane riyadi
['a', 'd', 'n', 'a', 'n', 'e', ' ', 'r', 'i', 'y', 'a', 'd', 'i']
['i', 'd', 'a', 'y', 'i', 'r', ' ', 'e', 'n', 'a', 'n', 'd', 'a']
None
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$
```

[exercice 8 :](#)

[methode 1 :](#)

```
ex2.py  ex8methode1.py X
TP1 > ex8methode1.py > ...
1  #pour methode 1, on peut utiliser la fonction counter qui calcule le nombre d'o
2  from collections import Counter
3
4
5  #fonction pour trouver le nombre d'occurence en utilisant la fonction counter de string
6  def freq(char1):
7      Val = Counter(str1)
8      print("le nombre d'occurence de ", char, "et:", Val[char1])
9
10
11
12  str1 = str(input("entrer votre phrase:\n"))
13
14  char=(input("entrer le caracter a chercher son nombre d'occurence:\n"))
15  freq(char)
16
17

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$ /bin/python3 "/home/adnane/Desktop/s
5 2021_2022/PYTHON_C++/TPS/TP1/ex8methode1.py"
entrer votre phrase:
adnane riyadi nn
entrer le caracter a chercher son nombre d'occurence:
n
le nombre d'occurence de  n et: 4
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$
```

methode 2 :

```
ex2.py  ex8methode2.py X
TP1 > ex8methode2.py > ...
1  #methode 2 sans utiliser la fonction counter
2  def nombre_occurence(l, char):
3      #boucle for pour tester chaque caracter de la phrase puis incrementer si c'est vrais et re
4      n=0
5      for i in l:
6          if( i == char):
7              n=n+1
8      return n
9
10  str1 = str(input("entrer votre phrase:\n"))
11  char=(input("entrer le caracter a chercher son nombre d'occurence:\n"))
12
13  print("le nombre d'occurence de ", char, "est", nombre_occurence(str1, char))

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$ /bin/python3 "/home/adnane/Desktop/s
5 2021_2022/PYTHON_C++/TPS/TP1/ex8methode2.py"
entrer votre phrase:
adnane riyadi nn
entrer le caracter a chercher son nombre d'occurence:
n
le nombre d'occurence de  n est 4
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$
```

exercice 9 :

```
ex2.py  ex9.py  X
TP1 > ex9.py > ...
1  #une fonction qui cherche un élément dans une matrice puis renvoi sa position « i,j »
2  def position_nombre(l, char):
3      lig=0
4      col=0
5      #2 boucles for pour trouver la position, du ligne first, apres la cologne
6      for i in range(len(l)) :
7          for j in range(len(l[i])):
8              #copier la position si elle est egale
9              if( l[i][j]== char):
10                 lig= i
11                 col=j
12                 print("la position de ", char, "est", (lig, col))
13
14
15
16 lst=[[1,2,3],[3,5,6],[7,8,9]]
17 char=int(input("entrer le caractere a chercher son position:\n"))
18 position_nombre(lst, char)
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$ /bin/python3 "/home/adnane/Desktop/s
5 2021_2022/PYTHON_C++/TPS/TP1/ex9.py"
entrer le caractere a chercher son position:
3
la position de 3 est (0, 2)
la position de 3 est (1, 0)
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$
```

bash
Python

ATELIER 2 :

1. SOLUTION DES EXERCICES:

exercice 1 :

1. 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.

Exemple :

[3, 6, 9, 12, 15, 18, 21]
[4, 8, 12, 16, 20, 24, 28]

Résultat :

[6, 12, 18, 4, 12, 20, 28]

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 :

2. Deviser la liste en 3 morceaux égaux et inverser chaque morceau

Exemple : [11, 45, 8, 23, 14, 12, 78, 45, 89]

Résultat : [8, 45, 11] [12, 14, 23] [89, 45, 78]

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 :

3. Écrire un programme pour itérer une liste donnée et compter l'occurrence de chaque élément et créer un dictionnaire pour montrer le nombre de chaque élément.

Exemple : [11, 45, 8, 11, 23, 45, 23, 45, 89] **Résultat :**{11: 2, 45: 3, 8: 1, 23: 2, 89: 1}

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 :

4. Trouver l'intersection (commune) de deux Sets et supprimez ces éléments du premier Set

Exemple :

{23, 42, 65, 57, 78, 83, 29}
{57, 83, 29, 67, 73, 43, 48}

Résultat :

Intersection {57, 83, 29}
Set 1 après suppression : {65, 42, 78, 23}

SOLUTION :

```
def inter( listA, listB):  
    intersection= {n for n in listA if n in listB}  
    print( "l'intersection 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 :

5. 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

Exemple :

[47, 64, 69, 37, 76, 83, 95, 97]
{'Yassine':47, 'Imane':69, 'Mohammed':76, 'Abir':97}

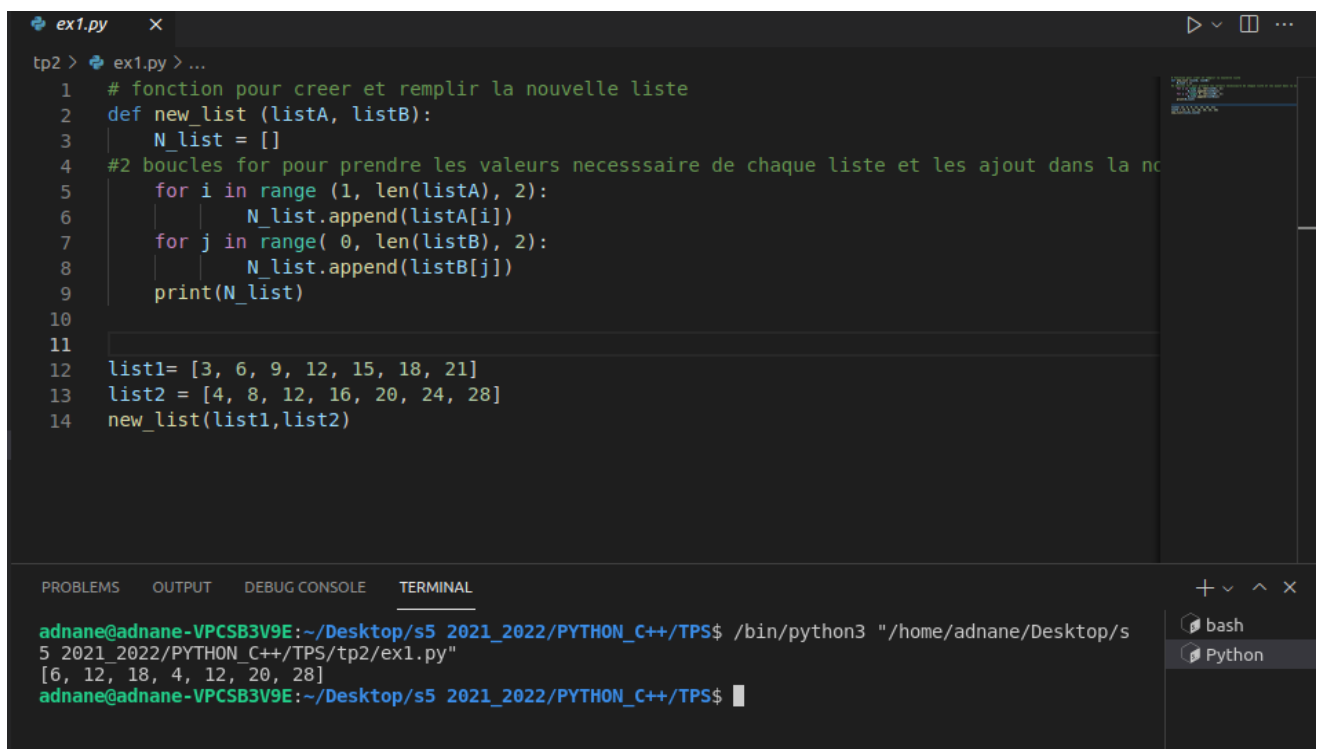
Résultat :

[47, 69, 76, 97]

SOLUTION :

```
def test_existence (dict, listA):  
    listB=[]  
    for i in listA:  
        if i in 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_existence(my_dict, list1))
```

2. code source et commentaires et sortie du programme:



```
ex1.py x
tp2 > ex1.py > ...
1  # fonction pour creer et remplir la nouvelle liste
2  def new_list (listA, listB):
3      N_list = []
4      #2 boucles for pour prendre les valeurs necessaire de chaque liste et les ajout dans la nc
5      for i in range (1, len(listA), 2):
6          N_list.append(listA[i])
7      for j in range( 0, len(listB), 2):
8          N_list.append(listB[j])
9      print(N_list)
10
11
12 list1= [3, 6, 9, 12, 15, 18, 21]
13 list2 = [4, 8, 12, 16, 20, 24, 28]
14 new_list(list1,list2)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
adnane@adnane-VPCSB3V9E:~/Desktop/s5_2021_2022/PYTHON_C++/TPS$ /bin/python3 "/home/adnane/Desktop/s
5_2021_2022/PYTHON_C++/TPS/tp2/ex1.py"
[6, 12, 18, 4, 12, 20, 28]
adnane@adnane-VPCSB3V9E:~/Desktop/s5_2021_2022/PYTHON_C++/TPS$
```

bash
Python

exercice 1 :

exercice 2 :

```
ex2.py x
tp2 > ex2.py > ...
1 #fonction pour deiviser une lsite en 3 morceau egaux
2 def divide_inverse(listA):
3     # division de la liste sur 3, en assurant que n soit interger pour lutiliser apres
4     n= len(listA)
5     n=int(n/3)
6     #declaration de la 1ere nouvelle liste et la remplir par les valeurs de la listeA de 0 a n
7     list1=[]
8     list1= listA[0:n]
9     list1.reverse()
10    print(list1)
11    #declaration de la 2eme nouvelle liste et la remplir par les valeurs de la listeA de n a -n
12    list2=[]
13    list2= listA[n:-n]
14    list2.reverse()
15    print(list2)
16    #declaration de la 3eme nouvelle liste et la remplir par les valeurs de la listeA de -n a 1
17    list3=[]
18    list3= listA[-n:]
19    list3.reverse()
20    print(list3)
21
22 list_a= [11, 45, 8, 23, 14, 12, 78, 45, 89,45,34,56]
23 divide_inverse(list_a)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
5 2021_2022/PYTHON_C++/TPS/tp2/ex2.py"
[23, 8, 45, 11]
[45, 78, 12, 14]
[56, 34, 45, 89]
adnane@adnane-VPCSB3V9E:~/Desktop/s5_2021_2022/PYTHON_C++/TPS$
```

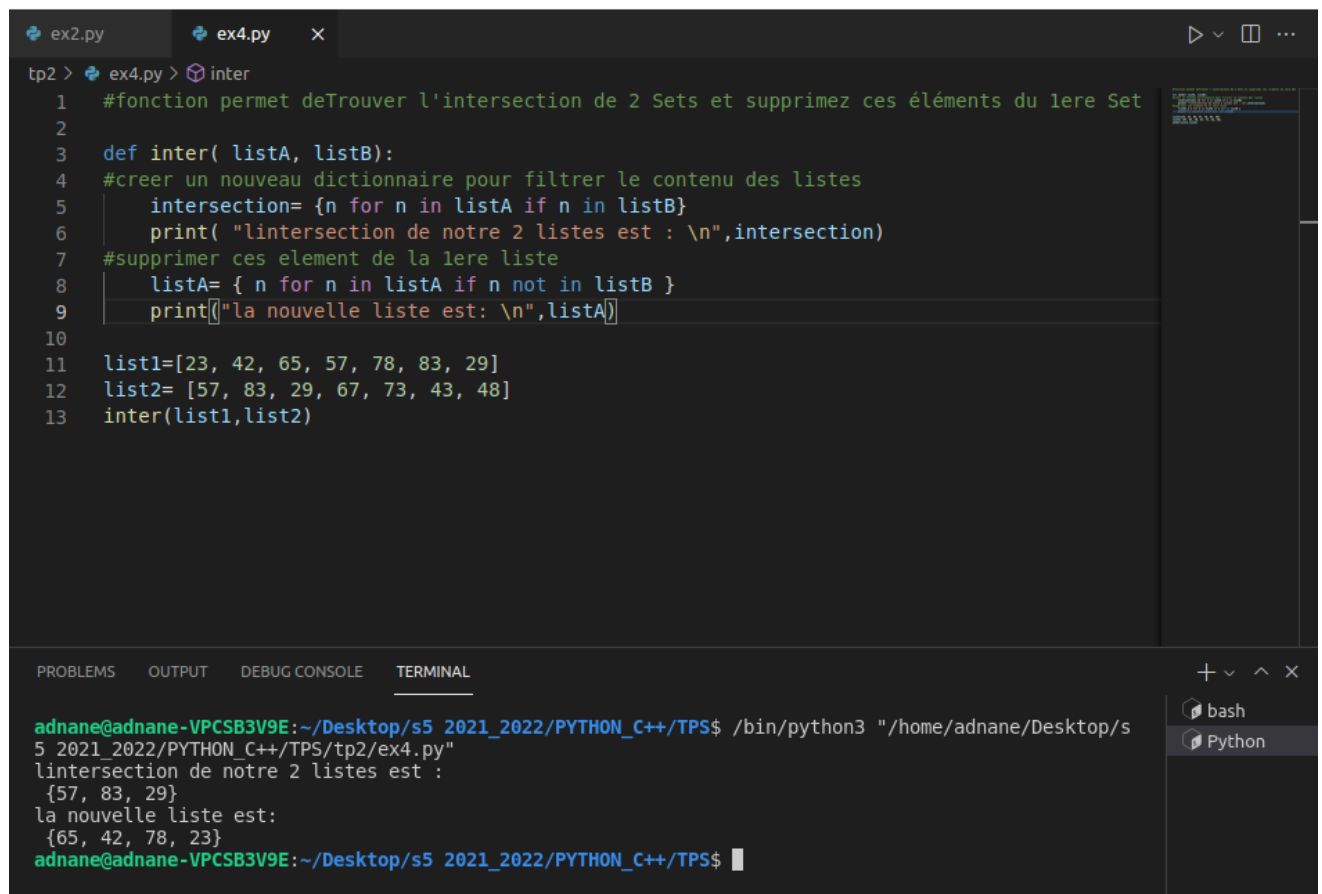
```
ex2.py ex3.py x
tp2 > ex3.py > ...
1 #fonction pour itérer une liste donnée et compter l'occurrence de chaque élément
2 def iterer(list):
3     my_dict={}
4     #boucle for pour iterer la fonction
5
6     for i in range(len(list)):
7         if list[i] in my_dict:
8             my_dict[list[i]]+= 1
9         else:
10            my_dict[list[i]]= 1
11
12    print(my_dict)
13
14    '''boucle if pour tester et calculer le nombre
15    d'occurrence de chaque variable, ou bien de creer un nouvel key dans le dictionner si
16    il n'existe pas déjà'''
17
18
19 list1= [11, 45, 8, 11, 23, 45, 23, 45, 89]
20 iterer(list1)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
adnane@adnane-VPCSB3V9E:~/Desktop/s5_2021_2022/PYTHON_C++/TPS$ /bin/python3 "/home/adnane/Desktop/s
5_2021_2022/PYTHON_C++/TPS/tp2/ex3.py"
{11: 2, 45: 3, 8: 1, 23: 2, 89: 1}
adnane@adnane-VPCSB3V9E:~/Desktop/s5_2021_2022/PYTHON_C++/TPS$
```

exercice 3 :

exercice 4 :



The image shows a Visual Studio Code editor window with two tabs: 'ex2.py' and 'ex4.py'. The 'ex4.py' tab is active, displaying a Python script. The script defines a function 'inter' that finds the intersection of two lists and removes elements from the first list. It then calls this function with two specific lists. Below the editor, the 'TERMINAL' panel shows the command to run the script and its output.

```
tp2 > ex4.py > inter
1  #fonction permet deTrouver l'intersection de 2 Sets et supprimez ces éléments du 1ere Set
2
3  def inter( listA, listB):
4      #creer un nouveau dictionnaire pour filtrer le contenu des listes
5          intersection= {n for n in listA if n in listB}
6          print( "l'intersection de notre 2 listes est : \n",intersection)
7      #supprimer ces element de la 1ere liste
8          listA= { n for n in listA if n not in listB }
9          print("la nouvelle liste est: \n",listA)
10
11 list1=[23, 42, 65, 57, 78, 83, 29]
12 list2= [57, 83, 29, 67, 73, 43, 48]
13 inter(list1,list2)
```

Terminal Output:

```
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$ /bin/python3 "/home/adnane/Desktop/s5 2021_2022/PYTHON_C++/TPS/tp2/ex4.py"
l'intersection de notre 2 listes est :
{57, 83, 29}
la nouvelle liste est:
{65, 42, 78, 23}
adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS$
```

exercice 5 :

ex2.py ex4.py ex5.py X

tp2 > ex5.py > test_existence

```
1 '''Itérer une liste donnée et vérifier si un élément donné existe en tant
2 que valeur de clé dans un dictionnaire. Sinon, supprimez-le de la liste
3 '''
4 def test_existence (dict, listA):
5     listB=[]
6     #boucle for pour tester tous les index de la liste
7     for i in listA:
8
9         if i in my_dict.values():
10             listB.append(i)
11     print(listB)
12
13 '''if: tester si les valeurs existe dans le dictionnaire
14 si oui, les ajoute dans la nouvelle liste'''
15
16 list1=[47, 64, 69, 37, 76, 83, 95, 97]
17 my_dict= {'Yassine':47, 'Imane':69, 'Mohammed':76, 'Abir':97}
18 print(test_existence(my_dict, list1))
19
20
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS\$ /bin/python3 "/home/adnane/Desktop/s5 2021_2022/PYTHON_C++/TPS/tp2/ex5.py"

[47, 69, 76, 97]

None

adnane@adnane-VPCSB3V9E:~/Desktop/s5 2021_2022/PYTHON_C++/TPS\$

+ ^ ^ X

bash

Python