

```
#####  
###---TP-1---###  
#####
```

#Exercice 3

```
##annee = 2025  
##print("votre annee de naissance:")  
##annee_naissance = int(input())  
##print("si cela et deja passer: 1 , sinon 0")  
##passege = int(input())  
##age = annee - annee_naissance - 1 + passege  
##  
##print("vous avez :" , age)
```

#Exercice 4

```
##age = int(input("veuillier saisir votre age:" ))  
##nom = str(input("votre prénom:"))  
##print(nom , " vous aurez" , age+5 , "dans 5 ans")
```

#Exercice 5

```
##farine = float(input("le nombre de kilo de farine :"))  
##kilo_total = farine * 2  
  
##chocolat = float(input("nombre de chocolat : "))  
##chocolat_total = chocolat * 2.5  
  
##resultat = print("votre prix est de", kilo_total + chocolat_total , " €")
```

#Exercice 6

```
##nombre_billet = int(input("combien de billet vous posseder :"))  
##  
##nombre_sac = nombre_billet//17  
##nombre_restant = nombre_billet%17  
##  
##print("voici le nombre de sac" , nombre_sac , "nombre billet restent" ,  
nombre_restant)
```

#Exercice 7

```
##euro = float(input("votre argent en euro:"))  
##dollar = euro + 0.09  
##  
##print("votre argent en dollar :" , dollar)  
  
##chapeaux = 3.99  
##nombre_chapeaux = dollar//chapeaux  
##  
##print("nombre de chapeau que on peut acheter :" , nombre_chapeaux)
```

#Exercice 8

```
##Hors_Taxe = float(input("Entrez le prix HT : "))  
##nombre_article = float(input("le nombre d'article : "))  
##TVA = float(input("Entrez le taux de TVA (%) : "))  
##  
##TTC = Hors_Taxe *(1 + TVA/100)  
##prix_article = nombre_article * TTC  
##  
##
```

```
##print("Le prix TTC est de : " , TTC , "le prix de l'article : " , prix_article)
```

```
#Exercice 9
```

```
##nombre_centime = int(input("combien de centime vous posseder :"))  
##  
##nombre_2 = nombre_centime//2  
##nombre_1 = nombre_centime%2  
##  
##print("voici le nombre de centime de 2:" , nombre_2 , "nombre de 1 en  
centime : " , nombre_1)
```

```
#####  
###---TP-2---###  
#####
```

```
#Exercice 2
```

```
##def double(n):  
##     nombre_double = n*2  
##     return nombre_double  
##  
##print(double(3))
```

```
#Exercice 3
```

```
##def moyenne(nb1, nb2 , nb3):  
##     moy = nb1 + nb2 + nb3 / nb3  
##  
##     return moy  
##  
##print(moyenne(10 , 9 , 11))
```

```
#Exercice 4
```

```
##def produit(nb1,nb2):  
##     return nb1*nb2  
##  
##print(produit(6,7))
```

```
#Exercice 5
```

```
##def echange(nb1 , nb2):  
##     return nb2 , nb1  
##  
##print(echange(12 , 98))
```

```
#Exercice 6
```

```
##import random as rd  
##  
##def deux_des():  
##     des_1 = rd.randint(1 , 6)  
##     des_2 = rd.randint(1 , 6)  
##     return des_1 , des_2  
##  
##print(deux_des())
```

```
#Exercice 7
```

```
##def conv_duree(s):  
##     heure = s // 3600
```

```

##     minute_1 = s - (heure * 3600)
##     minute_2 = minute_1 // 60
##     secondes = minute_1 -(minute_2 * 60)
##     return print( heure , " heure" , minute_2 , " minute", secondes , " secondes"
## )
##
## print(conv_duree(7422))

```

#Exercice 8

```

##a = int(input(" nombre de journaux :"))
##
##def prix(n):
##     prix_journaux = n*7
##     return prix_journaux
##
##
##
##def conversion(b):
##     livre = b // 240
##     reste = b % 240
##     sthilling = reste // 12
##     reste_sthilling = reste % 12
##     return print("livre = ", livre , "sthilling =", sthilling , "pences = ",
## reste_sthilling)
##
##
##
## print("le prix à payer pour les ", prix(a) ,"ce qui fait", conversion(a))

```

```

#####
###---TP-3---###
#####

```

#Exercice 2

```

##def maximum(a , b):
##     if a < b :
##         return b
##     else :
##         return a
##
## print(maximum(5 , 4))

```

#Exercice 3

```

##def exces_de_vitesse(vitesse):
##     if vitesse <= 130:
##         return "Limitation de vitesse respectée"
##     elif 130 < vitesse <= 150:
##         return "135 euros d'amende et 1 point"
##     else:
##         return "135 euros d'amende et 2 points"
##
## print(exces_de_vitesse(120))
## print(exces_de_vitesse(140))
## print(exces_de_vitesse(160))

```

#Exercice 4

```

##import random as rd
##
##def gain():

```

```

##     des = rd.randint(1 , 6)
##
##
##     if des < 6 and des % 2 == 0:
##         return print("vous gagner 3€ grace a votre dé :", des)
##     elif 6 :
##         return print("vous gagnez 4€ grace a votre dé :", des)
##     else :
##         return print("vous perdez 2€ a cause de votre dé :", des)
##
##print(gain())

```

#Exercice 5

```

##def prix(n):
##     if n <= 20:
##         photogopieuse = n * 0.20
##     else :
##         photogopieuse = 20 * 0.20 + ( n - 20 ) * 0.10
##     return photogopieuse
##
##print(prix(20))

```

#Exercice 6

```

##def prop_du_triangle(a ,b , c):
##     cotes = sorted([a, b, c])
##     if cotes[0]**2 + cotes[1]**2 == cotes[2]**2 :
##         if cotes[0] == cotes[1] or cotes[1] == cotes[2]:
##             return "Le triangle est rectangle et isocèle"
##         else:
##             return "Le triangle est rectangle"
##     else:
##         return "Le triangle n'est pas rectangle"
##
##print( prop_du_triangle(1, 2 ,3))

```

#Exercice 7

```

##for i in range(11):
##     print(i)

##for i in range(10):
##     if i%2 == 0:
##         print(i)

##for i in range(8 , 0 , -1):
##     print(i)

```

#Exercice 8

```

##import random as rd
##
##def gain(n):
##     gain_total = 0
##
##
##     for i in range(n):
##         des = rd.randint(1 , 6)
##         if des < 6 and des % 2 == 0:
##             gain_total += 3
##             print("vous gagner 3€ grace a votre dé :", des)
##         elif 6 :

```

```

##         gain_total += 4
##         print("vous gagnez 4€ grace a votre dé :", des)
##     else :
##         gain_total += 2
##         print("vous perdez 2€ a cause de votre dé :", des)
##
##     return print (gain_total , " €")
##
##print(gain(100))

```

#Exercice 9

```

##def maximun():
##    L = [1 , 5 , 3]
##    max_valeur = L[0]
##
##    for i in range(len(L)):
##        if L[i] > max_valeur :
##            max_valeur = L[i]
##            indice_max = i
##
##    return max_valeur , indice_max
##
##print(maximun())
##
##def indice_max():
##    L = [1 , 5 , 3 ]
##    max_valeur = L[0]
##
##    for i in range(len(L)):
##        if L[i] > max_valeur :
##            max_valeur = L[i]
##            indice_max = i
##
##    return indice_max
##
##print(indice_max())

```

```

#####
###---TP-4---###
#####

```

#Exercice 1

```

##for i in range(1 , 11):
##    print(i)
##
##for i in range(0 , 11 , 2):
##    print(i)
##
##for i in range(8 , 0 , -1):
##    print(i)

```

#Exercice 2

```

##def velo_club(n):
##    adherents = 220
##    for i in range(n):
##        adherents +=6
##    return adherents
##

```

```

##print(velo_club(1))

#Exercice 3

# regarde si la lettre e et dans le mot

##def myst(mot):
##    n = len(mot) # contient la chaine de caracter
##    for i in range(n):
##        if mot[i] == "e" :
##            return True
##    return False
##
##print(myst("mardi"))
##print(myst("vendredi"))

```

```

#Exercice 4

##import random as rd
##
##def combien_de_6(n):
##
##
##    nombre_6 = 0
##    for i in range(n):
##        de = rd.randint(1, 6)
##
##        if de == 6:
##            nombre_6 += 1
##            print("nombre " , de)
##
##        else :
##            print("nombre ", de)
##
##    return nombre_6
##
##print(combien_de_6(9))

```

```

#Exercice 5

##def liste_diviseurs(n):
##    for i in range(1 , n +1):
##        if n%i == 0 :
##            print(i)
##
##print(liste_diviseurs(12))

##def liste_diviseurs(n):
##    diviseurs = []
##    for i in range(1 , n +1):
##        if n%i == 0 :
##            diviseurs.append(i)
##
##    return print(diviseurs)
##
##
##print(liste_diviseurs(12))

```

```

#Exercice 6

##for i in range(11):
##    print(i , "x", i , "=" , i * i)
##    for k in range(11):
##        print(k , "x", i , "=" , k * i)

```

```

##      print()

#####
###---TP-5---###
#####

#Exercice 1

##import random as rd
##
##a = rd.randint(1,50)
##n = int (input("veulliez saisir une valeur :"))
##
##while n != a :
##    if n > a :
##        print("la valeur et trop grande")
##        n = int (input("veulliez saisir une valeur :"))
##    else :
##        print("la valeur et trop petit")
##        n = int (input("veulliez saisir une valeur :"))
##
##print("la valeur et bonne")

##import random as rd
##
##a = rd.randint(1,50)
##n = int (input("veulliez saisir une valeur :"))
##
##i = 1
##while n != a :
##    if n > a :
##        print("la valeur et trop grande")
##        n = int (input("veulliez saisir une valeur :"))
##    else :
##        print("la valeur et trop petit")
##        n = int (input("veulliez saisir une valeur :"))
##
##    i += 1
##
##print(i)
##print("la valeur et bonne")

#Exercice 2

##def velo_club(objet):
##    adherents = 220
##    n = 0
##    while adherents < objet :
##        adherents +=13
##        n += 1
##    return print(n , " ans")
##
##print(velo_club(270))

#Exercice 3

##def inflation():
##    prix = 500
##    i = 0
##    while prix <= 700 :
##        prix *= 1.08
##        i += 1
##

```

```

##    return print("prix de l'article :" , prix , "€ en " , i ,"ans")
##
##print(inflation())

```

#Exercice 4

```

##def prime_F(n):
##    prime = 0
##    for i in range(n):
##        if i < 10 :
##            prime = prime + 8000
##        else :
##            prime = prime + 11300
##    return prime
##
##def prime_G(n):
##    prime = 0
##    for i in range(n):
##        prime = prime + 10000
##    return prime
##
##def trouver_buts():
##    i = 1
##    while prime_G(i) > prime_F(i):
##        i += 1
##    return print(i)
##
##print(trouver_buts())

```

#Exercice 5

```

##def pgcd(n,m):
##    while m != 0:
##        n , m = m , n % m
##    return n
##
##print(pgcd(330,420))

```

```

#####
###---TP-6---###
#####

```

#Exercice 1

```

##def factorielle(n):
##    resultat = 1
##    for i in range(2, n+1):
##        resultat *= i
##    return resultat
##
##print(factorielle(6))

```

#Exercice 2

```

##def code_pin():
##
##    code = 8585

```



```

##     mdp = int(input("veullier saisir un code :"))
##     while code != mdp:
##         if code == mdp :
##             print("code trouver")
##         else :
##             print("code incorrect")
##             mdp = int(input("veullier saisir un nouveau code :"))
##     print("code trouver")
##
##
##print(code_pin())

```

#Exercice 3

```

##def somme_inverses(n):
##     somme = 0
##     for i in range(1 , n +1):
##         somme += 1 / i
##     return somme

```

```

##def seuil_inverses(A):
##     somme = 0
##     n = 0
##     while somme <= A:
##         n += 1
##         somme += 1 / n
##     return n
##
##
##print(seuil_inverses(8))

```

#Exercice 4

```

##def nb_bacteries(n):
##     population = 10000
##     for i in range(n):
##         population = population * 2 + 1000
##     return population
##
##print(nb_bacteries(12))

```

#Exercice 5

```

##def pgcd(n,m):
##     while m != 0:
##         n , m = m , n % m
##
##     return n
##
##print(pgcd(330,420))

##def nb_voyelles(mot):
##     voyelles = "aeiouyAEIOUY"
##     nb_voyelles = 0
##     mot_vide = ""
##
##     for i in mot:
##         if i in voyelles:
##             nb_voyelles += 1
##             mot_vide += i
##
##     return nb_voyelles , mot_vide
##
##print(nb_voyelles("mael")) #(2, 'ae')

```

#Exercice 6

```
##def myst():
##    chaine = input("Entrer une chaine de caractere :")
##    voyelles = "aeiouyAEIOUY"
##    nb_voyelles = 0
##
##    for i in chaine:
##        if i in voyelles: # verifie que i ( la lettre du mot ) et dans la
voyelle
##            nb_voyelles += 1
##
##    return nb_voyelles
##
##print(myst())
```

#Exercice 7

```
##//revoir\##
```

```
##def numero_lettre(lettre):
##    alphabet = "abcdefghijklmnopqrstuvwxyz"
##    lettre = lettre.lower() # Pour gérer majuscules et minuscules
##    return alphabet.index(lettre)
```

```
##def chiffrement_cesar(texte, decalage):
##    alphabet = "abcdefghijklmnopqrstuvwxyz"
##    texte = texte.lower()
##    resultat = ""
##
##    for i in texte:
##        if i in alphabet:
##            numero = numero_lettre(i)
##            nouveau_numero = (numero + decalage) % 26
##            resultat += alphabet[nouveau_numero]
##        else:
##            # On conserve les caractères non alphabétiques (espaces,
ponctuations)
##            resultat += i
##
##    return resultat
```

```
#####
###---TP-7---###
#####
```

#Exercice 2

```
##L = [1 , 2 , 3 , 8 , 8]
##
##def mystere(L , x):
##    n = len(L)
##    nombre = 0
##    for i in range(n):
##        if L[i] == x:
##            nombre += 1
##    return nombre
```

#Exercice 3

```
##L = [1 , 2 , 3 , 8 , 8]
##
##M = L[:]
##M[0] = 3
##
##print("M =" , M)
##print("L =" , L)
```

#Exercice 4

```
##L = [1 , 11 , 3 , 9 , 10]

##def max_liste(L):
##    M = L[0]
##
##    for i in range(len(L)):
##        if M < L[i]:
##            M = L[i]
##
##    return M

##def indice_max(L):
##    M = L[0]
##
##    compteur = 0
##
##    for i in range(len(L)):
##        if M < L[i]:
##            M = L[i]
##            compteur = i
##
##    return compteur
```

#Exercice 5

```
##import random as rd

##def somme(n):
##    M = 0
##
##
##    for i in range(len(L)):
##        M += L[i]
##
##    return M

##n = int(input("Entrez un entier : "))
##L = []
##
##for i in range(n):
##    nombre = rd.randint(1 , 10)
##    L.append(nombre)

##print(L)
##print(somme(L))
##print(max_liste(L))
##print(mystere(L ,8))
```

#Exercice 6

```

##//revoir\##

##L = [4 , 2 , 3 , 8 ]
##
##def tri_selection(L):
##    n = len(L)
##    for i in range(n - 1, 0, -1):
##
##        max_index = 0
##        for j in range(1, i + 1):
##            if L[j] > L[max_index]:
##                max_index = j
##
##        L[i], L[max_index] = L[max_index], L[i]
##    return L
##
##print(tri_selection(L))

```

#Exercice 7

```

##L = [4 , 2 , 3 , 8 ]
##import random as rd

##def est_triee(L):
##
##    for i in range(len(L) - 1):
##        if L[i] > L[i + 1]:
##            return False
##    return True

```

##//revoir\##

```

##def bogosort(L):
##
##    while not est_triee(L):
##        rd.shuffle(L)
##    return L
##
##print(bogosort(L))

```

```

#####
###---TP-8---###
#####

```

#Exercice 2

```

##mat = ([[1 , 2 , 3],
##        [4 , 5 , 6],
##        [7 , 8 , 9],])

##def trace(M):
##    res = 0
##
##    for i in range(len(M)):
##        res += M[i][i]
##
##    return res
##
##print(trace(mat))

```

#Exercice 3

```

##def identite(n):
##    mat = ((n , n))
##    for i in range(n):
##        mat = [i ,i] = 1
##
##    return mat
##
##print(identite(8))

##def identite(n):
##    mat = []
##    for i in range(n):
##        ligne = []
##        for j in range(n):
##            if i == j:
##                ligne.append(1)
##            else:
##                ligne.append(0)
##        mat.append(ligne)
##    return mat

#Exercice 4

##mat =([[1 , 2 , 3],
##        [4 , 5 , 6],
##        [7 , 8 , 9],])
##
##mat_1 =([[1 , 2 , 3],
##          [4 , 5 , 6],
##          [7 , 8 , 9],])
##
##def somme_mat(mat_1 , mat_2):
##    assert len(mat_1) == len(mat_2) and mat_1[0] == mat_2[0], "diffrent"
##
##    res = []
##    for i in range(len(mat_1)):
##        ligne = []
##        for j in range(len(mat_1[0])):
##            ligne.append(mat_1[i][j] + mat_2[i][j])
##        res.append(ligne)
##    return res
##
##print(somme_mat(mat_1 , mat))

```