

Correction Devoir Surveillé N°2

Durée : 1 h 30 min.

Exercice 1 : (2 points)

#Programme et Fonction Factorielle Récursivité

#Fonction Factorielle Récursivité

```
def Factorielle_Recursive(n):
```

```
    if n==0:
```

```
        return(1)
```

```
    else:
```

```
        return(n*Factorielle_Recursive(n-1))
```

#Programme Factorielle Récursivité

```
n=int(input('Donnez le nombre n:'))
```

```
print("la factorielle de %d est= %d"%(n,Factorielle_Recursive(n)))
```

Exercice 2 : (5 points)

#Programme Nombres Triangulaires & Pyramidaux

```
N=int(input("Donner le nombre d'éléments des tableaux T et P : N="))
```

#Saisie Nombres Triangulaires

```
T=[k*(k+1)//2 for k in range(1,N+1)]
```

```
P=N*[0]
```

```
print(T)
```

#Saisie Nombres Pyramidaux

```
p=0
```

```
for i in range (N):
```

```
    p=p+T[i]
```

```
    P[i]=p
```

```
print(P)
```

#Classement des Nombres Pyramidaux

```
j=N
```

```
for i in range(j):
```

```
    if P[i]%2==0:
```

```
        for j in range(N-1,i,-1):
```

```
            if P[j]%2!=0:
```

```
                P[i],P[j]=P[j],P[i]
```

```
            else:
```

```
                continue
```

```
print("P=",P)
```

Correction Devoir Surveillé N°2

Durée : 1 h 30 min.

Exercice 3 : (10 points)

A) Programme principale :

#Programme Tailles

```
N=int(input("Donner le nombre d'individus dans l'échantillon : N="))
```

#Saisie les tailles dans Tailles[N] & Calcul de la moyenne

```
Som=0
```

```
Tailles=N*[0]
```

```
for i in range(N):
```

```
    Tailles[i]=int(input("Donner la taille de la personne%d:"%(i+1)))
```

```
    Som=Som+Tailles[i]
```

```
Moy=Som/N
```

```
print(Tailles)
```

```
print("la taille moyenne est : %f cm"%(Moy))
```

```
Nbr_Inf_Moy=0
```

```
Indic_Inf_Moy=[]
```

```
Nbr_Sup_180=0
```

```
for i in range(N):
```

```
    if Tailles[i]<Moy:
```

```
        Nbr_Inf_Moy=Nbr_Inf_Moy+1
```

```
        Indic_Inf_Moy=Indic_Inf_Moy+[i]
```

```
    elif Tailles[i]>180:
```

```
        Nbr_Sup_180=Nbr_Sup_180+1
```

```
print("Le nombre de personne ayant une taille inferieure à la taille moyenne %f cm est %d"%(Moy,Nbr_Inf_Moy))
```

```
print("La liste des indices des personnes ayant une taille inferieure à la taille moyenne",Moy,"cm est:",Indic_Inf_Moy)
```

```
print("Le nombre de personne ayant une taille superieure à 180 cm est %d"%(Nbr_Sup_180))
```

#Tri à bulles de la liste Tailles[N] & Recherche

```
print(Tailles)
```

```
Tri_bulle(Tailles)
```

```
print(Tailles)
```

```
t=int(input("Donner la valeur à chercher dans la liste Tailles : t="))
```

```
if Recherche_dichotomique(Tailles,t)==0:
```

```
    print("la valeur %d ne se trouve pas dans la liste Tailles"%(t))
```

```
else:
```

```
    print("la valeur %d se trouve dans la liste Tailles dans la position%d"%(t,Recherche_dichotomique(Tailles,t)))
```

Correction Devoir Surveillé N°2

Durée : 1 h 30 min.

B) Module (Tri à bulles et recherche dichotomique) :

1)

Procédure du Trie à Bulles

```
def Tri_bulle(Tailles):  
    N=len(Tailles)  
    Indicateur=True  
    while Indicateur :  
        Indicateur=False  
        for i in range(N-1) :  
            if Tailles[i]>Tailles[i+1] :  
                Tailles[i],Tailles[i+1]=Tailles[i+1],Tailles[i]  
                Indicateur=True
```

2)

Fonction Recherche dichotomique

```
def Recherche_dichotomique(L,t):  
    N=len(L)  
    Inf=0  
    Sup=N-1  
    while Inf<=Sup:  
        Milieu=(Inf+Sup)//2  
        if t==L[Milieu]:  
            return(Milieu+1)  
        elif t>L[Milieu]:  
            Inf=Milieu+1  
        else:  
            Sup=Milieu-1  
    return(0)
```

Exercice 4 : (3 points)

#Programme Centre de Masse

```
P=[2*[0.] for i in range(2)]  
Mass=0  
S=0  
for i in range(2):  
    P[i][0]=float(input('Donner la masse du point %d'%(i+1)))  
    Mass=Mass+P[i][0]  
    P[i][1]=float(input('Donner l'abscisse du point %d'%(i+1)))  
    S=S+(P[i][0]*P[i][1])  
print("l'abscisse du centre de masse est :",S/Mass)
```