

CORRIGÉ DU TP 5

```
#####
#                CORRIGE DU TP 4
#####
```

```
from math import *
```

```
#####
# EXERCICE 1
```

```
# QUESTION 1
```

```
def AireD(R):
    return pi * R * R
```

```
# QUESTION 2
```

```
def VolCyl(R,h):
    return AireD(R) * h
```

```
#####
# EXERCICE 2
```

```
# Version 1
```

```
def Moyenne1(L):
    SommeVal =0
    for k in range(len(L)):
        SommeVal +=L[k] # synonyme de SommeVal = SommeVal + L[k]
    return SommeVal / len(L)
```

```
# Version 2
```

```
def Moyenne2(L):
    SommeVal =0
    for elem in L:
        SommeVal +=elem # synonyme de SommeVal = SommeVal + L[k]
    return SommeVal / len(L)
```

```
#####
# EXERCICE 4
```

```
# QUESTION 1, version 1
```

```
def MaxList1(L):
    Max =L[0]
    for k in range(len(L)):
        if L[k] > Max:
            Max =L[k]
    return Max
```

QUESTION 1, version 2

```
def MaxList2(L):
    Max =L[0]
    for elem in L:
        if elem > Max:
            Max =elem
    return Max
```

QUESTION 2

```
def Max_et_Ind(L):
    Max =L[0]
    Ind =0
    for k in range(len(L)):
        if L[k] > Max:
            Max =L[k]
            Ind =k
    return Max, Ind
```

EXERCICE 5

QUESTION 1

```
def Facto(n):
    Res =1
    for k in range(1,n+1):
        Res =Res * k
    return Res
```

QUESTION 2

```
def Somme(N):
    Somme =0
    for indice in range(0,N+1):
        Somme +=1/Facto(indice)
    return Somme
```

EXERCICE 6

QUESTION 1

```
def Polaire(x,y):
    Rayon =sqrt(x**2+y**2)
    if x !=0:
        Theta =atan(y/x)
        if x <0:
            Theta +=pi
    else:
        if y >0:
            Theta =pi/2
        else:
            Theta =-pi/2
    return(Rayon,Theta)
```

```
# QUESTION 2
```

```
def Carte(r,theta):  
    return (r*cos(theta), r*sin(theta))
```

```
*****  
# EXERCICE 7
```

```
# QUESTION 1
```

```
def TestP(n):  
    PREMIER =True  
    diviseur_potentiel =2  
    while (PREMIER ==True) and (diviseur_potentiel <=sqrt(n)):  
        if n % diviseur_potentiel ==0:  
            PREMIER =False  
            diviseur_potentiel +=1  
    if not PREMIER or n <2:  
        return ('nombre non premier')  
    else :  
        return ('nombre premier')
```

```
# QUESTION 2
```

```
def PGCD(a,b):  
    if a < b:  
        a,b =b,a  
    reste =a%b  
    while reste > 0:  
        a, b =b, reste  
        reste =a%b  
    return b
```