

Exercice 1

Qu'affichera le programme suivant ?

```
1 x=17
2 def f(x):
3     x=(x+3)%6
4     print("x=",x)
5
6 def g():
7     global x
8     x=(x+3)%6
9     print("x=",x)
10
11 def h(x):
12     x=(x+3)%6
13     print("x=",x)
14     return x
15
16 f(x)
17 print("x=",x)
18 g()
19 print("x=",x)
20 x=h(x)
21 print("x=",x)
```

```
1 x=2
2 x=17
3 x=2
4 x=2
5 x=5
6 x=5
```

Exercice 2

Dans tout cet exercice, il est interdit d'utiliser l'opérateur `**`.

1) Ecrire une fonction **Puissance** qui, étant donné deux nombres a et b , calcule a^b .

```
1 def Puissance(a,b):
2     p=1
3     for i in range(b):
4         p=p*a
5     return p
```

2) Ecrire une fonction **f_n** qui associe (x, y) à $(\sin(x^n), \cos(y^n))$

```
1 import math
2 def f_n(x,y,n):
3     return (Puissance(math.sin(x),n),Puissance(math.cos(x),n))
```

3) Ecrire une fonction **Affiche** qui affiche les valeurs de f_n en $(0,0)$, $(1,1)$, ..., $(10,10)$

```
1 def Affiche(n):
2     for i in range(0,11):
3         print(f_n(i,i,n))
```

Exercice 3

1) Ecrire une fonction **Fact** qui calcule la factorielle de n .

```
1 def Fact(n):
2     p=1
3     for i in range(1,n+1):
4         p=p*i
5     return p
```

2) Ecrire une fonction **Comb** qui calcule le nombre de combinaisons de p parmi n .

Rappelons que $C(p, n) = \frac{n!}{p!(n-p)!}$

```
1 import math
2 def Comb(p,n):
3     return Fact(n)/(Fact(p)*Fact(n-p))
```

3) Ecrire une fonction **Triangle_Pascal** étant donné le nombre de lignes n , affiche le triangle de Pascal en calculant les coefficients binomiaux.

Exemple : pour $n=5$

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

```
1
2 def Triangle_Pascal(n):
3     for i in range(n+1):
4         for j in range(i):
5             print(Comb(i,j),end=" ")
6     print()
```

Exercice 4

Dans cet exercice, les seules fonctions permises sont : len et range.

1) Écrire la fonction **minimum(L)** qui reçoit en paramètres une liste de nombres L. La fonction renvoie le plus petit élément de L.

```
1 def minimum(L):
2     m=L[0]
3     for i in range(1,len(L)):
4         if m>L[i]:
5             m=L[i]
6     return m
```

2) Écrire la fonction **moyenne(L)** qui reçoit en paramètres une liste de nombres L. La fonction renvoie la moyenne des nombres de L.

```
1 def moyenne(L):
2     S=0
3     for i in range(len(L)):
4         S=S+L[i]
5     return S/len(L)
```

3) Écrire la fonction **grands(L,x)** qui reçoit en paramètres une liste de nombres L, et un élément x de L. La fonction renvoie le nombre d'éléments de L qui sont supérieurs strictement à x.

```
1 def grands(L,x):
2     ctr=0
3     for i in range(len(L)):
4         if L[i]>x:
5             ctr=ctr+1
6     return ctr
```

4) Écrire la fonction **petits(L,x)** qui reçoit en paramètres une liste de nombres L, et un élément x de L. La fonction renvoie le nombre d'éléments de L qui sont inférieurs strictement à x.

```
1 def petits(L,x):
2     ctr=0
3     for i in range(len(L)):
4         if L[i]<x:
5             ctr=ctr+1
6     return ctr
```

5) Soit L est une liste de taille n qui contient des nombres, et m un élément de L. L'élément m est un médian de L, si les deux conditions suivantes sont vérifiées :

- ▷ Le nombre d'éléments de L, qui sont supérieurs strictement à m, est inférieur ou égale à $\frac{n}{2}$.
- ▷ Le nombre d'éléments de L, qui sont inférieurs strictement à m, est inférieur ou égale à $\frac{n}{2}$.

Exemple : On considère la liste $L = [25, 12, 6, 17, 3, 10, 20, 12, 15, 38]$, de taille $n = 10$. L'élément 12 est un médian de L, car :

- ▷ 3 éléments de L sont supérieurs strictement à 12, et $3 \leq \frac{n}{2}$.
- ▷ 5 éléments de L sont inférieurs strictement à 12, et $5 \leq \frac{n}{2}$.

Écrire la fonction **median(L)** qui reçoit en paramètre une liste de nombres L non vide, et qui renvoie un élément médian de la liste L.

```
1 def median(L):
2     n=len(L)
3     for i in range(n):
4         if grands(L,L[i])<=n//2 and petits(L,L[i])<=n//2 :
5             return L[i]
```