

TP4 : Fonctions - CORRECTION

Exercice 1 :

Définissez une fonction `volBoite(x1,x2,x3)` qui renvoie le volume d'une boîte parallélépipédique dont on fournit les trois dimensions `x1`, `x2`, `x3` en arguments.

Par exemple, l'exécution de l'instruction :

`print(volBoite(5.2, 7.7, 3.3))` doit donner le résultat : **132,132**.

Exercice 2 :

On souhaite dresser le tableau de valeurs sur $[0;10]$ avec un pas de 0,5 d'un polynôme

$$P(x) = a_0 + a_1x + \dots + a_nx_n .$$

a) Ecrire une fonction `ImagePoly(l,x)` qui renvoie l'image de `x` par le polynôme `P` dont les coefficients sont stockés dans la liste `l` (si $P(x) = 2 + 3x + x^2$ alors $l[0] = 2$, $l[1] = 3$ et $l[2] = 1$).

b) Ecrire le programme qui demande à l'utilisateur de saisir :

- le degré du polynôme `P`
- les coefficients du polynôme `P` et les stocke dans une liste `l`
- stocke le tableau de valeurs de `P`

```
def ImagePoly(l,x):
    """Calculer l'image de x par un polynôme P dont les coefficients sont contenus dans la liste l"""
    s=0
    for i in range(len(l)):
        s=s+l[i]*(x**i)
    return(s)

def TabVal(l):
    """créé un tableau de valeurs du polynome dont les coefficients sont dans l de 0 à 10 avec un pas de 0.5"""
    tab=[]
    i=0
    while i<=10:
        tab.append((i,ImagePoly(l,i)))
        i=i+0.5
    return(tab)

n=int(input('quel est le degré du polynôme ?'))
l=[]

"On crée la liste des coefficients"
for i in range(n+1):
    a=float(input())
    l.append(a)
print(l)

"On affiche le tableau de valeurs"
print(TabVal(l))
```

Exercice 3 :

Que fait la fonction suivante ?

```
def puissance_rec(x,n):  
•   if n==0:  
•       return 1  
•   else:  
•       return x*puissance(x,n-1)
```

1°) Saisir cette fonction et l'appeler pour différentes valeurs.

2°) La modifier pour faire apparaître chaque étape du calcul.

Remarque : une fonction qui comporte un appel à elle-même est dite **récursive**, autrement elle est dite **itérative**.

Exercice 4 : Ecrire une version itérative puis une version récursive d'une fonction qui calcule

la factorielle d'un entier : $n! = \begin{cases} 1 & \text{si } n = 0 \\ n \times (n-1)! & \text{si } n \geq 1 \end{cases}$.

```
def factIter(n):  
•   """Calcule n! de manière itérative"""  
•   p=1  
•   for i in range(2,n+1):  
•       p=p*i  
•   return p  
  
def factRec(n):  
•   """Calcule n! de manière récursive"""  
•   if n==0:  
•       return(1)  
•   else:  
•       return(n*factRec(n-1))  
  
• n=int(input())  
• print(factIter(n))  
• print(factRec(n))
```

Exercice 5 : On souhaite créer une liste contenant le nom, le prénom et une note sur 20 pour n élèves.

1°) Quel type de donnée est le plus adapté ?

2°) Ecrire une fonction **Liste(n)** qui demande à l'utilisateur de compléter une telle liste de taille n et renvoie cette liste.

3°) Ecrire une fonction **Tri_Note(l)** qui range la liste selon les notes croissantes.

4°) Tester cet algorithme pour 3 élèves.

```

def Liste(n):
    """Cr e une liste de 3 uplets de taille n"""
    l=[]
    for i in range(n):
        nom=str(input('quel est le nom de l' ve ?'))
        prenom=str(input('quel est son pr nom ?'))
        note=float(input('quelle est sa note ?'))
        l.append((nom,prenom,note))
    return(l)

def Tri_Note(l):
    """Trie la liste l de 3 uplets selon les notes croissantes"""
    for i in range(len(l)):
        for j in range(i+1,len(l)):
            if l[j][2]<l[i][2]:
                aux=l[i]
                l[i]=l[j]
                l[j]=aux
    return(l)

n=int(input('quel est le nombre d' ves ?'))
l=Liste(n)
print(l)
print(Tri_Note(l))

```