

Exercice 1

```
def fact(n):
    print("Appel fact(",n,")",sep="")
    if n== 0:
        return 1
    else:
        k = fact(n-1)
        print("fact(",n,")=",n,"*",k,sep="")
        return n * k
```

L'appel de fact(6) produit :

```
In[1]: fact(3)
Appel fact(3)
Appel fact(2)
Appel fact(1)
Appel fact(0)
fact(1)=1*1
fact(2)=2*1
fact(3)=3*2
Out[1]: 6
```

Exercice 2

Tri rapide

```
def Tri_rapide(T):
    N = len(T)
    if N <= 1: # Appel terminal
        return T
    e = T.pop(N//2) # Retirer l'element au milieu du tableau
    T1, T2 = [ ], [ ]
    for x in T: # Constitution de T1 et T2
        if x <= e:
            T1.append(x)
        else:
            T2.append(x)
    return Tri_rapide(T1) + [e] + Tri_rapide(T2) # Appel rec.
```

```
from random import random
L1 = [random() for i in range(100)] # liste de 100 nombres aléatoires
L2 = [random() for i in range(1000)] # liste de 1000 nombres aléatoires
from time import clock
print("Tri de 100 nombres aléatoires : ")
debut = clock()
tri_rapide(L1)
fin = clock()
print("Temps d'exécution :", fin-debut,"secondes")
print("Tri de 1000 nombres aléatoires : ")
debut = clock()
tri_rapide(L2)
fin = clock()
print("Temps d'exécution :", fin-debut,"secondes")
```

Exercice 3

```
def sedgeSort(T):
    N = len(T)
    if N <= 15: # Appel terminal
        return tri_insertion(T)
    e = T.pop(N//2) # Retirer l'element au milieu du tableau
    T1, T2 = [ ], [ ]
    for x in T: # Constitution de T1 et T2
        if x <= e:
            T1.append(x)
        else:
            T2.append(x)
    return sedgeSort(T1) + [e] + sedgeSort(T2) # Appel rec.
```