Corrigé du TP 7

```
##########################################################
#                 CORRIGE DU TP 7
##########################################################



#*************************************************************
# EXERCICE 1

# QUESTION 1

def sans_quatre(L):
    for compteur in range(len(L)):
        if L[compteur] ==4:
            L[compteur] =0
    return L

# QUESTION 2

def rotation(L):
    Lrot =len(L)*[0]
    for k in range(len(L)-1):
        Lrot[k+1] =L[k]
    Lrot[0] =L[len(L)-1]
    return Lrot

# QUESTION 3

def inverser(L):
    return L[::-1]

# QUESTION 4

def supprimer_rang(L,i):
    return L[:i] +L[i+1:]

# QUESTION 5

def supprimer_element(L,a):
    L2 =[]
    for bubulle in range(len(L)):
        if L[bubulle] !=a:
            L2 +=[L[bubulle]]
    return L2
```

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

# QUESTION 1

L1 = [2*schtroumpf for schtroumpf in range(1,11)]

# QUESTION 2

L2 = [pronom +'chat' for pronom in ['mon ','ton ','son ','leur ','le ','ce ']]

# QUESTION 3

L3 = [ entier % 2 for entier in range(8)]

#*********************************************************************
# EXERCICE 3

# QUESTION 1

def indices_max(L):
    max =L[0]
    list_ind = [0]
    for k in range(len(L)):
        if L[k] ==max:
            list_ind +=[k]
        elif L[k] >max:
            max =L[k]
            list_ind = [k]
    return (max, list_ind)

# QUESTION 2

def second_max(L):
    max1, max2 =L[0], L[1]
    if max1 <max2:
        max1, max2 =max2, max1
    for k in range(2,len(L)):
        if L[k] >max1:
            max1, max2 =L[k], max1
        elif L[k] >max2:
            max2 =L[k]
    return max2
```

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

# QUESTION 1

def suite(n):
    u =0
    L =[u]
    for bidule in range(n):
        u =2*u +1
        L +=[u]
    return L

# QUESTION 2

def somme_suite(N):
    Somme =0
    L = suite(N)
    for elem in L:
        Somme +=elem
    return Somme

#*****************************************************************
# EXERCICE 5

# QUESTION 1

def Temps(Tf, N):
    return [k*Tf/N for k in range(N+1)]

# QUESTION 2

from math import *

def Charge(E,tau,Tf,N):
    L =Temps(Tf,N)
    return [E*(1-exp(-L[k]/tau)) for k in range(N+1)]
```

```
#*****************************************************************
# EXERCICE 6

# 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 False
    else:
        return True

def facteurs_premiers(n):
    L =[]
    for  k in range(2,n+1):
        if  TestP(k) and n%k ==0:
            L =L +[k]
    return L

# QUESTION 2

def  liste_premiers (n):
    L =[]
    for  machin in range(2,n):
        if  TestP(machin):
            L +=[machin]
    return L
```