

```

## TP 16

## exo 1
# q1
# on constate que la liste [x0, x1, x2, ..., x(n-1)] correspond au nombre
#  $x_0 \cdot 10^{(n-1)} + x_1 \cdot 10^{(n-2)} + \dots + x_{(n-2)} \cdot 10^{*1} + x_{(n-1)} \cdot 10^{*0}$ 
def list_to_number(L):
    S = 0
    n = len(L)
    for k in range(n):
        S = S+L[k]*10**(n-1-k)
    return S

#deuxième méthode
def list_to_number_bis (L):
    S = 0
    n = len(L)
    for k in range(n):
        S = S+L[n-1-k]*10**(k)
    return S

# q2
def dernier(n):
    r = n%97
    return 97-r

# q3
def number_to_list(m):
    a = m%10 # chiffre des unités
    b = m//10 # chiffre des dizaines
    return [b,a]

# q4
def complete(L):
    n = list_to_number(L)
    m = dernier(n)
    M = number_to_list(m)
    return L + M

## exo 2
# q1
import random as rd
def liste_alea(n):
    L = [k for k in range(10)]
    return [rd.choice(L) for k in range(n)]

#ou bien
def liste_alea_bis(n):
    L = [k for k in range(10)]
    L_bis=[]
    for k in range (n):
        x=rd.choice(L)
        L_bis.append(x)
    return L_bis

# q2
def bonne_place(L,M):
    compt = 0
    for k in range(len(L)):
        if L[k]==M[k]:
            compt += 1
    return compt

# q3

```

```

def distrib(L):
    M=[]
    for k in range(10):
        comp=0
        for x in L:
            if x==k:
                comp+=1
        M.append(comp)
    return M
#ou bien
def distrib_bis(L):
    M = 10*[0] # M est initialisé avec des 0 partout
    for x in L: # pour chaque élément x de L
        M[x] = M[x]+1 # on compte 1 de plus dans le nombre de x, c'est-à-dire dans M[x]
    return M

# q4
def elem_commun(L,M):
    DL = distrib(L)
    DM = distrib(M)
    S = 0
    for k in range(10):
        S = S+min(DL[k],DM[k])
    return S

# q5
mystere = liste_alea(5)

def jeu(L):
    a = bonne_place(L,mystere)
    tot = elem_commun(L,mystere)
    b = tot - a
    return a,b

```