```python
## exo 1
# q1 et q2
def f(n):
    L=[]
    for k in range(1,n+1):
        L.append(1/k)
    return L

def g(n):
    return [1/k for k in range(1,n+1)]

# q3 et q4
def inverse(L):
    L_res = []
    for x in L:
        L_res.append(1/x)
    return L_res

def inverse_bis(L):
    return [1/x for x in L]

## exo 2
def moy(L):
    return sum(L)/len(L)

def ecart(L):
    m = moy(L)
    S = 0
    for x in L:
        S = S + (x-m)**2
    return (S/len(L))**(1/2)

L = [(2*k+1)**(1/2) for k in range(0,1012)]
# L vaut environ [1, 1.73, 2.23 ... , 44.97]
# moy(L) vaut environ 29.992
# ecart(L) vaut environ 10.603

## exo 3
# q1
def suite(n):
    L = []
    u = 5
    for k in range(n):
        L.append(u)
        u = (1+u**3)/(1+u**2)
    return L

# q2
# il faut mettre en abscisses les valeurs 1, 2, 3, ..., n

n=50
absi = range(1,n+1)
ordo = suite(n)
import matplotlib.pyplot as plt
plt.plot(absi,ordo,'o')
plt.show()

## q3
def suitev(n):
    L=[]
    S=0
    for k in range(1,n+1):
        S = S+1/(k**2)
        L.append(S)
    return L
```

```
n = 1000
absi = range(1,n+1)
ordo = suitev(n)
plt.plot(absi,ordo,'o')
plt.show()

# on constate sur le dessin que la suite (v_n) semble converger !

## exo 4
# q1
def tronque(L,m):
    L_res = []
    for x in L:
        if x<=m:
            L_res.append(x)
    return L_res

# q2
def select(L):
    L_res = []
    for x in L:
        if x%3 == 0:
            L_res.append(x)
    return L_res

## exo 5
# q1
def cle(L):
    S = sum(L)
    return S%10

# q2
def securise(L):
    M = [x for x in L] # pour copier la liste L et pouvoir modifier M sans modifier L
    s = cle(L)
    M.append(s)

# q3
# à essayer dans la console

# q4
def verifie(M):
    s_recu = M.pop() # on a donc enlevé son dernier terme à M, s_recu est la clé dans
le message reçu
    s_theorique = cle(M) # s_theorique est la clé qu'il devrait y avoir s'il n'y a pas
d'erreur de transmission
    return s_recu == s_theorique
```