Codes du TP9

```
import matplotlib.pyplot as plt
import numpy as np

#Exercice 1.1

def a(n):
    return (n**2+3+np.log(n))/(np.exp(-n)+np.sqrt(n**4+n**2))

def b(n):
    return np.sqrt(n**2+2*n)-np.sqrt(2*n**2+1)

def c(n):
    return np.log(n**2+np.sqrt(n)+1)-np.log(n-np.sqrt(n)+1)

x=np.arange(1,50)

plt.plot(x,a(x),'+')
plt.show()

plt.plot(x,b(x),'+')
plt.show()

plt.plot(x,c(x),'+')
plt.show()


#Exercice 1.2

def suite2(n):
    u=0
    for k in range(1,n+1):
        u=np.cos(np.exp(u))+u/2
    return u

def suite2vec(n):
    U=np.zeros(n+1)
    u=0
    U[0]=u
    for k in range(1,n+1):
        u=np.cos(np.exp(u))+u/2
        U[k]=u
    return U

abscisse=np.arange(0,51,1)
plt.plot(abscisse, suite2vec(50),'+')
plt.show()
```

```
def convergence2():
    n=0
    u=0
    v=np.cos(np.exp(u))+u/2
    while np.abs(u-v)>2*10**-3:
        n=n+1
        v=u
        u=np.cos(np.exp(u))+u/2
    return n

#Exercice 1.3

def suite3(n):
    u=0
    for k in range(1,n+1):
        u=u**2+1
    return u

def divergence3():
    n=0
    u=0
    while u<10**5:
        n=n+1
        u=u**2+1
    return n

#Exercice 1.4

def suite4(n):
    u=1
    for k in range(1,n+1):
        u=-2*u+3*k+2
    return u

def suite4vec(n):
    U=np.zeros(n+1)
    u=1
    U[0]=u
    for k in range(1,n+1):
        u=-2*u+3*k+2
        U[k]=u
    return U

def suite4bis(n):
    u=1
    for k in range(n):
        v=u-k-1/3
        u=-2*u+3*k+2
    return u,v
```

```
#Exercice 2.1

def somme(n):
    S=0
    for k in range(1,n+1):
        S=S+(2*k+1)/(k**2*(k+1)**2)
    return S

def T(n):
    return n*(n+2)/(n+1)**2

#Exercice 2.2

def calcul_pn(n):
    P=1
    for k in range(n):
        P=P*(1-k/365)
    return 1-P

def convergence6(x):
    n=1
    p=0
    while p< x:
        n=n+1
        p=1-(1-p)*(1-(n-1)/365)
    return n
```