

TP 6 : while, corrigé

Exo 1

q1

```
def log_3(n):  
    k = 0  
    while 3**k < n:  
        k += 1  
    return k
```

q2

```
def seuil_somme(M):  
    n = 0  
    S = 0  
    while S <= M:  
        n += 1  
        S = S + n/(n+4)  
    return n
```

q3

```
def seuil(M):  
    u = 1  
    n = 0  
    while u < M:  
        u = 15*u/(u+10)  
        n += 1  
    return n
```

Exo 2

q1

```
def suite(n):  
    u = 0  
    for k in range(n):  
        u = k*(-1)**k + u*(k-5)/(3*k+2)  
    return u
```

q2

```
def audela(seuil):  
    n = 0  
    while suite(n) < seuil :  
        n = n+1  
    return n
```

q3

```
def endessous(seuil):  
    n = 0  
    while suite(n) >= seuil:  
        n = n+1  
    return n
```

q4

il n'est pas judicieux d'utiliser la fonction suite
car elle reprend le calcul de u_n depuis le départ à chaque fois
on calcule plutôt les termes de la suite successivement

```
def audelabis(seuil):
    u = 0
    n = 0
    while u < seuil:
        u = n*(-1)**n + u*(n-5)/(3*n+2)
        n = n+1
    return n
```

```
def endessousbis(seuil):
    u = 0
    n = 0
    while u >= seuil :
        u = n*(-1)**n + u*(n-5)/(3*n+2)
        n = n+1
    return n
```

Exo 3

Question 1

On trouve la suite 3, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, ...
 # Quand on atteint la valeur 1 on boucle sur le cycle 4, 2, 1.

Question 2

```
def f(k) :
    if k%2==0 :
        return k//2
    else :
        return 3*k+1
```

Question 3

```
def syracuse(N,n):
    u = N
    for k in range(n):
        u = f(u)
    return u
```

Question 4

```
def TempsVol(N):
    u = N
    n = 0
    while u !=1 :
        u = f(u)
        n += 1
    return n
```

Question 5

```
import matplotlib.pyplot as plt

N_max = 1000
for N in range(1, N_max+1):
    plt.plot(N, TempsVol(N), 'bo')
plt.show()
```

Question 6

#Pour $N < 0$ on trouve d'autres cycles