

```

# tp4corr.py

## TP 4 info 2025-26

## exo 1

# q1
for k in range(6,12):
    print(k)

# q2
for k in range(101,122,2):
    print(k)

# q3
for k in range(20,-21,-5):
    print(k)

## exo 2

# q1
def somme1(n):
    S = 0
    for k in range(3,n+2):
        S = S + (2*k+1)**(1/2)
    return S
# on vérifie dans la console que somme1(4) renvoie
# racine(7)+racine(9)+racine(11)
# soit environ 8.962

# q2
def fact(n):
    P = 1
    for k in range(1,n+1):
        P = P*k
    return P
# on vérifie par exemple que fact(3) renvoie 6

# q3
def somme2(x,n):
    S = 0
    for k in range(n+1):
        S = S + x**k/fact(k)
    return S
# on vérifie par exemple que somme2(7,4)
# vaut environ 189.7

# q4
def suite1(n):
    u = 2
    for k in range(1,n+1):
        u = 2*u+1/u
    return u
# on vérifie dans la console que suite1(10)
# vaut environ 2379.37

```

```

# q5
def suite2(n):
    v = 1.4
    for k in range(2,n+1):
        v = v*(3-v)
    return v
# on vérifie dans la console que suite2(10)
# vaut environ 2.1626

## exo 3

# q1
def suite(n):
    u = 1
    for k in range(1,n+1):
        u = 3*u**(1/2)+1
    return u
# on vérifie par exemple que suite(0) vaut 1,
# suite(1) vaut 4, suite(2) vaut 7

# q2
def somme(n):
    S = 0
    for k in range(0,n+1):
        S = S + suite(k)
    return S
# on vérifie par exemple que somme(2) vaut
# u0 + u1 + u2 = 1 + 4 + 7 = 12

# q3
# Pour calculer somme(5), Python fait appel à
# suite(0), suite(1), suite(2), suite(3), suite(4) et suite(5).
# Or pour chacun de ces appels, la fonction suite
# recalcule les termes de la suite à partir de u0.
# Ainsi u1 a en fait été recalculé 5 fois !

# q4
# Il faut utiliser deux variables en parallèle :
# une pour u et une pour S.
# Une façon de procéder est la suivante :
def somme_bis(n):
    u = 1
    S = 0
    for k in range(n+1):
        S = S + u
        u = 3*u**(1/2)+1
    return S
# on teste dans la console que somme_bis(10)
# renvoie bien la même chose que somme(10)

# q5
import time
T0 = time.time()
somme(3000)
T1 = time.time()
temps_pour_somme = T1-T0

```

```
T0_bis = time.time()
somme_bis(3000)
T1_bis = time.time()
temps_pour_somme_bis = T1_bis-T0_bis
# on vérifie dans la console que
# temps_pour_somme_bis > temps_pour_somme
```

```
## exo 4
```

```
# q1
```

```
def divisible(a,b):
    r = a%b
    return r==0
```

```
# q2
```

```
def premier(n):
    for k in range(2,n):
        if divisible(n,k):
            return False
    return True
```

```
# q3
```

```
for n in range(2,101):
    if premier(n):
        print(n)
```