TD2.1bio2(24).py

```
001| ## Exo 1 : Suite géométrique
002| print("Exo 1")
003| n = eval(input("Entrez la valeur de l'entier naturel n : "))
004| q = eval(input("Entrez la valeur de q : "))
005| s = 0
006| for k in range(1,n+1):
007|     s += q**k
008| print("S{} = {}".format(n,s))
009|
010| ## Exo 2 : Suite géométrique bis
011| print()
012| print("Exo 2")
013| def somme(n,q):
014|     s = 0
015|     for k in range(1,n+1):
016|         s += q**k
017|     print(s)
018|
019| ## Exo 3 : Algorithme de Héron ou de Babylone
020| print()
021| print("Exo 3")
022| # conjecture : heron(x,precision) affiche une approximation
023| # de la racine carrée de x
024| def heron(x, precision):
025|     u = x
026|     v = (x+1)/2
027|     while abs(v-u) > precision:
028|         u, v = v, (v+x/v)/2
029|     print("Une valeur approchée de racine carrée de", x, "est", v)
030|
031| def heron_bis(x, precision):
032|     u = x
033|     v = (x+1)/2
034|     while abs(v-u) > precision:
035|         u, v = v, (v+x/v)/2
036|     return v
037|
038| ## Exo 4 : Factorielle
039| print()
040| print("Exo 4")
041| def facto(n):
042|     f = k = 1
043|     while True:
044|         f *= k
045|         if k >= n:
046|             break
047|         k += 1
048|     return f
049|
050| def facto_bis(n):
051|     f = k = 1
052|     while True:
053|         f *= k
054|         if k >= n:
055|             return f
056|         k += 1
057|
058| print("Ces deux fonctions sont des programmes pédagogiques.")
059|
```

```
print("Ceci dit, on déconseille fortement l'en-tête while True")

print("dans vos futurs programmes.")

## Exo 5
print()
print("Exo 5")
from math import *
n = eval(input("Entrez n pour obtenir u[n-1] : "))
u = eval(input("Entrez a : "))
v = eval(input("Entrez b : "))
for k in range(n):
    u, v = v, sqrt(u)+sqrt(v)
print(u)

print("Pour a=1, b=2 et n=1000 le programme affiche \
3.9999999999999996")

print("Pour a=1/2, b=1/5 et n=1000 le programme affiche \
également 3.9999999999999996")

print("On conjecture que la limite est 4 quel que soit (a,b)")
## Exo 6
print()
print("Exo 6 question 1")
def suiteu(n):
    u = 1
    for k in range(n):
        u = u + u**2
    return u

print("u[7] =",suiteu(7))
print("u[8] =",suiteu(8))

print()
print("Exo 6 question 2")
def suitev(n):
    return log(suiteu(n))/2**n

print("v[7] =",suitev(7))
print("v[8] =",suitev(8))

print("On conjecture que (vn) converge vers 0.4686966618056758...")

print()
print("Exo 6 question 3")
def limite(p):
    n, u1, u2 = 0, 1, 2
    while abs(log(u2)/2**(n+1) - log(u1)/2**n) > p:
        u1, u2 = u2, u2+u2**2
        n += 1
    return log(u2)/2**(n+1)

print("limite(1) =",limite(1))
print("limite(0.0001) =",limite(0.0001))

print()
print("Exo 6 question 4")
def comparaison(n,p):
    b = limite(p)
    u = suiteu(n)
```

```
121|     ex = exp(b*2**n)
122|     return u/ex, u-ex
123|
124| print()
125| print("Exo 6 question 5")
126| print("comparaison(6,1/10000) \
127| renvoie (1.0000003064249336, 3263441.994140625)")
128| print("On peut conjecturer que le premier terme de \
129| comparaison tend vers 1")
130| print("mais le second tend vers l'infini. \
131| La conjecture ne se confirme qu'à moitié.")
```