TD : While

1. Écrire un programme qui renvoie la plus petite valeur de n telle que la suite (u_n) est supérieure ou égale à 10 000, où la suite est définie par : $u_0 = 3$ et $\forall n \in \mathbb{N}$, $u_{n+1} = 2u_n + \ln(u_n)$.

2. Écrire un programme qui renvoie la plus petite valeur de n telle que la suite (u_n) est inférieure ou égale à 0,01, où la suite est définie par : $u_0 = 0{,}9$ et $\forall n \in \mathbb{N}$, $u_{n+1} = u_n^2$.

3. Écrire un programme qui renvoie la plus petite valeur de n telle que la suite (u_n) est supérieure strictement à 500, où : $u_0 = 1$ et $\forall n \in \mathbb{N}$, $u_{n+1} = (n+1)u_n^2$.

4. Écrire un programme qui renvoie la plus petite valeur de n pour laquelle la somme $\sum_{j=1}^{n} j + \ln(j)$ est supérieure ou égale à 100.

5. Écrire un programme qui détermine et affiche le plus petit entier naturel n tel que $n^2 > 6324$, en utilisant une boucle While.

6. Que se passe-t-il se on exécute le programme suivant ? Expliquer pourquoi.

```
u=50
n=0
while u<60:
    u=u*0.5
    n=n+1
print(n)
```

7. Déterminer par le calcul la valeur affichée par le programme suivant :

```
u=1
n=0
while u<1000:
    u=2*u
    n=n+1
print(n)
```

8. *(D'après EDHEC 2016 voie E)* On considère une suite $(u_n)_{n\in\mathbb{N}}$ qui vérifie :

$$\forall n \in \mathbb{N}, 0 \leqslant u_n - n \leqslant \mathrm{e}^{-\sqrt{n}}$$

   a) Compléter le script suivant pour qu'il affiche un entier n tel que un-n est inférieur ou égal à 0,0001.

```
n=0
while ... :
    n....
print(...)
```

On admet qu'une valeur approchée au dixième de ln(10) est 2,3.

b) Le programme suivant affiche l'une des valeurs de n : 55, 70 ou 85. Laquelle est-ce ?

9. *(D'après Ecricome 2017 voie S)* On considère les deux suites :
$\forall n \in \mathbb{N},\ u_n = \dfrac{(-1)^n}{(n+1)^{n+1}}$. $S_n = \sum_{k=0}^{n} u_k$
   a) Écrire un programme qui affiche la valeur de S100.
   b) On admet que : $\forall n \in \mathbb{N},\ |I - S_n| \leqslant \frac{1}{e^{n+1}(n+1)!}$ Où I est un réel que l'on ne connait pas.
   En déduire un programme permet de donner une valeur approchée de I à 0,01 près.

10. Soit la suite (S_n) définie sur N* par $S_n = \sum_{k=1}^{n} \frac{1}{k}$.
   a) Écrire un programme qui détermine et affiche le plus petit entier naturel n tel que $S_n > 5$.
   b) Décrire un procédé pour conjecturer la limite de la suite Sn.

11. Soit la suite (S_n) définie sur N* par $S_n = \sum_{k=1}^{n} \frac{1}{k^2}$.

a) Écrire un programme qui renvoie la plus petite valeur de n telle que Sn est supérieure ou égal à 2.

b) Lancer le programme. Que se passe-t-il ? Quelle conclusion en tirer sur la convergence de la suite (Sn) ?

c) Le programme suivant affiche 0.99939. Conjecturer.

```
import numpy as np
S=0
for k in range(1,1000):
  S=S+1/k**2
a=6*S/np.pi**2
print(a)
```

On rappelle que np.pi renvoie la valeur du nombre pi.

Corrigé

Exercice 1

```
from math import *          # pour utiliser log
U=3
n=0
while U<10000:
    U=2*U+log(U)
    n=n+1
print(n)
```

Exercice 2

```
U=0.9
n=0
while U>0.01:
    U=U**2
    n=n+1
print(n)
```

Exercice 3

```
U=1
n=0
while U<500:
    U=(n+1)*U**2
    n=n+1
print(n)
```

Exercice 4

```
from math import *
S=0
n=1
while S<100 :
    S=S+n+log(n)
    n+=1          #on augmente n car on est dans while et pas for !
print(n)          # on veut n et non pas la valeur de la somme !
```

Exercice 5

```
n =0
while n**2<=6324 :
    n=n+1
print(n)
```

Exercice 6

La boucle s'exécute à l'infini car la suite posée décroit vers 0 en partant de 50 (suite géométrique de raison 0,5) donc u n'est jamais supérieur ou égal à 60.

Exercice 7

On résout $2^n \geq 1000$, ce qui équivaut à $n \geq$ln(1000)/ln(2), donc n égal à partie entière de ln(1000)/ln(2) +1.

Exercice 8

a)

```
n=0
while exp(-sqrt(n))>0.001:
    n=n+1
print(n)
```

Dans cet exo nul besoin de calculer les termes de la suite !

Exercice 9

a)

```
S=0 ; n=0
for k in range(1000):
    S=S+(-1)**n/(n+1)**(n+1)
    n=n+1
print(S)
```

b)

```
S=0 ; n=0
while 1/e**(n+1)/factorial(n+1)>0.01:
    S=S+(-1)**n/(n+1)**(n+1)
    n=n+1
print(S)
```

Exercice 10

(a)

```
S=0
k=1
while S<=5 :
    S+=1/k
    k+=1
print(k)
```

(b) On peut remplacer 5 par des valeurs plus grandes et regarder si l'algorithme tourne en boucle en non. Ce procédé reste périlleux car la suite Sn tend vers + l'infini (Série de Riemann, nous verrons cela dans le chapitre série), mais très peu vite (on montrera dans le chapitre intégrale que Sn vaut à peu près ln(n)).

Exercice 11

a)

```
S=0
n=1
```

```
while S<=2 :
    S+=1/n**2
    n+=1
print(n)
```

b) Le programme tourne dans le vide. Ce qui suggère que Sn ne dépasse jamais 2 donc que la suites est majorée par un réel inférieur à 2, donc que la suite converge (elle est croissante).
c) Ainsi, il semble que Sn converge vers (pi**2)/6. (Résultat prouvé par Euler au XVIIIe siècle).