

TP5

BOUCLES WHILE

Corrigé

Exercice 1 Quelle est la valeur de la variable s à la suite de ces instructions ? On complètera le tableau. *Inutile de recopier ce programme sur l'ordinateur.*

```
s = 0
q = 21
i = 4
while q != 0 and i > 0:
    s = s + (q%2)*2**(4-i)
    q = q//2
    i = i-1
print(s)
```

s	q	i
0	21	4
1	10	3
1	5	2
5	2	1
5	1	0
$s = 5$		arrêt

Exercice 2 Exemple classique : "déterminer le premier n tel que ..."

On considère la suite $(u_n)_{n \in \mathbb{N}}$ définie par
$$\begin{cases} u_0 = 0 \\ \forall n \in \mathbb{N}, u_{n+1} = u_n - 3n \end{cases}$$

On peut montrer que $(u_n)_{n \in \mathbb{N}}$ est décroissante et tend vers $-\infty$.

- Déterminer le plus petit entier $n \in \mathbb{N}$ tel que $u_n \leq -900$

```
u = 0
n = 0
while u > -900 :
    u = u-3*n
    n = n+1
    # print("la valeur de u",n,"est", u) # pour vérification
print(n) # réponse 25
```

```
la valeur de u 1 est 0
la valeur de u 2 est -3
la valeur de u 3 est -9
la valeur de u 4 est -18
la valeur de u 5 est -30
la valeur de u 6 est -45
la valeur de u 7 est -63
la valeur de u 8 est -84
la valeur de u 9 est -108
la valeur de u 10 est -135
la valeur de u 11 est -165
la valeur de u 12 est -198
la valeur de u 13 est -234
la valeur de u 14 est -273
la valeur de u 15 est -315
la valeur de u 16 est -360
la valeur de u 17 est -408
la valeur de u 18 est -459
la valeur de u 19 est -513
la valeur de u 20 est -570
```

```

la valeur de u 21 est -630
la valeur de u 22 est -693
la valeur de u 23 est -759
la valeur de u 24 est -828
la valeur de u 25 est -900

```

u	n	Condition
$u = 0 = u_0$	$n = 0$	vraie
$u = u_0 - 3 \times 0 = 0 = u_1$	$n = 1$	vraie
$u = u_1 - 3 \times 1 = -3 = u_2$	$n = 2$	vraie
...
Fin : $u = u_N$	$n = N$	Fausse, c'est-à-dire $u_N \leq -900$

La réponse est bien la valeur finale de n .

Exercice 3 Transformer le programme suivant pour utiliser une boucle while.

```

for k in range(1,11) :
    print( k**2 )

```

```

k=1
while k <= 10 :
    print(k**2)
    k=k+1

```

La boucle for est ici plus pertinente (plus concise, plus claire).

Exercice 4 Soit $v_1 = 2$ et $v_{n+1} = \exp\left(\frac{v_n}{n+1}\right)$ pour $n \geq 1$.

- Écrire une fonction `suite_v(n)` qui, étant donné n , renvoie la valeur de v_n .

```

def suite_v(n):
    v = 2
    for k in range(1,n):
        v = exp(v/(k+1))
    return v
# ou alors
def suite_v(n):
    v = 2
    for i in range(2,n+1):
        v = exp(v/i)
    return v

```

- Déterminer le plus grand $n \in \mathbb{N}$ tel que $v_n \geq 1,001$.

```

n = 1
v = 2
while v >= 1.001:
    v = exp(v/(n+1))
    n = n+1
print(n-1)
# attention, avant dernière valeur de n
# réponse : 1001

```

```

# deuxième solution
n = 1
v = 2
while v >= 1.001:
    n = n+1
    v = exp(v/n)
print(n-1)

# troisième solution, moins efficace en termes de temps de calcul
n = 1
while suite_v(n) >= 1.001:
    n = n+1
print(n-1)

```

v	n	Condition
$v = 2 = v_1$	$n = 1$	vraie
$v = \exp(v_1/2) = v_2$	$n = 2$	vraie
...
$v = v_{n-1}$	$n - 1$	vraie, donc $v_{n-1} \geq 1,001$
Fin : $v = v_n$	n	fausse, donc $v_n < 1,001$

La réponse est bien l'avant dernière valeur de n .

3. Comment vérifier à l'aide de la fonction `suite_v(n)` ?

La réponse est 1001. On vérifie que `suite_v(1001) >= 1.001` et que `suite_v(1002) < 1.001`

Exercice 5 Suite de Fibonacci. On souhaite modéliser une population de lapins. Soit L_n le nombre de couples de lapins au n -ième mois.

On suppose que $L_0 = 1$, $L_1 = 1$ et que pour $n \in \mathbb{N}$, $L_{n+2} = L_{n+1} + L_n$.

Déterminer au bout de combien de mois la population totale dépassera le million de lapins.

```

n = 0
L = 1 #L0
M = 1 #L1
while L <= 1000000/2 : # attention, L = couples de lapins
    temp = L # on stocke la valeur de L
    L = M
    M = M + temp
    n = n+1
print(n)
# Réponse : 28

```

Exercice 6 Jeu du nombre mystère

Écrire un programme qui définit un nombre mystère aléatoire entre 1 et 100, puis demande à l'utilisateur de le deviner. Ce programme doit demander de saisir un nombre tant que le résultat n'a pas été trouvé et, à chaque étape, indiquera si le nombre mystère est plus grand ou plus petit que la proposition donnée.

```

from random import randint
mystere = randint(1,100)
devine = int(input("Quel est le nombre mystère entre 1 et 100 ?"))

while devine != mystere :
    if devine > mystere:
        devine = int(input("Ton nombre est trop grand, retente ta chance :
"))
    else:
        devine = int(input("Ton nombre est trop petit, retente ta chance :"))
print("Gagné ! Le nombre était", mystere)

>>> (executing lines 28 to 36 of "TP04-While.py")
quel est le nombre mystere entre 1 et 100 ?12
ton nombre est trop petit, retente ta chance 19
ton nombre est trop petit, retente ta chance 56
ton nombre est trop petit, retente ta chance 60
ton nombre est trop petit, retente ta chance 80
ton nombre est trop grand, retente ta chance 70
ton nombre est trop petit, retente ta chance 75
ton nombre est trop petit, retente ta chance 78
ton nombre est trop petit, retente ta chance 79
Gagné ! Le nombre était 79

# Autre proposition
mystere = randint(1,100)
jeu = 0
devine = int(input("quel est le nombre mystere entre 1 et 100 ?"))
while jeu == 0 :
    if devine > mystere:
        devine = int(input("ton nombre est trop grand, retente ta chance "))
    elif devine < mystere:
        devine = int(input("ton nombre est trop petit, retente ta chance "))
    else:
        jeu = 1
        print("tu as gagné, bravo")

>>> (executing lines 28 to 41 of "TP04-While.py")
quel est le nombre mystere entre 1 et 100 ? 10
ton nombre est trop petit, retente ta chance 20
ton nombre est trop petit, retente ta chance 30
ton nombre est trop grand, retente ta chance 25
ton nombre est trop grand, retente ta chance 24
tu as gagné, bravo

```