

Exercice 1

Ecrire un programme qui demande à l'utilisateur d'entrer un entier n , puis affiche cet entier n et recommence jusqu'à ce que l'utilisateur saisisse la valeur 0.

Exercice 2

Si n est un entier strictement positif, on note $image(n)$ le nombre entier tel que :

- si n est pair : $image(n) = \frac{n}{2}$
- si n est impair : $image(n) = 3 \times n + 1$

Par exemple, $image(4) = 2$ puisque 4 est pair ; et $image(3) = 10$ puisque 3 est impair et $3 \times 3 + 1 = 10$

a) programmer en python la fonction $image(n)$

b) écrire un programme qui :

- demande à l'utilisateur de saisir un nombre entier n strictement positif.
- Affiche l'image de n , puis l'image de l'image de n , et ainsi de suite, jusqu'à ce que la valeur affichée soit 1.

c) tester ce programme avec différentes valeurs de n .

Exercice 3

1) En utilisant l'exercice 2, écrire un programme qui fait la même chose que dans l'exercice 2, en affichant en plus :

- le nombre d'étapes pour arriver au nombre 1
- la valeur maximale qui a été rencontrée en chemin.

Pour le nombre de départ 7, le nombre d'étapes doit être 16 et le nombre maximal obtenu doit être 52.

2) parmi tous les entiers de 1 à 20,

- quel est l'entier de départ qui donne le nombre maximal d'étapes pour arriver à la valeur 1
- quel est l'entier de départ qui donne la plus grande valeur maximale rencontrée en chemin.

Exercice 4 : devine le nombre

Devine le nombre est un jeu dont le but est de deviner un nombre secret.

A chaque tour, le joueur propose une valeur et le maître du jeu (ici l'ordinateur) indique si le nombre secret est plus petit, plus grand, ou si la partie est gagnée.

Par exemple, si le nombre secret est 5, une partie pourra se dérouler ainsi :

Devine le nombre !

8

plus petit

4

plus grand

6

plus petit

5

gagné au bout de 4 tours !

1) Ecrire un programme qui réalise ce jeu, avec un nombre secret choisi au hasard entre 1 et 100.

2) tester le programme !

3) Modifier le programme pour limiter le nombre de tours, par exemple à 8 tours au maximum.

Exercice 5 Pour chacun des programmes suivants :

- Déterminer une valeur de n pour laquelle le programme termine
- Déterminer une valeur de n pour laquelle le programme ne termine pas

```
#1
n = int(input("Entrer n :"))
while n != 0:
    n = n-1
    print(n)
print('Fin')
```

```
#2
n = int(input("Entrer n :"))
while n<0:
    n = int(input("Entrer n :"))
while n != 0:
    n = n-2
    print(n)
print('Fin')
```

```
#3
n = int(input("Entrer n :"))
while n != 0:
    n = n // 2
    print(n)
print('Fin')
```

```
#4
t = [10,20,30,40]
n = int(input("Entrer n :"))
i=0
while i < len(t):
    if n == t[i]:
        print(n, 'est une valeur du tableau', t)
```

Exercice 6

On prétend que les deux programmes suivants effectuent la multiplication de a par b, si a et b sont des entiers positifs. Ces programmes terminent-ils ? Identifier, pour chacun d'eux, un variant de boucle.

```
#1
a = int(input("a ="))
b = int(input("b ="))
m = 0
while b > 0:
    m += a
    b -= 1
print("a * b =", m)
```

```
#2
a = int(input("a ="))
b = int(input("b ="))
i=0
m = 0
while i < b:
    m += a
    i += 1
print("a * b =", m)
```

Exercice 7

On prétend que les deux programmes suivants calculent le nombre de chiffres d'un entier saisi par l'utilisateur : ces programmes terminent-ils ? Identifier, pour chacun d'eux, un variant de boucle.

```
#1
n = int(input("Entrer un entier strictement positif:"))
i = 0
while n > 0:
    n = n // 10
    i += 1
print("nombre de chiffres :", i)
```

```
#2
n = int(input("Entrer un entier strictement positif:"))
i = 0
k = 1
while k <= n:
    k = k*10
    i += 1
print("nombre de chiffres :", i)
```

Exercice 8

On donne la définition d'une fonction `mystere` :

```
def mystere(t, v):  
    i = 0  
    while i < len(t) and t[i] != v :  
        i += 1  
    return i == len(t)
```

1) Compléter le tableau de suivi de l'exécution de : `mystere([10,20,30], 20)`

t						
v						
i						
i < len(t)						
t[i] != v						
i < len(t) and t[i] != v						

Quelle est la valeur **renvoyée** par : `mystere([10,20,30], 20)` ?

2) Compléter le tableau de suivi de l'exécution de : `mystere([10,20,30], 40)`

t						
v						
i						
i < len(t)						
t[i] != v						
i < len(t) and t[i] != v						

Quelle est la valeur **renvoyée** par : `mystere([10,20,30], 40)` ?

3) Justifier que la fonction termine en déterminant un variant de boucle.

Remarque : les deux codes suivants ne sont pas équivalents :

```
while i < len(t) and t[i] != v :  
while t[i] != v and i < len(t) :
```

```
>>> False and x
```

```
>>> True or x
```

```
>>> True and "Autre"
```

```
>>> False or "Expression"
```