Correction « WHILE »

SNT - Python

M.SALAH

a. Suite de Syracuse pour $n = 10$

En suivant la règle donnée : 10 → 5 → 16 → 8 → 4 → 2 → 1

Longueur du vol : 6 étapes avant d'atteindre 1.

Hauteur du vol : 16 (c'est le maximum atteint dans la suite).

b. Suite de Syracuse pour $n = 28$

En suivant les opérations : 28 → 14 → 7 → 22 → 11 → 34 → 17 → 52 → 26 → 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1

Longueur du vol : 18 étapes avant d'atteindre 1.

Hauteur du vol : 52 (c'est le maximum atteint).

b. Suite de Syracuse pour $n = 38$

En suivant les opérations : 38 → 19 → 58 → 29 → 88 → 44 → 22 → 11 → 34 → 17 → 52 → 26 → 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1

Longueur du vol : 21 étapes avant d'atteindre 1.

Hauteur du vol : 88 (c'est le maximum atteint).

2. Conjecture :

La conjecture de Syracuse (aussi appelée conjecture de Collatz) est la suivante :

Quelle que soit la valeur initiale n, la suite finit toujours par atteindre la valeur 1.

3. Tentative de démonstration :

Il est difficile de démontrer cette conjecture, mais voici une explication simple pour illustrer le raisonnement :

On remarque que lorsque n n est pair, la valeur diminue rapidement ($n \rightarrow n / 2$).

Lorsque n n est impair, le passage à $3 n + 1$ peut faire augmenter la valeur, mais elle redescend souvent rapidement après quelques étapes.

II. Script Python pour modéliser la suite de Syracuse :

a. Script 1 : Résultat

1

2

4

8

16

32

64

b. Script 2 : Résultat

50 et 99

75 et 98

87 et 97

93 et 96

96 et 95

- `def syrac(n):` qui renvoie l'entier obtenu en transformant (une fois) l'entier passé
- `def long_vol(n):` qui renvoie la longueur du vol pour l'entier passé en argument.

```
1  def syrac(n):
2      if n%2==0:
3          n=n//2
4      else:
5          n= ...
6      return n
7
8  def long_vol(n):
9      long = 0
10     while n!=...:
11         n = syrac(n)
12         long = ...
13     return long
```

- `def syrac(n):` qui renvoie l'entier obtenu en transformant (une fois) l'entier passé
- `def long_vol(n):` qui renvoie la longueur du vol pour l'entier passé en argument.

```
1  def syrac(n):
2      if n%2==0:
3          n=n//2
4      else:
5          n= 3 * n + 1
6      return n
7
8  def long_vol(n):
9      long = 0
10     while n!=...:
11         n = syrac(n)
12         long = ...
13     return long
```

- `def syrac(n):` qui renvoie l'entier obtenu en transformant (une fois) l'entier passé
- `def long_vol(n):` qui renvoie la longueur du vol pour l'entier passé en argument.

```
1  def syrac(n):
2      if n%2==0:
3          n=n//2
4      else:
5          n= 3 * n + 1
6      return n
7
8  def long_vol(n):
9      long = 0
10     while n!=1:
11         n = syrac(n)
12         long = ...
13     return long
```

- `def syrac(n):` qui renvoie l'entier obtenu en transformant (une fois) l'entier passé
- `def long_vol(n):` qui renvoie la longueur du vol pour l'entier passé en argument.

```
1  def syrac(n):
2      if n%2==0:
3          n=n//2
4      else:
5          n= 3 * n + 1
6      return n
7
8  def long_vol(n):
9      long = 0
10     while n!=1:
11         n = syrac(n)
12         long += 1
13     return long
```