[19]:
```python
import doctest
doctest.testmod()

def syracuse(u_n):
    """
    Affiche les termes de la suite de Syracuse.

    exemple :
    >>> syracuse(5)
    5
    16
    8
    4
    2
    1
    """
    print(u_n)
    if u_n > 1:
        if u_n % 2 == 0:
            syracuse(u_n//2)
        else:
            syracuse(3*u_n+1)
```

```
5
16
8
4
2
1
```





## 1.2 — Activités

**ACTIVITÉ**

**Donner** une définition récursive qui correspond au calcul de la fonction factorielle $n!$ définie par :

$$n! = \begin{cases} 1 & \text{si } n = 0 \\ 1 \times 2 \times \ldots \times n & \text{si } n > 0 \end{cases}$$

**Donner** d'une fonction `fact(n)` qui implémente cette définition.

**CORRECTION**

La fonction mathématique est :

$$n! = \begin{cases} 1 & \text{si } n = 0 \\ n \times (n-1)! & \text{si } n > 0 \end{cases}$$

**CORRECTION**

[1]:
```python
import doctest
doctest.testmod()

def fact(n):
    """
    Calcule le n factoriel, c'est-à-dire :
    n × (n-1) × ... × 2 × 1

    exemple:
    >>> fact(0)
    1
    >>> fact(5)
    120
    """
    if n==0:
        return 1
    else:
        return n * fact(n-1)
```





**ACTIVITÉ**

Soit $u_n$ la suite d'entiers définie par :

$$u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ est pair,} \\ 3 \times u_n + 1 & \text{sinon.} \end{cases}$$

avec $u_0$ un entier plus grand que 1.

Écrire une fonction récursive `syracuse(u_n)` qui affiche les valeurs successives de la suite $u_n$ **tant que** $u_n$ **est plus grand que 1**.

**REMARQUE**

La conjecture de Syracuse affirme que, quelle que soit la valeur de $u_0$, il existe toujours un indice $n$ dans la suite tel que $u_n = 1$. Cette conjecture défie toujours les mathématiciens.

**CORRECTION**