

**Exercice 1 :** La somme des entiers s'écrit :

$$0 + 1 + 2 + \dots + n$$

1. Donner une définition récursive de la somme des entiers.
2. Implémenter la fonction **somme**(**n : int**) → **int**.

**Exercice 2 :** La fonction factorielle est définie par :

$$n! = 1 \times 2 \times 3 \dots \times n \quad \text{si} \quad n > 0 \quad \text{et} \quad 0! = 1$$

1. Donner une définition récursive qui correspond au calcul de la fonction factorielle.
2. Implémenter la fonction **factorielle**(**n : int**) → **int**.

**Exercice 3 :** Soit  $u_n$  la suite d'entiers définie par  $u_0 > 1$  et :

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

Écrire la fonction *syracuse*(*u:int*) → *None* qui affiche les valeurs successives de la suite  $u_n$  tant que  $u_n > 1$ . L'appel de la fonction s'effectuera avec une valeur de  $u_0$  quelconque.

**Exercice 4 :**

1. Écrire une fonction récursive **entiers**(**i : int, k : int**) → **None** qui affiche les entiers entre i et k. Par exemple, **entiers**(0,3) doit afficher 0 1 2 3.
2. Écrire une fonction récursive **impairs**(**i : int, k : int**) → **None** qui affiche les nombres impairs entre i et k.

**Exercice 5 :** Écrire la fonction récursive **pgcd**(**a : int, b : int**) → **int** qui renvoie le Plus Grand Commun Diviseur de a et b. On donne comme précondition :  $a < b$ .

méthode d'Euclide : (20,35)

$$\begin{aligned} 35 &= \overbrace{20}^{a \rightarrow b} \times 1 + \overbrace{15}^{b \% a \rightarrow a} \\ 20 &= 15 \times 1 + 5 \\ 15 &= 5 \times 3 + 0 \\ \text{pgcd} &= 5 \end{aligned}$$

**Exercice 6 :** Écrire une fonction récursive **nombre\_chiffres**(**n : int**) → **int** qui renvoie le nombre de chiffres qui compose n.

**Exercice 7 :** La formulation récursive ci-après permet de calculer les coefficients binomiaux :

$$C(n, p) = \begin{cases} 1 & \text{si } p = 0 \text{ ou } n = p \\ C(n-1, p-1) + C(n-1, p) & \text{sinon} \end{cases}$$

1. Écrire une fonction récursive **C**(**n : int, p : int**) → **int** qui renvoie la valeur de  $C(n, p)$ .
2. Le triangle de Pascal est une présentation des coefficients binomiaux sous la forme d'un triangle. Dessiner le triangle de Pascal à l'aide d'une double boucle *for* pour n variant de 0 à 10.

Pour ceux qui ont fini : palindrome

```
1 def palindrome(s):  
2     if len(s)<2:  
3         return True  
4     elif not(s[0]==s[-1]):  
5         return False  
6     else:  
7         return palindrome(s[1:-1])  
8  
9 print(palindrome("ressasser"))  
10 print(palindrome("ressadsser"))
```

tour de hanoi pour les plus avancés