

## OBJECTIFS

L'objectif de ce TP est d'étudier plusieurs techniques de calculs de coefficients binomiaux et de les comparer entre elles.

- Écrire une fonction récursive telle que l'exécution de la commande `CB1(k, n)` renvoie la valeur de  $\binom{n}{k}$ .

```
1 def CB1(k, n) :
2     if k > n :
3         return 0
4     elif k == 0 :
5         return 1
6     else :
7         return CB1(k-1, n-1) + CB1(k, n-1)
```

- En utilisant l'expression des coefficients binomiaux à l'aide des factorielles, écrire alors une fonction telle que l'exécution de la commande `CB2(k, n)` renvoie la valeur de  $\binom{n}{k}$ .

```
1 from math import factorial
2
3 def CB2(k, n) :
4     return factorial(n) / (factorial(k) * factorial(n-k))
```

### ♣ INDICATION...

La commande `factorial()` de la bibliothèque `math` permet de calculer des factorielles.

- Vérifier :  $\forall n \in \mathbb{N}^*, \forall k \in \llbracket 1; n \rrbracket, \binom{n}{k} = \frac{n(n-1) \times \dots \times (n-k+1)}{k!}$ .

Écrire alors une fonction telle que l'exécution de la commande `CB3(k, n)` renvoie la valeur de  $\binom{n}{k}$ .

```
1 def CB3(k, n) :
2     res = 1
3     for i in range(1, min(k, n-k) + 1) :
4         res = res * (n-i+1) / i
5     return res
```

- Compléter la fonction suivante afin que l'exécution de `CB4(k, n)` renvoie le triangle de Pascal complété jusqu'à la  $n$ -ième ligne ainsi que la valeur de  $\binom{n}{k}$ .

```
1 import numpy as np
2
3 def CB4(k, n) :
4     CB = np.zeros([n+1, n+1])
5     for j in range(0, n+1) :
6         CB[j][0], CB[j][j] = 1, 1
7     for i in range(2, n+1) :
8         for j in range(1, i) :
9             CB[i][j] = CB[i-1][j-1] + CB[i-1][j]
10    return CB
```

- Vérifier :  $\forall n \in \mathbb{N}^*, \forall k \in \llbracket 1; n \rrbracket, \binom{n}{k} = \frac{n}{k} \binom{n-1}{k-1}$ .

Écrire alors une fonction récursive telle que l'exécution de la commande `CB5(k, n)` renvoie la valeur de  $\binom{n}{k}$ .

```
1 def CB5(k, n) :
2     if k > n :
3         return 0
4     elif k == 0 :
5         return 1
6     else :
7         return n/k * CB5(k-1, n-1)
```