

## TD numéro 5

### Exercice 1 : récursivité profonde

- Écrire une fonction calculant le nombre total d'éléments d'une liste quelconque.  
`(compter '(1 2 (q w e) (r (e (w t)) 3 (e)) (o))) -> 12`
- Écrire une fonction qui prend une liste et ajoute 2 à chaque nombre de cette liste en profondeur.  
`(ajoute2 '(2 b (10 a (56 3) 5) 4)) -> (4 b (12 a (58 5) 7) 6)`

### Exercice 2 : le triangle de Pascal

```

1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
    
```

- Écrire une fonction de deux entiers p et n, qui calcule le nombre de combinaisons de p éléments parmi n :  $C_n^p = C_{n-1}^p + C_{n-1}^{p-1}$  avec  $C_n^0 = 1$  et  $C_n^n = 1$
- Écrire une fonction qui construit une liste Li contenant les coefficients des binômes de degré i (la i<sup>ème</sup> ligne du triangle de Pascal).
- Écrire une fonction qui construit une liste de listes correspondant aux n premières lignes du triangle de Pascal.

### Exercice 3 : une autre version du triangle de Pascal

- Écrire une fonction som2 qui, étant donnée une liste de nombres, rend la liste des sommes des éléments consécutifs de cette liste.  
`(som2 '(1 2 3 4)) -> (3 5 7)`
- Utiliser la fonction som2 pour écrire une autre version de la fonction qui construit une liste de listes correspondant aux n premières lignes du triangle de Pascal.