


```
def add(a, b):
    if b == 0:
        return a
    else:
        return add(a, b-1) + 1
```

1. Déterminer en fonction de b le nombre d'appels de la fonction `add` pour effectuer $a + b$.
2. Expliquer pourquoi la terminaison n'est assurée que dans le cas où b est un entier naturel.
3. Démontrer la correction de la fonction, autrement dit, prouver que l'expression `add(a, b)` a pour valeur la somme $a + b$.
4. Ecrire en suivant le même modèle une fonction récursive `mul` qui effectue la multiplication de deux nombres entiers naturels. Utiliser la fonction `add`; l'opérateur `+` n'est pas autorisé.
5. Ecrire une fonction récursive calculant a^b , avec des valeurs de b entières positives ou nulles. Utiliser la fonction `mul`; ici l'opérateur `*` n'est pas autorisé.

Exercice 4

On considère les deux fonctions `compte1` et `compte2` définies ci-dessous.

```
def compte1(n):
    if n >= 0:
        print(n)
        compte2(n-1)

def compte2(n):
    if n >= 0:
        compte1(n-1)
        print(n)
```

Le paramètre n est un entier naturel. Déterminer le résultat de l'appel `compte1(3)` puis celui de l'appel `compte2(3)`.