

Algorithmique Avancée

-- Travaux Dirigés (Série 2) --

► **Exercice 1.** Pour chacune des fonctions suivantes, l'exécuter à la main sur un exemple, dire pour quelles valeurs du paramètre n elle se termine, et quand c'est le cas ce qu'elle fait ou calcule.

```
public static int f(int n)
{
    if (n == 0) return 1;
    else return f(n+1);
}

public static int sommeBis(int n)
{
    if (n == 0) return 0;
    else {
        int result = sommeBis(n-1);
        result += n;
        return result;
    }
}

public static int g(int n)
{
    if (n <= 1) return 1;
    else return 1 + g(n-2);
}
```

► **Exercice 2.** Construire la forme récursive de la fonction $g()$ définie ci-dessous.

```
public static void g(int n){
    for(int i = 0; i < n; i++)
        System.out.println(i);
    return; }
```

► **Exercice 3.** Écrire une fonction récursive qui calcule la somme de nombres de 1 à n , si $n > 0$ et renvoie 0 sinon. Quelle est la complexité ?

► **Exercice 4.** Donner un algorithme récursif pour calculer x^n , x et n positifs non nuls. Quelle est la complexité ? Peut-on calculer x^n avec moins de multiplications ?

► **Exercice 5.** Écrire une fonction récursive $\text{pgcd}(m, n)$ qui calcule le plus grand diviseur commun des deux entiers (non-négatifs) m et n .