

## **SÉRIE DE TD N°3 (LA RÉCURSIVITÉ)**

Module : Algorithmique et structures de données 2

Année universitaire : 2019/2020

---

### **Exercice 1**

Ecrire des modules **récurifs** qui permettent de :

- 1) Calculer la somme :  $1+2+ \dots +(N-1)+N$
- 2) Calculer la puissance :  $X^N$

### **Exercice 2**

Le calcul du pgcd de deux entiers a et b positifs peut se faire en suivant l'algorithme d'Euclide. Cet algorithme prend pour base le théorème d'Euclide suivante :

- Si  $a > b$  et si on effectue la division euclidienne de a par b :  $a = q * b + r$  avec  $0 \leq r < b$  alors le pgcd de a et b **est égal** au pgcd de b et r.
- Pour calculer pgcd de a et b, il suffit d'itérer (répéter) cette technique : on obtient un reste nul en un nombre fini d'étapes. Le **dernier reste non nul** est alors le pgcd de a et b.

Ecrire un algorithme permettant de calculer le plus grand diviseur commun de deux entiers Nbr1 et Nbr2 d'une manière réursive.

### **Exercice 3**

Soit **V** un vecteur d'entiers, écrire les modules récurives suivantes :

- 1) La procédure **RemplirVE** permettant de remplir V.
- 2) La procédure **AfficherVE** permettant d'afficher les valeurs de V.
- 3) La fonction **Somme** qui retourne la somme des éléments du vecteur.
- 4) La fonction **Max** qui retourne le maximum du vecteur.
- 5) La fonction **Appartient** permettant de vérifier si un élément existe dans le vecteur ou non.

### **Exercice 4 (supplémentaire)**

La recherche dichotomie d'un élément dans un vecteur ordonné s'effectue comme suit :

- On divise le tableau en deux parties sensiblement égales,
- On compare la valeur à chercher avec l'élément du milieu,
- Si elles ne sont pas égales, on s'intéresse uniquement la partie contenant les éléments voulus et on délaisse l'autre partie.
- On recommence ces 3 étapes jusqu'à avoir un seul élément à comparer.

Écrire une fonction réursive qui recherche par dichotomie une valeur **Val**. La fonction retourne le rang de cette valeur si elle existe et -1 sinon ?