

Enseignant : M. C. PIOMBO

## TRAVAUX PRATIQUES

### Série 5

### Procédures et Fonctions récursives

NB : Vous devez impérativement utiliser les méthodes de programmation structurée pour développer ces programmes (structure de contrôle, procédure, fonction).

#### Exercice 1 : Le PGCD

Ecrire une fonction récursive qui calcule le PGCD de deux entiers positifs sachant que :  
 Pour trouver le PGCD de deux nombres, on divise le plus grand nombre par le plus petit. Ensuite, on divise le plus petit par le reste de la division précédente, puis on répète le procédé jusqu'au moment où le reste est nul. Le dernier diviseur est le PGCD recherché.

#### Exercice 2 : La factorielle

Ecrire une fonction récursive qui calcule le produit des entiers positifs de 1 à n sachant que :

0 ! = 1                                      \*\* condition d'arrêt  
 si n > 0 alors n ! = n \* (n-1) !        \*\* définition de la récursivité

#### Exercice 3 : Recherche dichotomique de la première occurrence d'une valeur

Faire un programme qui recherche l'occurrence d'une valeur saisie au clavier dans un tableau ordonné de 1000 entiers. On fera apparaître dans la solution une procédure récursive qui recevra en paramètre le tableau dans lequel doit s'effectuer la recherche. Cette procédure retournera la place occupée par la première occurrence de la valeur dans le tableau. Si la valeur recherchée n'existe pas une exception sera levée. Afin d'avoir une suite aléatoire de valeurs entières utiliser la fonction **random()** du package générique **ada.numerics.discret\_Random**.

#### NB :

On rappelle que rechercher un élément x dans un intervalle [a..b] d'un tableau ordonné T revient à :  
 Tester si T[i] = x pour un i appartenant à l'intervalle [a..b],                                      \*\* condition d'arrêt  
 Ou bien si T[i] /= x, rechercher x dans l'intervalle [i+1..b] ou dans l'intervalle [a..i-1].        \*\* définition de la récursivité

### Structures de données récursives

#### Exercice 4 : Calcul polynomial

On se propose de réaliser un programme permettant le calcul d'un polynôme de la forme :  $2x^8 + 5x^6 + 12x^5 + x^3 + 5$   
 Etant donnée que ce polynôme peut être de degrés N ( non défini à l'avance ), on utilisera pour stocker en mémoire les différents coefficients (entiers) ainsi que les degrés (entiers) définissant chaque terme de ce polynôme une liste dynamique simplement chaînée. Chaque cellule de la liste représentera donc la paire (coefficient, degrés) d'un terme. Le programme fera apparaître au minimum les sous-programmes suivants :

Créer( ? ) : création d'un polynôme vide,  
 Ajouter( ? ) : ajout d'un terme au polynôme (Coefficient + degrés),  
 Evaluer( ? ) : calculer le polynôme,  
 Détruire( ? ) : libérer l'espace occupé par les différents termes du polynôme.

#### NB :

un seul terme par degrés,  
 On doit être capable par lecture du polynôme de connaître son degrés maximum sans avoir à parcourir toute la liste.

#### Exercice 5 : évaluation d'une expression arithmétique

Réaliser un programme qui évalue une expression arithmétique préfixée (ex :  $( * ( + 5 ( * 2 3 ) ) ( + ( * 10 10 ) ( * 9 9 ) ) )$  )  
 saisie au clavier et implémentée en mémoire sous la forme d'un arbre binaire. Les opérateurs binaires admis seront +, -, \*, / sur des entiers.