



Structure de données **Réalisation en langage C : Listes chaînées**
TD 2

Exercice 1 (Liste chaînée simple)

Un polynôme peut être représenté par une liste chaînée dans laquelle chaque nœud contient un coefficient, un exposant et un pointeur vers le terme suivant.

- 1) Définir une fonction de création d'un polynome (**Polynome * CreePolynomeVide()**).
- 2) Définir une fonction permettant d'ajouter des nœuds à un polynome (**void AjouterNoeud(Polynome * po, int coef, int exp)**).
- 3) Etablir un algorithme d'addition de deux polynômes (**polynome1** et **polynome2**).
- 4) Ecrire le programme C correspondant à vos fonctions et à votre algorithme.

Exercice 2 (Liste chaînée ordonnée)

On cherche à gérer une liste chaînée formée de nombres entiers. Chaque nombre introduit est immédiatement placé dans la liste en respectant l'ordre décroissant.

- 1) Ecrire un algorithme qui permet de gérer cette liste chaînée
 - a) Fonction de création de liste **InitListe()**
 - b) Fonction **ListeVide()**
 - c) Fonction **ListePleine()**
 - d) Fonction d'**ajout (Liste *L, int nombre)**
 - e) Fonction de **suppression (Liste *L, int nombre)**
 - f) Fonction d'**affichage (Liste * L)**
- 2) Ecrire le programme C correspondant à votre algorithme.

Exercice 3 (Liste chaînée bidirectionnelle)

Une liste bidirectionnelle est une liste d'éléments qui sont chaînés dans les deux sens.

- 1) Ecrire un algorithme pour insérer, supprimer et chercher des éléments dans une liste bidirectionnelle.
- 2) Traduire votre algorithme en C.