Département IMA / 3A (S6) / 2012 **Programmation Avancée**
Bernard Carré, Nathalie Devésa, Julie Jacques, Shahin Gelareh et Laure Gonnord.

TP7 Listes chaînées et algorithmes récursifs

Objectifs

- Savoir déclarer, construire des listes chaînées en C.
- Savoir reconnaître et implémenter les algorithmes classiques de listes chaînées à l'aide de fonctions *récursives*

Contexte et préparation Nous allons utiliser les listes chaînées (non ordonnées) pour représenter des **ensembles d'entiers**. Les opérations classiques sur les ensembles d'entiers seront réalisées à l'aide d'algorithmes classiques sur les listes, en version récursive.

1 Questions du TP (à faire impérativement)

1.1 Copier/coller rapides

Rapidement, à partir du TP6 (polynômes) :

1. Déclarer un type cellule, pointeur de cellule, liste d'entiers.
2. Écrire une fonction d'ajout en tête d'un entier dans une liste d'entiers (avec allocation)
3. Écrire une fonction de suppression de la première cellule d'une liste.
4. Écrire une fonction qui permet à l'utilisateur de créer une liste d'entiers par lecture dans un fichier (cf tp2).

1.2 Fonctions récursives simples sur les listes

Écrire les fonctions/actions **récursives** suivantes :

1. Impression d'une liste donnée.
2. Destruction d'une liste complète. (Vérifier qu'il n'y a pas de fuite mémoire avec valgrind).
3. Recherche d'un élément donné.

1.3 Ensembles dans des listes

On suppose maintenant que les listes codent des ensembles, c'est-à-dire qu'aucun entier n'apparaît deux fois. Écrire des fonctions/actions **récursives** pour :

1. Vérifier qu'une liste donnée est un ensemble.
2. Calculer le cardinal (nombre d'éléments) d'un ensemble (taille d'une liste !)
3. Transformer une liste en un ensemble.

2 Questions s'il vous reste du temps

Coder les opérations usuelles sur les ensembles : union, intersection, différence, toujours avec des procédures et fonctions récursives.