Département IMA / 3A (S6) / 2013 **Programmation Avancée**
Jonathan Aceituno, Vincent Aranéga, Nathalie Devésa, Rachid Elkouri et Laure Gonnord

TP séance 6 Listes chaînées V2 : 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 utilisons le même contexte que le tp précédent, mais nous désirons cette fois coder des multiensembles, c'est à dire des ensembles qui peuvent contenir plusieurs fois le même élément.

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)

Toutes les questions seront testées au fur et à mesure (appels dans le main), *comme d'habitude!*

1.1 Copier/coller rapides

Rapidement, à partir du tp précédent, dans un nouveau fichier :

1. Déclarer un type cellule, pointeur de cellule, liste de structures "entier avec multiplicité".
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 d'impression d'une liste "multiensemble" donnée (valeur, multiplicité, pour tous les élements de la liste)

Tester !

1.2 Construction de liste à partir d'un fichier

1. Écrire une fonction *récursive* d'insertion d'un entier dans une liste "multiensemble" supposée triée. Si l'élément apparaît dans la liste, on augmente sa multiplicité, sinon, on insère une nouvelle cellule au bon endroit.
2. Utiliser cette fonction pour construire un multiensemble à partir d'un fichier.

La liste "multiensemble" ainsi créée à partir d'un fichier, sera utilisée pour tester les fonctions qui vont suivre.

1.3 Fonctions récursives

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

1. Vérification qu'une liste "multiensemble" est triée croissante.
2. Destruction d'une liste complète. (Vérifier qu'il n'y a pas de fuite mémoire avec valgrind).
3. Recherche de la multiplicité d'un élément donné dans une liste multiensemble.
4. Calcul du cardinal du multiensemble : un entier apparaissant n fois compte pour n.

2 Questions s'il vous reste du temps

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