

## TP4 Listes chaînées avec algorithmes itératifs

### 1 Objectifs

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

**Contexte et préparation :** Nous allons utiliser les listes chaînées (non ordonnées, puis ordonnées) pour représenter des ensembles<sup>1</sup> d'entiers. Les opérations classiques sur les ensembles d'entiers seront réalisées à l'aide d'algorithmes classiques sur les listes, en version itérative.



#### IMPORTANT

Les 2 principales causes de *segmentation fault* avec les listes chaînées :

- une liste non initialisée ;
- un accès à `P->val` ou `P->suiv` avec `P=NULL` .

### 2 Questions du TP (à faire impérativement)

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

#### 2.1 En direct du cours et/ou TD – Listes non triées

1. Déclarer une liste chaînée.
2. Écrire la fonction de création d'une liste chaînée par ajout en tête.
3. Écrire une fonction d'impression d'une liste chaînée.
4. Écrire une fonction qui teste si une liste chaînée est triée.
5. Écrire une fonction de suppression de la cellule de tête d'une liste chaînée.
6. Écrire une fonction de désallocation d'une liste.
7. Tester les fonctions précédentes. Vérifier avec *valgrind* l'absence de fuite mémoire à l'exécution.

#### 2.2 Listes triées

On va maintenant faire en sorte de maintenir une liste triée **sans doublon**.

1. Écrire un algorithme d'insertion d'un entier dans une liste triée par ordre croissant. On ne fera l'insertion que si l'élément n'existe pas déjà dans la liste.
2. Écrire une fonction qui permet de créer une liste d'entiers triée par lecture dans un fichier. Les entiers contenus sur le fichier peuvent être négatifs et dans n'importe quel ordre.

### 3 Questions s'il vous reste du temps

Écrire des fonctions pour :

1. Tester si une liste quelconque ne contient pas de doublons.
2. Faire l'union de deux ensembles (c.a.d., l'union de deux listes sans doublon).
3. Faire l'intersection de deux ensembles.
4. Transformer une liste quelconque en un ensemble (supprimer les doublons).

---

1. Un *ensemble* est une **collection** ou un groupement d'**objets distincts** ; ces objets s'appellent les éléments de cet ensemble.