

## DÉPARTEMENT DE MATHÉMATIQUES, D'INFORMATIQUE ET DE GÉNIE

### Structures de données et algorithmes

#### Devoir 3 — Énoncé

---

SIGLE : INF21307  
TITRE : Structures de données et algorithmes  
GROUPE : MS  
PROFESSEUR : Steven Pigeon  
K-212  
steven\_pigeon@uqar.ca

MODALITÉ : À faire seul ou en équipe de deux.  
DATE DE REMISE : 15 avril 2025, avant minuit.

---

Les listes doublement chaînées sont fort utiles, spécialement lorsque les données sont dodues et qu'il est coûteux de les déplacer. En classe (épisode 7), nous avons discuté de la possibilité de marquer un nœud d'une liste comme étant « libre » plutôt que de procéder à une libération immédiate et totale d'un nœud. Cela permet d'effacer paresseusement une valeur et de réutiliser un nœud libre plus tard, nous dispensant de passer par des allocations et des libérations de mémoire (qui peuvent s'avérer lentes).

Langage de programmation au choix.

#### 1. 15 pts.

(a) **10 pts.** Implémentez un objet/classe liste (doublement) chaînée. Supposez pour simplifier l'implémentation que les données sont des entiers. Votre classe doit offrir les fonctionnalités suivantes :

i. **2 pts.** Avoir une fonction *ajouter*, qui permet d'ajouter au premier trou libre. Il faut gérer

- correctement la liste initialement vide.
- ii. **2 pts.** Avoir une fonction *trouver*, qui retourne un pointeur/référence/itérateur sur la première instance d'une valeur. Si la valeur n'est pas présente, alors il doit être possible de reconnaître le pointeur/référence/itérateur comme étant invalide (voir *itérateur*, plus bas).
  - iii. **2 pts.** Avoir une fonction *supprimer*, qui prend un pointeur/référence en paramètre (celui retourné par *trouver*, par exemple) et marque un nœud de la liste comme étant « libre ».
  - iv. **2 pts.** Avoir une fonction *compacter*, qui balaye la liste et libère tous les nœuds marqués comme libres. Il faut gérer correctement les listes vides (y compris après compaction).
  - v. **2 pts.** Avoir un *itérateur* qui permet de visiter un par un les éléments de la liste, en sautant par dessus ceux qui sont marqués libres. Votre itérateur doit donc commencer au début de la liste. Votre itérateur doit par ailleurs réserver une valeur spéciale pour la fin (une valeur nulle, ou quelque chose d'autre, comme  $-1$ ), ce qui nous permet d'arrêter de boucler.
- (b) **5 pts.** Votre programme doit inclure une démonstration adéquate des fonctionnalités (une interface à la console suffira amplement). Votre interface devra permettre d'insérer une valeur (entière), supprimer une valeur, lister les valeurs contenues, et lancer la compaction.
- (c) **(bonus) 5 pts.** Implémentez une liste doublement chaînée avec sentinelle. Vous serez notés sur l'élégance de l'implémentation.

\*  
\* \*

Le **plagiat** ne sera pas toléré. Comme les devoirs sont *individuels* (ou par équipe de deux), il vous faudra donc éviter les « collaborations », déclarées ou non. De plus, vous devrez *toujours* déclarer les sources dont vous vous serez *inspirés* (livres, articles, site Web, etc.). Une collaboration non avouée est considérée comme un plagiat et sera traitée conséquemment. Il en va de même pour tout matériel dont vous ne seriez pas l'auteur (code, texte, image, etc.). Sera évidemment considéré comme un plagiat tout code copié-collé (oui, même en changeant les noms de variables), qu'il y ait une référence ou non à la source. Le plagiat mène automatiquement aux conséquences disciplinaires prévues au règlement (Règlement 5, article 15). Notez que le but n'est pas de vous empêcher d'utiliser des sources externes, mais de vous obliger à les déclarer, les citer, et à les *comprendre* suffisamment pour arriver à vos solutions originales.

\*  
\* \*

La qualité du français écrit est de mise, et jusqu'à 10% de la note finale pourraient être retranchés. N'oubliez pas de citer vos sources, mêmes celles dont vous vous inspirez plus ou moins vaguement. Enfin, vous vous exposez à 20% de pénalité par jour de retard, jusqu'à concurrence de quatre jours — sauf circonstances *exceptionnelles*.

Pour la remise du devoir, préparez une archive (.zip) avec votre solution *sans les fichiers intermédiaires et temporaires*, et dont le nom sera *votrenom.devoir3.zip* (ou un autre type d'archive). Cette archive sera déposée sur Moodle dans le dépôt prévu à cet effet. Votre archive doit spécifier (dans un *readme.txt* par

exemple) quel système vous utilisez (Windows+Visual Studio, Linux+Makefile, etc.).