

420-977-RO Concepts de structuration des données  
informatiques  
**TP1 – Manipulations de données en C++**

Sylvain Labranche — Collège de Rosemont

Printemps 2017

## 1 Énoncé

Vous discutez avec un ami à qui vous mentionnez que vous êtes en train d'apprendre le langage C++. Appréhivé, il vous questionne à savoir si c'est vraiment si difficile que la rumeur le dit. Vous lui répondez qu'il y a beaucoup de ressemblances avec Java, mais qu'il existe des différences importantes, notamment en ce qui a trait à la gestion de la mémoire. Vous continuez en expliquant la différence entre la mémoire sur la pile et sur le tas, qu'il faut allouer et désallouer la mémoire sur tas. Vous comprenez à son regard qu'il serait pertinent d'illustrer vos explications avec un programme d'exemple. Vous décidez d'écrire le programme suivant, illustrant les bases de C++ que vous maîtrisez:

Le programme ouvre un fichier, dont le format est toujours le même: un nombre  $n$  indique combien il y a de nombres dans le fichier, suivi de  $n$  nombres sur une autre ligne. Le programme enregistre les données dans un tableau *alloué dynamiquement sur le tas*. Dans votre programme, expliquez en commentaire pourquoi il serait incorrect d'allouer le tableau sur la pile. Un menu s'affiche et offre les options suivantes:

1. Somme;
2. Moyenne;
3. Inversion;
4. Mémoire dynamique;
5. Quitter.

L'utilisateur entre son choix au clavier. Le traitement de chacune des options se fera **dans une méthode**, avec pour résultat:

1. Somme : La somme des nombres dans le tableau est affichée. Le tableau reste inchangé.
2. Moyenne : La moyenne des nombres dans le tableau est affichée. Le tableau reste inchangé.
3. Inversion de l'ordre des éléments : L'ordre des éléments *dans le tableau d'origine* est changé. Le résultat est affiché. Par exemple, avec un tableau contenant *1; 2; 3*, la méthode affichera *3; 2; 1*.

4. Mémoire dynamique: Localement, la méthode attribue dynamiquement (sur le tas avec un pointeur) un tableau de  $n$  éléments et copie chacun des éléments du tableau d'origine dans ce nouveau tableau. Les éléments sont affichés à partir du nouveau tableau, puis la mémoire est libérée. Le tableau original reste inchangé.

Précisions : Cette méthode fait une copie du tableau et affiche les éléments à partir de la copie.

Vous devez déterminer vous-même les arguments nécessaires aux méthodes.

Pensez aussi à l'utilisation du mot clé **const** lorsque c'est nécessaire. Le menu s'affiche tant que l'utilisateur n'a pas choisi de quitter.

## 2 Modalités

Ce travail est fait individuellement. Vous pouvez collaborer, mais vous **devez** écrire votre propre code. Un plagiat, même partiel, du code d'un autre étudiant entraînera la note 0 pour les deux étudiants.

## 3 Remise

La date limite pour la remise est prévue **le vendredi 5 mai à 23h59**.

Tout retard entraînera une diminution de 10 % de la note maximale par jour de retard.

## 4 Évaluation

Ce travail compte pour 15 % de votre session.

Il y a quatre méthodes à implémenter, chacune comptant pour 3 %. Le programme principal (contenu de la méthode *main*) comptera aussi pour 3 %. Dans chaque cas, le bon fonctionnement, la qualité du code et l'efficacité seront considérés.