

---

## Feuille 1

### Introduction et Outils

---

Chaque TP sera divisé en deux phases qui doivent, dans la mesure du possible, être traitées simultanément. La première phase consiste en l'implémentation pure de la classe et de ses méthodes. La seconde phase s'attachera à l'analyse et à l'amélioration du code.

---

L'objectif de ce premier TP est de mettre en place les objets qui seront nécessaires à la réalisation des prochains TP, ainsi que l'utilisation de différents outils de programmation.

Dans un premier temps, vous implémenterez une classe vecteur dont l'élément de base sera le double, cette classe sera nommée `Dvector`. Cette classe devra utiliser l'allocation dynamique pour le stockage des données.

#### Phase 1 : Implémentation

##### 1. Implémentation des constructeurs et destructeur :

- Implémenter le constructeur par défaut.
- Implémenter un constructeur prenant en argument la taille du vecteur à créer et un second argument optionnel qui spécifie la valeur à utiliser pour initialiser les éléments du vecteur.
- Implémenter un constructeur par recopie.
- Implémenter le destructeur.

Pour l'instant, on ne demande pas de surcharger l'opérateur `=`, par conséquent on prêtera la plus grande attention à la différence qui existe entre les 2 écritures :

```
Dvector x;                               Dvector x = Dvector (3, 1.);
x = Dvector (3, 1.);
```

##### 2. Implémentation des méthodes d'accès et de manipulations de base :

- Implémenter une méthode `int size ()` renvoyant la taille du vecteur.
- Implémenter une méthode pour l'affichage d'un `Dvector`.
- Implémenter une méthode qui génère un vecteur selon la loi uniforme sur  $[0, 1]$ . On pourra dans un premier temps utiliser la fonction `random` de la librairie C.

##### 3. Implémenter un constructeur permettant de créer un vecteur à partir d'un fichier dans lequel les éléments du vecteur sont stockés en colonne. Afin de réaliser la lecture, utiliser les flux.

#### Phase 2 : Analyse

**Question 1.** *Pour chacune des méthodes précédentes, écrire un programme test.*

**Question 2.** *Ajouter aux constructeurs une instruction d'affichage pour repérer les appels implicites aux constructeurs et faire de même avec le destructeur.*

**Question 3.** *Utiliser `valgrind` pour vérifier que toute la mémoire a été libérée.*