

**Evaluation n°1****Rendu :**

- Le code devra être rendu lundi 6 avril 2020 soir au plus tard.
- Le code doit être indenté, et pourvu d'un Makefile.
- La structure des fichiers doit être cohérente.
- Respectez les noms donnés.
- La présence de structures du C (printf, malloc, .h au lieu de .hpp, ...) sera pénalisée, sauf pour la génération de nombres aléatoires (bonus si faite en C++).

**Exercice 1**

On veut programmer une classe *Vector* représentant un vecteur bidimensionnel. On souhaite que cette classe possède deux attributs (privés)  $x$  et  $y$  représentant les coordonnées, et les méthodes suivantes :

- Un constructeur par défaut créant un vecteur nul
- Un constructeur permettant de spécifier les coordonnées
- Une méthode **void** affiche() qui affiche les coordonnées au format  $(x; y)$
- Une méthode **double** norme() qui renvoie la norme-2 du vecteur
- Des accesseurs (getters and setters) pour  $x$  et  $y$
- Une méthode **void** addition(const **Vector** & v) qui ajoute v au vecteur (*this*) et le modifie.
- Une méthode **void** initRandom(double inf, double sup) qui initialise le vecteur avec des valeurs uniformément aléatoires comprises entre "inf" et "sup".

1. Implémentez les méthodes ci-dessus.
2. Écrire un test pour chacune de ces méthodes dans le *main*
3. Y a-t-il besoin d'un destructeur ? Justifiez en commentaires, et ajouter le code si besoin.

**Exercice 2**

Dans le même header, définissez la classe *DynVector*, qui implémente de manière dynamique un vecteur  $n$ -dimensionnel sous forme de tableau. Cette classe aura pour seul attribut (privé) un pointeur sur un **double**. On s'intéresse ici à sa construction, pas à son utilisation. Les méthodes suivantes doivent apparaître :

- Un constructeur permettant de spécifier la taille du vecteur, et qui le remplit de zéros.
- Une méthode **double** norme() qui renvoie la norme-2 du vecteur
- Une méthode **void** affiche() qui affiche les coordonnées au format  $(x_1; \dots; x_n)$
- Une méthode **void** initRandom(double inf, double sup) qui initialise le vecteur avec des valeurs uniformément aléatoires comprises entre "inf" et "sup".

1. Implémentez les méthodes ci-dessus.
2. Appelez chacune de ces méthodes dans le *main*
3. Y a-t-il besoin d'un destructeur ? Justifiez en commentaires, et ajouter le code si besoin.