ENSAE

TD n°5 C++

Exercice 1 (opérateurs)

Récupérer, si possible, la classe `Point` de l'exercice 4 du précédent TP. Sinon créer une classe `Point` représentant les points de l'espace à deux dimensions $\mathbb{R}^2$ ainsi qu'une classe `Vecteur` de ce même espace.
On implémentera les opérateurs permettant d'obtenir un vecteur à partir de deux vecteurs, un point à partir d'un point et d'un vecteur, comme par exemple:

```
Point o;
Vecteur v1(5,5), v2(1,0);
Point p = o+v1;
Vecteur v3 = v1+v2;
```

Exercice 2 (opérateurs)

Créer une classe permettant de représenter des Fractions (dont les numérateurs et dénominateurs seront des champs de type `int`). Implémenter les opérateurs arithmétiques permettant d'obtenir la somme, la différence, le produit, la division des Fractions. On implémentera aussi l'opérateur unaire permettant d'obtenir l'inverse et l'opérateur - pour l'opposé. De sorte que l'on puisse écrire quelque chose comme :

```
Fraction f1(1,2); // Un demi
Fraction f2(45); // 45
Fraction f3 = -f1*f2+3*!f1;
```

Exercice 3 (opérateurs et flux)

Reprendre les diverses classes des derniers TPs et implémenter l'opérateur permettant d'afficher les objets de cette classe comme pour les types ordinaires. Soit par exemple:

```
Fraction f1(1,2);
cout << "Voici un demi " << f1 << endl;
```

Exercice 4 (opérateurs de conversion)

Reprendre l'exercice sur les fractions afin d'y ajouter les opérateurs de conversion vers des types primitifs, afin que l'on puisse convertir une fraction en double ou en int.

Exercice 5 (opérateurs)

Reprendre l'exercice sur les vecteurs (tp 4) et implémenter un opérateur d'accès aux membres du vecteur comme aux éléments d'un tableau. Par exemple :

```
Vecteur v(12,45,89);
cout << v[0] << endl;
```

Exercice 6

Implémenter un type Matrix permettant de manipuler des matrices (carrées en 2 dimensions suffira) à l'aide des opérateurs «habituels». On pensera au produit avec des vecteurs, avec des scalaires, etc.