

Travaux Pratiques N°4

Exercice 1 :

Soit la classe *Vector* dont la spécification est donnée comme suit :

```
class Vector {
    const int Size;
    int* Elements;
public :
    Vector(int S);
    Vector(int S, int* Elts);
    Vector(const Vector& V);
    ~Vector();
    void Show();
    void Set(); // Effectue la saisie de tous les éléments du vecteur
    void Set(int Index, int Value);
    int Get(int Index);
};
```

- 1- Donner les définitions des méthodes de la classe *Vector*.
- 2- Donner une fonction principale qui montre l'instanciation de la classe *Vector* et l'utilisation de ses différentes méthodes.

Exercice 2 :

Soit la classe *Matrix* dont la spécification est donnée comme suit :

```
class Matrix {
    const int NbLines;
    const int NbColumns;
    int** Elements;
public :
    Matrix (int L, int C);
    Matrix (int L, int C, int** Elts);
    Matrix(const Matrix& M);
    ~Matrix();
    void Set(int LineIndex, int ColumnIndex, int Value);
    int Get(int LineIndex, int ColumnIndex);
    void Show(); // Affiche le contenu de la matrice
    void Set(); // effectue la saisie de tous les éléments
};
```

- 1- Donner les définitions des méthodes de la classe *Matrix*.
- 2- Donner une fonction principale qui montre l'instanciation de la classe *Matrix* et l'utilisation de ses différentes méthodes.

Exercice 3 :

- 1- Proposer une fonction *Multiply* externe et amie aux deux classes *Vector* et *Matrix* qui effectue le produit d'une matrice et d'un vecteur.
- 2- Transformer la fonction amie *Multiply* de façon à ce qu'elle devienne une méthode de la classe *Matrix*. Indiquer comment faut-il modifier l'organisation du code pour pouvoir compiler avec succès.