Practical Work No. 4

Exercise 1:

Let the class Vector be specified as follows:

```
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();// Grabs all elements of the vector
      void Set(int Index, int Value);
      int Get(int Index);
};
```

1- Give the definitions of the methods of the Vector class.
2- Give a main function that shows the instantiation of the Vector class and the use of its different methods.

Exercise 2:

Let the class Matrix be given the following specification:

```
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();// Displays the contents of the matrix
      void Set();// performs the capture of all elements
};
```

1- Give the definitions of the methods of the Matrix class.
2- Give a main function that shows the instantiation of the Matrix class and the use of its different methods.

Exercise 3:

1- Propose an external and friendly Multiply function to both Vector and Matrix classes which performs the product of a matrix and a vector.
2- Transform the friend function Multiply so that it becomes a method of the Matrix class. Indicate how the code organization must be modified to compile successfully.