Programming Assignment 6:
Linear Equation System Solver

Name: Derek (劉哲瑋) Number: D1262032

I first define a class named Matrix, and two subclasses named SMatrix and Vector respectively, as shown in the pictures below.

```
1  #ifndef matrix
2  #define matrix
3
4  #include <iostream>
5
6  using namespace std;
7
8  // Matrix operations
9  class Matrix {
10
11 friend ostream &operator<<(ostream&, const Matrix&); // friend output funct
12 friend istream &operator>>(istream&, Matrix&); // friend input function
13 friend Matrix operator+(const double &, const Matrix &); // scalar-matrix
       addition
14 friend Matrix operator-(const double &, const Matrix &); // scalar-matrix
       subtraction
15 friend Matrix operator*(const double &, const Matrix &); // scalar-matrix
       multiplication
16
17 protected:
18     int row, col; // row size and column size of a matrix
19     double** m; // a pointer to matrix elements
20     void allocateMatrix(); // allocate matrix elements
21     void deallocateMatrix(); // deallocate matrix elements
22
23     public:
24         Matrix(int=0, int=0); // default matrix constructor
25         Matrix(const Matrix&); // matrix copy constructor
26         ~Matrix(); // matrix destructor
27
28         void setSize(int, int); // set row size and column size
29         void setElement(int, int, double); // set a matrix element
30         double getElement(int, int) const; // get a matrix element
```

```
#ifndef smatrix
#define smatrix

#include "matrix.h"

// Square matrix subclass.
class SMatrix: public Matrix {

    public:
        SMatrix(int=0); // default square matrix constructor
        SMatrix(const Matrix&); // copy constructor from a matrix
        SMatrix(const SMatrix&); // copy constructor from a square matrix
        void setSize(const int); // set the size of a square matrix
        double determinant() const; // determinant function
};

#endif
```

```
#ifndef vector
#define vector

#include "matrix.h"

// Vector subclass, representing a column vector.
class Vector: public Matrix { // inherit class Matrix

    public:
        Vector(int=0); // default vector constructor
        Vector(const Matrix&, const int); // copy constructor from a matrix a
            column index
        Vector(const Vector&); // copy constructor from a vectori
        void setSize(const int); // set the size of a vector
        Matrix vector_replace(int, const Matrix&); // Replace a column of a
            matrix by the vector
};

#endif
```

Then, I define n, i and j using int data type, A(n) using SMatrix, C(n) and X(n) using Vector and V(n, 1) using Matrix.

Next, I write some details about the functions defined. Among them, the function “determinant” is the most difficult, as shown below.

```
double SMatrix::determinant() const{
    double d=0.0;
    int j, p, q;
    if(row==1){ d=m[0][0]; }
    else if(row>1){
        SMatrix cofactor(row-1);
        for(j=0;j<row;j++){
            for(p=0;p<row-1;p++){
                for(q=0;q<row-1;q++){
                    if(q<j) cofactor.setElement(p, q, m[p+1][q]);
                    if(q>=j) cofactor.setElement(p, q, m[p+1][q+1]);
                }
            }
            d+=pow(-1, j%2)*m[0][j]*cofactor.determinant();
        }
    }
    else cout<<"The matrix is empty."<<endl<<endl;
    return d;
}
```

Next, I print the coefficient matrix A, constant vector C, and the system of linear equations and solve the linear equation system to obtain the result of solution vector X by using function “vector_replace”.

Finally, I use the code below to verify solution of the linear equation system.

```
    cout<<endl<<endl<<"Verify solution of the linear equation system:"<<endl;
    V=A*X-C;
    for(i=0;i<n;i++){
        if(V.getElement(i, 0)<0.0000001) cout<<"    Equation "<<i<<"
            passes."<<endl;
        else cout<<"    Equation "<<i<<" fails."<<endl;
    }
```