```
#include <iostream>
#include <fstream>
#include <vector>
#include <iomanip>
#include <cmath>
using namespace std;

class tridiagonal_matrix
{
private:
        int dimension;
        vector <double> diag;
        vector <double> upperdiag;
        vector <double> lowerdiag;
        vector <double> hatupperdiag; // modified upper diag entries
        vector <double> r; // vector containing denominators
        bool transformed;

public:
        tridiagonal_matrix(int m);

        tridiagonal_matrix(const tridiagonal_matrix *mat);

        int get_dimension() const;

        void set_diagonal_entry(int i, double val);

        void set_upper_diagonal_entry(int i, double val);

        void set_lower_diagonal_entry(int i, double val);

        double get_diagonal_entry(int i) const;

        double get_upper_diagonal_entry(int i) const;

        double get_lower_diagonal_entry(int i) const;

        double get_r_entry(int i) const;

        double get_hat_upper_diagonal_entry(int i) const;

        bool is_transformed() const;

        // diag[i] = diag[i] + val;
        void add_to_diagonal_entry(int i, double val);

        // upperdiag[i] = diag[i] + val;
        void add_to_upper_diagonal_entry(int i, double val);

        // lowerdiag[i] = diag[i] + val;
        void add_to_lower_diagonal_entry(int i, double val);

        void transform();

        vector <double> solve_linear_system(const vector<double> & rhs) const;

        // perform matrix vector multiplication
        vector<double> Mult(const vector<double> & lhs) const;
```

```
        ~tridiagonal_matrix();
};
```