P4 – Matrices and Vectors

Prep

2.1 Preparation Section 1

```cpp
int main() {
    default_random_engine rndeng;                       //generator seed

    uniform_int_distribution<int> distribution(1, 10);  //generator distribution

    vector<int> v;

    for(int i=0; i<10; i++) {
        v.push_back(distribution(rndeng));

        cout << "Random Number: " << v[i] << endl;
    }

    return 0;
}
```

```
D:\2020+21\ELEC1204 - Advanced Programming\P4\src\Prep>main.exe
Random Number: 1
Random Number: 2
Random Number: 8
Random Number: 5
Random Number: 6
Random Number: 3
Random Number: 1
Random Number: 7
Random Number: 7
Random Number: 10
```

2.2 Preparation Section 2

```cpp
#define MAT_S 3

int main() {
    default_random_engine rndeng;                       //generator seed

    uniform_real_distribution<double> distribution(1, 10);  //generator distribution

    vector<vector<double>> v;

    for(int i=0; i<MAT_S; i++) {

        vector<double> r;

        for(int j=0; j<MAT_S; j++) {
            r.push_back(distribution(rndeng));


        }

        v.push_back(r);
    }


    for(int i=0; i<v.size(); i++) {

        for(int j=0; j<v[i].size(); j++) {
            cout << " " << v[i][j];
        }

        cout << endl;
    }



    return 0;
}
```

```
D:\2020+21\ELEC1204 - Advanced Programming\P4\src\Prep>main.exe
 2.18384 5.12785 2.97063
 7.10978 9.41224 5.67475
 1.31115 5.7673 1.06928

D:\2020+21\ELEC1204 - Advanced Programming\P4\src\Prep>
```

2.3 Preparation Section 3

```
vector<double> TridiagonalSolve(double E, vector<double> D, vector<double> R) {
    vector<double> c;
    float id;

    vector<double> L;

    for(int i=0; i<D.size(); i++) {
        c.push_back(E);
    }

    c[0] /= D[0];
    R[0] /= D[0];

    for(int i=1; i<D.size(); i++) {
        id = D[i] - c[i - 1] * E;
        c[i] /= id;
        R[i] = (R[i] - R[i - 1] * E) / id;
    }

    L[D.size() - 1] = R[D.size() - 1];

    for(int i=D.size() - 2; i>=0; i++) {
        L[i] = R[i] - c[i] * L[i + 1];
    }

    return L;
}
```