Листинг окончательного варианта программы и тесты к ней.

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int sqrt(int n)

{

  for(int res = 0; res <= n; res++)

  {

    if(res\*res == n)

    {

      return res;

    }

  }

  throw n;

}

vector<vector <int>> mult(vector<vector <int>> m1, vector<vector <int>>m2)

{

  int n = m1[0].size();

  vector<vector <int>> result = m1;

  for(int i = 0; i < n; i++)

  {

    for(int j = 0; j < n; j++)

    {

      int s = 0;

      for(int p = 0; p < n; p++)

      {

        s += m1[i][p]\*m2[p][j];

      }

      result[i][j] = s;

    }

  }

  return result;

}

void printM(vector<vector <int>> m)

{

  int n = m[0].size();

  for(int i = 0; i < n; i++)

  {

    for(int j = 0; j < n; j++)

    {

      cout << m[i][j] << ' ';

    }

    cout << endl;

  }

}

int main()

{

  vector<int> line;

  int val;

  while(cin >> val)

  {

    line.push\_back(val);

  }

  int n = sqrt(line.size());

  vector<int> mLine;

  vector<vector<int>> matrix, tMatrix, multedMatrix;

  for(int i = 0; i < n; i++)

  {

    for(int j = 0; j < n; j++)

    {

      mLine.push\_back(line[i\*n+j]);

    }

    matrix.push\_back(mLine);

    mLine.erase(mLine.begin(), mLine.end());

  }

  tMatrix = matrix;

  int temp;

  for(int i = 0; i < n; i++)

  {

    for(int j = i; j < n; j++)

    {

      temp = tMatrix[i][j];

      tMatrix[i][j] = tMatrix[j][i];

      tMatrix[j][i] = temp;

    }

  }

  multedMatrix = mult(matrix, tMatrix);

  cout << "-------------------------" << endl;

  printM(matrix);

  cout << "-------------------------" << endl;

  printM(tMatrix);

  cout << "-------------------------" << endl;

  printM(multedMatrix);

  cout << "-------------------------" << endl;

  return 0;

}

Тесты:

Так как в задании указывается, что матрицы размерами вплоть до 8x8. То будет 8 тестов с соответствующими входными матрицами.

-------------------------

1

-------------------------

1

-------------------------

1

-------------------------

1 2

3 4

-------------------------

1 3

2 4

-------------------------

5 11

11 25

-------------------------

1 2 3

4 5 6

7 8 9

-------------------------

1 4 7

2 5 8

3 6 9

-------------------------

14 32 50

32 77 122

50 122 194

-------------------------

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

-------------------------

1 5 9 13

2 6 10 14

3 7 11 15

4 8 12 16

-------------------------

30 70 110 150

70 174 278 382

110 278 446 614

150 382 614 846

-------------------------

1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

16 17 18 19 20

21 22 23 24 25

-------------------------

1 6 11 16 21

2 7 12 17 22

3 8 13 18 23

4 9 14 19 24

5 10 15 20 25

-------------------------

55 130 205 280 355

130 330 530 730 930

205 530 855 1180 1505

280 730 1180 1630 2080

355 930 1505 2080 2655

-------------------------

1 2 3 4 5 6

7 8 9 10 11 12

13 14 15 16 17 18

19 20 21 22 23 24

25 26 27 28 29 30

31 32 33 34 35 36

-------------------------

1 7 13 19 25 31

2 8 14 20 26 32

3 9 15 21 27 33

4 10 16 22 28 34

5 11 17 23 29 35

6 12 18 24 30 36

-------------------------

91 217 343 469 595 721

217 559 901 1243 1585 1927

343 901 1459 2017 2575 3133

469 1243 2017 2791 3565 4339

595 1585 2575 3565 4555 5545

721 1927 3133 4339 5545 6751

-------------------------

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31 32 33 34 35

36 37 38 39 40 41 42

43 44 45 46 47 48 49

-------------------------

1 8 15 22 29 36 43

2 9 16 23 30 37 44

3 10 17 24 31 38 45

4 11 18 25 32 39 46

5 12 19 26 33 40 47

6 13 20 27 34 41 48

7 14 21 28 35 42 49

-------------------------

140 336 532 728 924 1120 1316

336 875 1414 1953 2492 3031 3570

532 1414 2296 3178 4060 4942 5824

728 1953 3178 4403 5628 6853 8078

924 2492 4060 5628 7196 8764 10332

1120 3031 4942 6853 8764 10675 12586

1316 3570 5824 8078 10332 12586 14840

-------------------------

1 2 3 4 5 6 7 8

9 10 11 12 13 14 15 16

17 18 19 20 21 22 23 24

25 26 27 28 29 30 31 32

33 34 35 36 37 38 39 40

41 42 43 44 45 46 47 48

49 50 51 52 53 54 55 56

57 58 59 60 61 62 63 64

-------------------------

1 9 17 25 33 41 49 57

2 10 18 26 34 42 50 58

3 11 19 27 35 43 51 59

4 12 20 28 36 44 52 60

5 13 21 29 37 45 53 61

6 14 22 30 38 46 54 62

7 15 23 31 39 47 55 63

8 16 24 32 40 48 56 64

-------------------------

204 492 780 1068 1356 1644 1932 2220

492 1292 2092 2892 3692 4492 5292 6092

780 2092 3404 4716 6028 7340 8652 9964

1068 2892 4716 6540 8364 10188 12012 13836

1356 3692 6028 8364 10700 13036 15372 17708

1644 4492 7340 10188 13036 15884 18732 21580

1932 5292 8652 12012 15372 18732 22092 25452

2220 6092 9964 13836 17708 21580 25452 29324