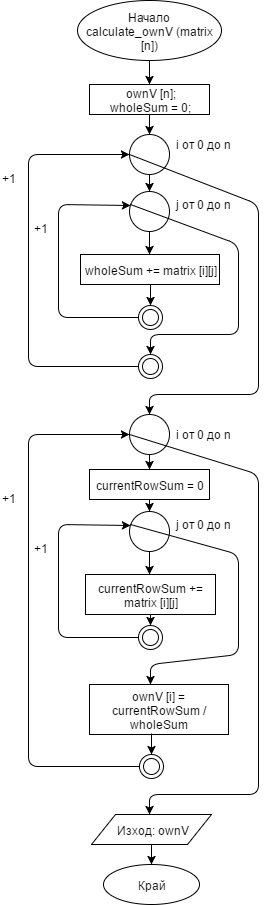
Приемаме,че SIZE е размера на матрицата. (За да е валидно вдигането и на 2ра степен, тя е задължително квадратна.)



vector<double> calculate\_ownV(double\*\* matrix)

{

vector<double> ownV(SIZE);

double wholeSum = 0;

for (int i = 0; i < SIZE; ++i)

{

for (int j = 0; j < SIZE; ++j)

{

wholeSum += matrix[i][j];

}

}

for (int i = 0; i < SIZE; ++i)

{

double currentRowSum = 0;

for (int j = 0; j < SIZE; ++j)

{

currentRowSum += matrix[i][j];

}

ownV[i] = currentRowSum / wholeSum;

}

return ownV;

}

bool areVectorsEqual

(vector<double> v1, vector<double> v2, double epsilon)

{

for (int i = 0; i < v1.size(); ++i)

{

if (abs(v2[i] - v1[i]) > epsilon)

{

return false;

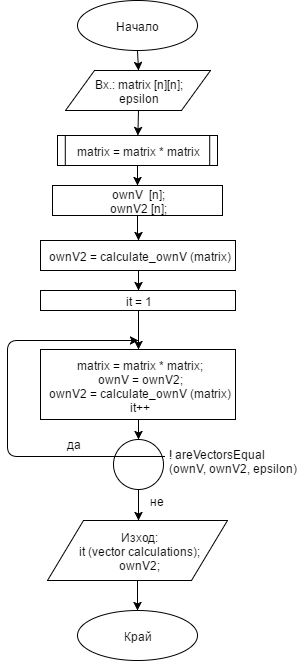
}

}

return true;

}

int main()



{

double\*\* matrix = new double\*[SIZE];

for (int i = 0; i < SIZE; ++i)

{

matrix[i] = new double[SIZE];

cout << "Enter row " << i << endl;

for (int j = 0; j < SIZE; ++j)

{

cin >> matrix[i][j];

}

}

double epsilon;

cout << "Enter epsilon precision for

the own vector difference: ";

cin >> epsilon;

vector<double> ownV(SIZE);

vector<double> ownV2(SIZE);

double\*\* res = multiplyMatrices(matrix, matrix);

ownV2 = calculate\_ownV(res);

printMatrix(res);

printOwnVector(ownV);

int it = 1;

do

{

res = multiplyMatrices(res, res);

printMatrix(res);

ownV = ownV2;

ownV2 = calculate\_ownV(res);

printOwnVector(ownV2);

it++;

}

while (!areVectorsEqual(ownV, ownV2, epsilon));

cout << "Program ended with " << it << " vector calculations.\n";

return 0;

}