Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы численного анализа

**ОТЧЁТ**

к лабораторной работе

на тему

Решение систем линейных алгебраических уравнений (СЛАУ)

методом Гаусса и с помощью его модификаций

Выполнил: студент группы 153503

Кончик Денис Сергеевич

Проверил: Анисимов Владимир Яковлевич

Минск 2022

**CОДЕРЖАНИЕ**

1. ЦЕЛЬ3

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ4

3. ЗАДАНИЕ8

4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ9

5. РЕЗУЛЬТАТЫ12

6. ОЦЕНКА14

7. ВЫВОД15

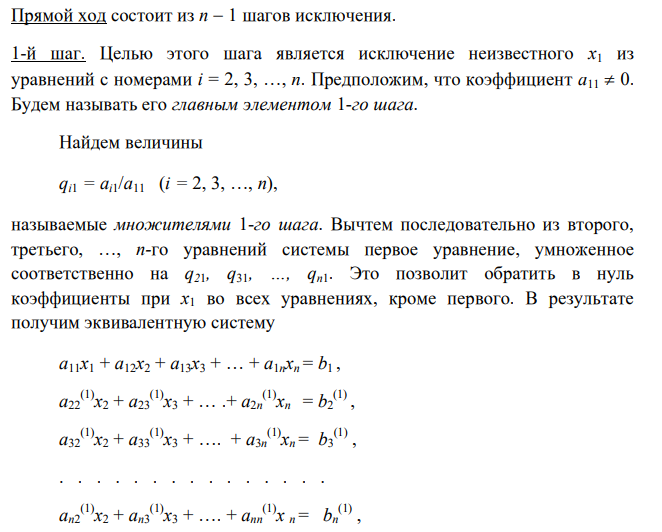
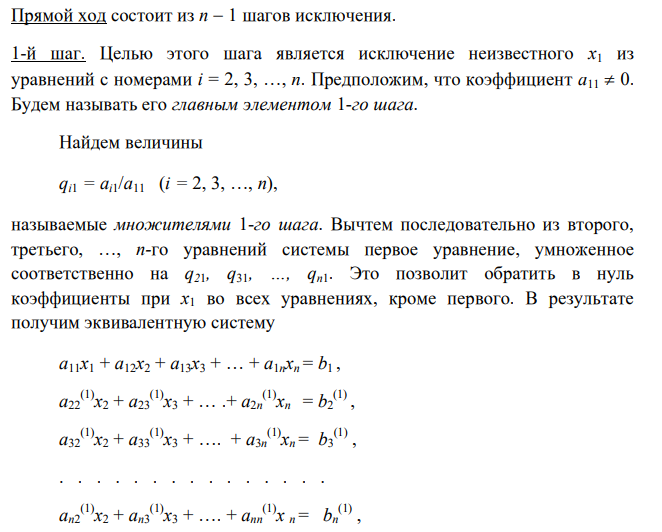
# **ЦЕЛЬ**

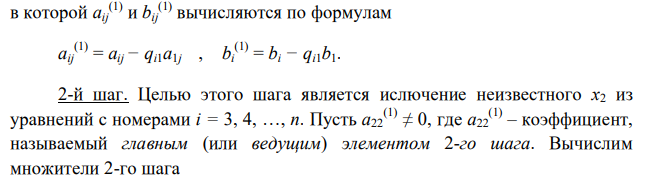
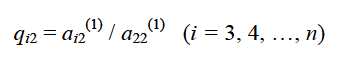
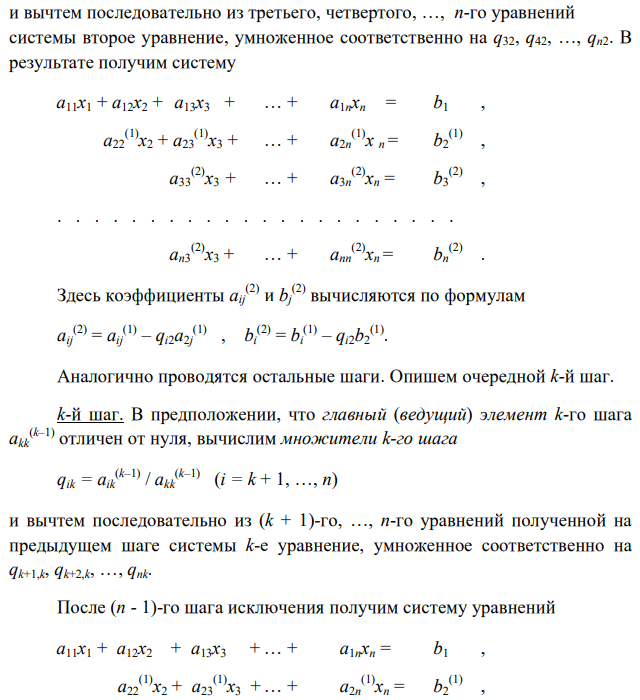
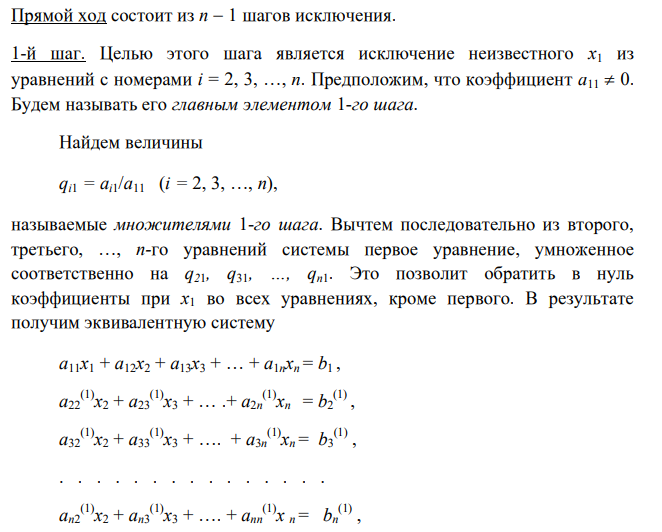
1. Изучить метод Гаусса и его модификации, составить алгоритм метода и программу его реализации, получить численное решение заданной СЛАУ.
2. Составить алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ.
3. Составить программу решения СЛАУ по разработанному алгоритму;
4. Выполнить тестовые примеры и проверить правильность работы программы.

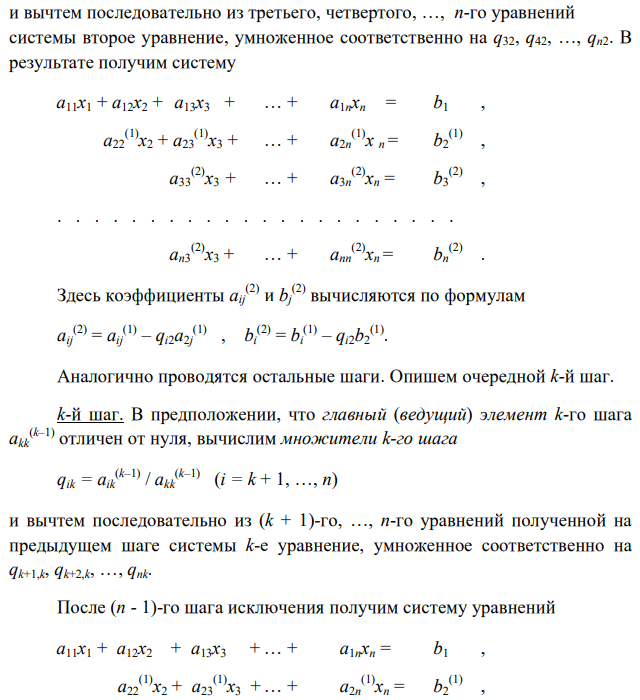
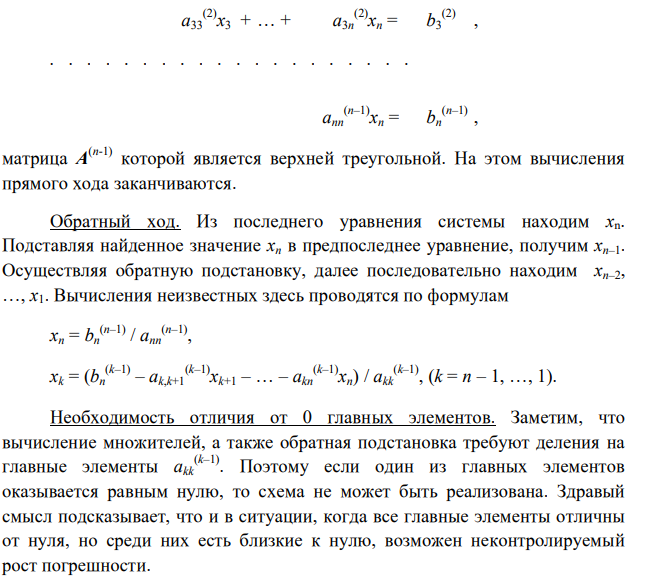
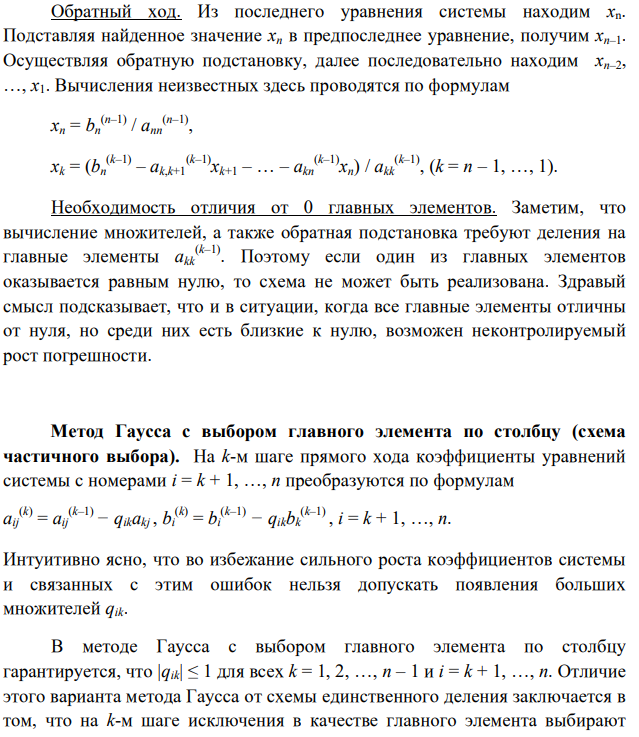
# **КРАТКИЕ ТЕОРИТИЧЕСКИЕ СВЕДЕНИЯ**

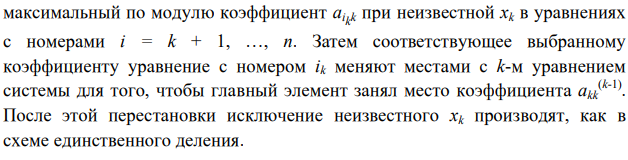
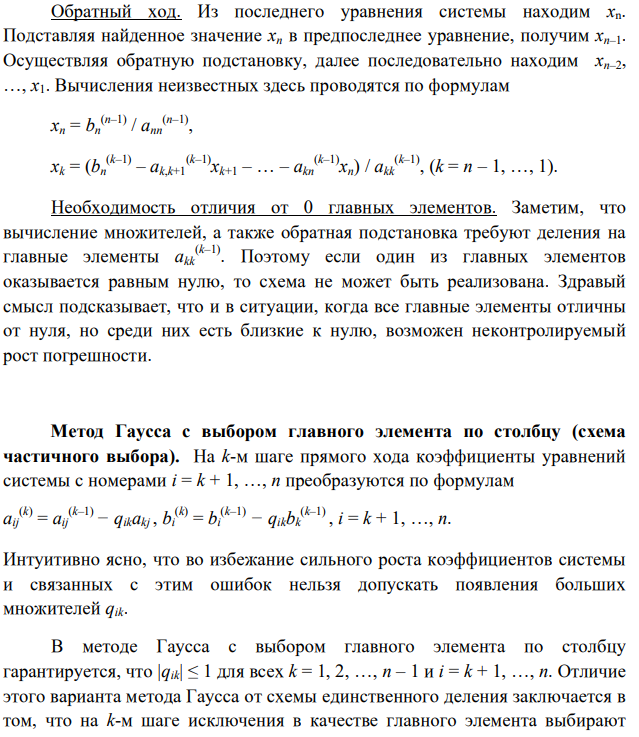
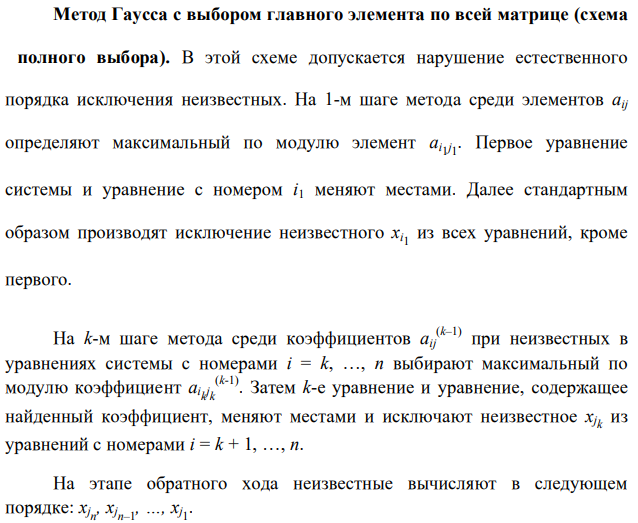
Одним из самых распространенных методов решения СЛАУ является метода Гаусса. Этот методы (который также называют *методом последовательного исключения неизвестных*) широко известен в различных вариантах.

Вычисления с помощью метода Гаусса заключаются в последовательном исключении неизвестных из системы для преобразования ее к эквивалентной системе с верхней треугольной матрицей. Вычисления значений неизвестных производят на этапе обратного хода.

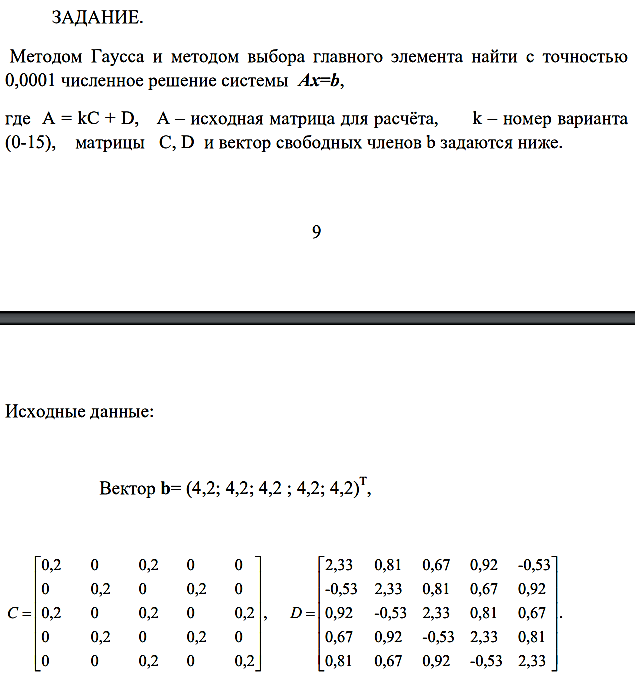
Рассмотрим сначала простейший вариант метода Гаусса, называемый *схемой единственного деления*.







# **ЗАДАНИЕ**

 Вариант 11

# **ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

#include <iostream>

#include <iomanip>

#include <vector>

#include "Matrix.h"

using namespace std;

int k = 11;

double\*\* create\_C() {

double\*\* C = new double\* [5];

C[0] = new double[5]{ 0.2, 0, 0.2, 0, 0 };

C[1] = new double[5]{ 0, 0.2, 0, 0.2, 0 };

C[2] = new double[5]{ 0.2, 0, 0.2, 0, 0.2 };

C[3] = new double[5]{ 0, 0.2, 0, 0.2, 0 };

C[4] = new double[5]{ 0, 0, 0.2, 0, 0.2 };

return C;

}

double\*\* create\_D() {

double\*\* D = new double\* [5];

D[0] = new double[5]{ 2.33, 0.81, 0.67, 0.92, -0.53 };

D[1] = new double[5]{ -0.53, 2.33, 0.81, 0.67, 0.92 };

D[2] = new double[5]{ 0.92, -0.53, 2.33, 0.81, 0.67 };

D[3] = new double[5]{ 0.67, 0.92, -0.53, 2.33, 0.81 };

D[4] = new double[5]{ 0.81, 0.67, 0.92, -0.53, 2.33 };

return D;

}

void print\_roots(const vector<double>& vec, string text, int w, int precision) {

if (text == "\n")

cout << "\n\n";

else

cout << "\n" << text << "\n";

for (const auto& i : vec)

cout << fixed << setw(w) << setprecision(precision) << i << "\n";

}

int main()

{

setlocale(LC\_ALL, "Ru");

Matrix<double> C(create\_C(), 5, 5);

Matrix<double> D(create\_D(), 5, 5);

Matrix<double> A = k \* C + D;

vector<double> d = { 4.2, 4.2, 4.2, 4.2, 4.2 };

A = A + d;

A.print("Исходная матрица");

type1(A); // Схема единственного деления

type2(A); // Выбор главного элемента по столбцу

type3(A); // Выбор главного элемента по всей матрице

}

double eps = 0.0000000001;

void type1(Matrix<double> A) {

double methodIsUseful = true;

for (int k = 0; k < A.getRows() - 1; k++) {

for (int i = 1 + k; i < A.getRows() && methodIsUseful; i++) {

if (abs(A.at(k, k)) < eps)

methodIsUseful = false;

double q = A.at(i, k) / A.at(k, k);

for (int j = k; j < A.getColumns(); j++)

A.at(i, j) -= A.at(k, j) \* q;

}

}

string methodName = "Схема единственного деления";

if (methodIsUseful) {

A.print(methodName);

vector<double> roots(A.getRows());

for (int k = A.getRows() - 1; k >= 0; k--) {

double difference = A.at(k, A.getColumns() - 1);

for (int i = A.getRows() - 1; i > k; i--)

difference -= A.at(k, i) \* roots[i];

roots[k] = difference / A.at(k, k);

}

print\_roots(roots, "Корни");

}

else {

cout << methodName << "\n";

cout << "Данный метод не применим для матрицы с данными коэффициентами\n";

}

}

void type2(Matrix<double> A) {

for (int k = 0; k < A.getRows() - 1; k++) {

A.swapRows(A.rowOfMaxItemInColumnFromPoint(k, k), k);

for (int i = 1 + k; i < A.getRows(); i++) {

double q = A.at(i, k) / A.at(k, k);

for (int j = k; j < A.getColumns(); j++)

A.at(i, j) -= A.at(k, j) \* q;

}

}

A.print("Выбор главного элемента по столбцу");

vector<double> roots(A.getRows());

for (int k = A.getRows() - 1; k >= 0; k--) {

double difference = A.at(k, A.getColumns() - 1);

for (int i = A.getRows() - 1; i > k; i--)

difference -= A.at(k, i) \* roots[i];

roots[k] = difference / A.at(k, k);

}

print\_roots(roots, "Корни");

}

void type3(Matrix<double> A) {

map<int, int> rootsOrder;

for (int i = 0; i < A.getRows(); i++)

rootsOrder[i] = i;

for (int k = 0; k < A.getRows() - 1; k++) {

Point maxItem = A.posOfMaxItemInMatrixFromPoint(k, k);

A.swapRows(maxItem.row, k);

A.swapColumns(maxItem.column, k);

rootsOrder[maxItem.column] = k;

rootsOrder[k] = maxItem.column;

for (int i = 1 + k; i < A.getRows(); i++) {

double q = A.at(i, k) / A.at(k, k);

for (int j = k; j < A.getColumns(); j++)

A.at(i, j) -= A.at(k, j) \* q;

}

}

A.print("Выбор главного элемента по всей матрице");

vector<double> roots(A.getRows());

for (int k = A.getRows() - 1; k >= 0; k--) {

double difference = A.at(k, A.getColumns() - 1);

for (int i = A.getRows() - 1; i > k; i--)

difference -= A.at(k, i) \* roots[i];

roots[k] = difference / A.at(k, k);

}

vector<double> rootsInOrder(A.getRows());

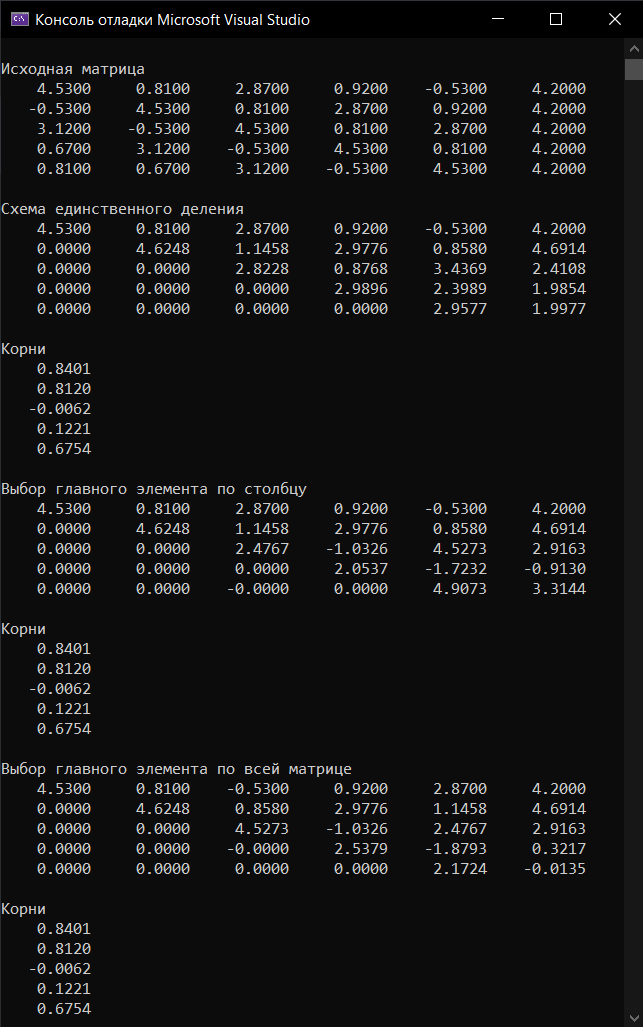
for (int i = 0; i < roots.size(); i++)

rootsInOrder[i] = roots[rootsOrder[i]];

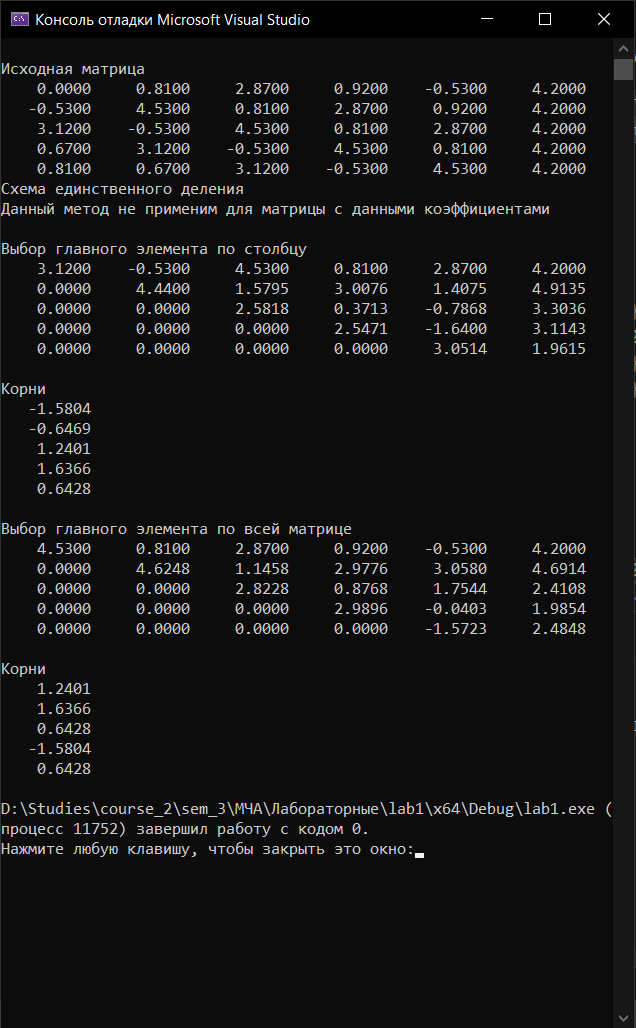
print\_roots(rootsInOrder, "Корни");

}

# **ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ**

 Результат работы программы с тестовым примером

Приведенный выше тестовый пример можно считать неудачным, так как она дает одинаковые результаты для всех методов решений. Это таковым не является в том случае, если главный элемент на любом этапе равен нулю или близок к нулю, так как это приводит к неконтролируемому росту погрешности.

 Пример с главным элементом, равном нулю, на первом шаге k = 1 (). Программа обрабатывает данный случай.

# **ОЦЕНКА**

- значение, полученное по заданию с точностью .

– более точное () значение, выдаваемое компьютером.

=0.0000096181

0.0000458189

# **ВЫВОД**

В ходе лабораторной работы был применен метод Гаусса для решения СЛАУ в трех вариантах: схема единственного деления, а также схемы частичного и полного выборов. Для решения поставленной задачи тремя способами была создана программа на языке C++, а также произведена оценка.

Также можно сделать вывод о том, что схема единственного деления может привести к неконтролируемому росту погрешность в случае, если на каком-либо шаге главный элемент будет равным нулю или близким к нему.