eМинистерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное   
учреждение высшего образования

Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

**Отчет по лабораторной работе**

**«Вычисление арифметических выражений»**

**Выполнил**:

студент/ка группы 382003-1

Ларионов Алексей Игоревич

**Проверил**:

ассистент каф. МОСТ,

Волокитин В.Д.

Нижний Новгород

2021

**Содержание**

[Постановка задачи 3](#_Toc26962562)

[Метод решения 4](#_Toc26962563)

[Руководство пользователя 5](#_Toc26962564)

[Описание программной реализации 6](#_Toc26962565)

[Подтверждение корректности 7](#_Toc26962566)

[Результаты экспериментов 8](#_Toc26962567)

[Заключение 9](#_Toc26962568)

[Приложение 10](#_Toc26962569)

# Постановка задачи

Необходимо реализовать метод Гаусса (Жордана-Гаусса) для квадратной матрицы произвольного размера.

# Метод решения

Функция J\_G – основная функция в реализации алгоритма Жордана-Гаусса. При вызове этой функции над исходной матрицей начиная с первого столбца производятся такие операции:

1) Происходит проверка знака элементов столбца, если элемент отрицателен, то строка, содержащая его, поэлементно умножается на (-1).

2) Цикл for сравнивает элемент главной диагонали с каждым элементом стобца. Если элемент столбца оказывается больше вызывается функция swap(), которая меняет местами строку с бывшим ведущим элементом и строку с элементом который оказался больше.

3) Вызывается функция nullify\_row(), которая зануляет элементы ниже и выше главной диагонали.

4) Элементы главной диагонали приводятся к 1 путем деления строки на ведущий элемент.

5) Элементы квадратной матрицы сравниваются по модулю с достаточно малым числом 0.1e-14 если они оказываются меньше – клетка зануляется. Это реализовано чтобы избавиться от погрешностей появившихся в ходе арифметических операций.

Функция swap() принимает два аргумента типа size\_t – номера строк которые нужно поменять местами. В ней же создается временный массив temp. В цикле for элементы первой строки копируются в temp, элементы второй строки копируются в первую, элементы из temp копируются во вторую строку.

Функция nullify\_row() принимает один аргумент size\_t - номер строки и столбца выбранного ведущего элемента. Эта функция проверяет равен ли ведущий элемент нолю. Если равен – ничего не происходит, иначе проверяет отрицателен элемент или нет, если элемент отрицателен – умножает строку на (-1) поэлементно. Далее идет зануление строк ниже главной диагонали. Для этого проверяется чтобы элемент в столбце ведущего был больше нуля, если он нулевой то со строкой ничего не происходит. Если он отрицателен, вся строка помножается на (-1). Далее в цикле for для каждой строки ниже главной диагонали временной переменной temp присваивается результат деления элемента текущей строки, текущего столбца на ведущий элемент и из текущей строки поэлементно вычитаются элементы строки, включающей в себя ведущий элемент, помноженные на temp. Аналогично для элементов выше главной диагонали.

Функция print\_answer() - выводит столбец последний столбец матрицы.

Функция test() - с помощью цикла for и счетчика проверяет есть ли в матрице нулевые строки – если есть выводит сообщение “ Бесконечно много решений.”, а если есть строки в которых все элементы равны нулю кроме последнего – выводит сообщение "Система не совместна."

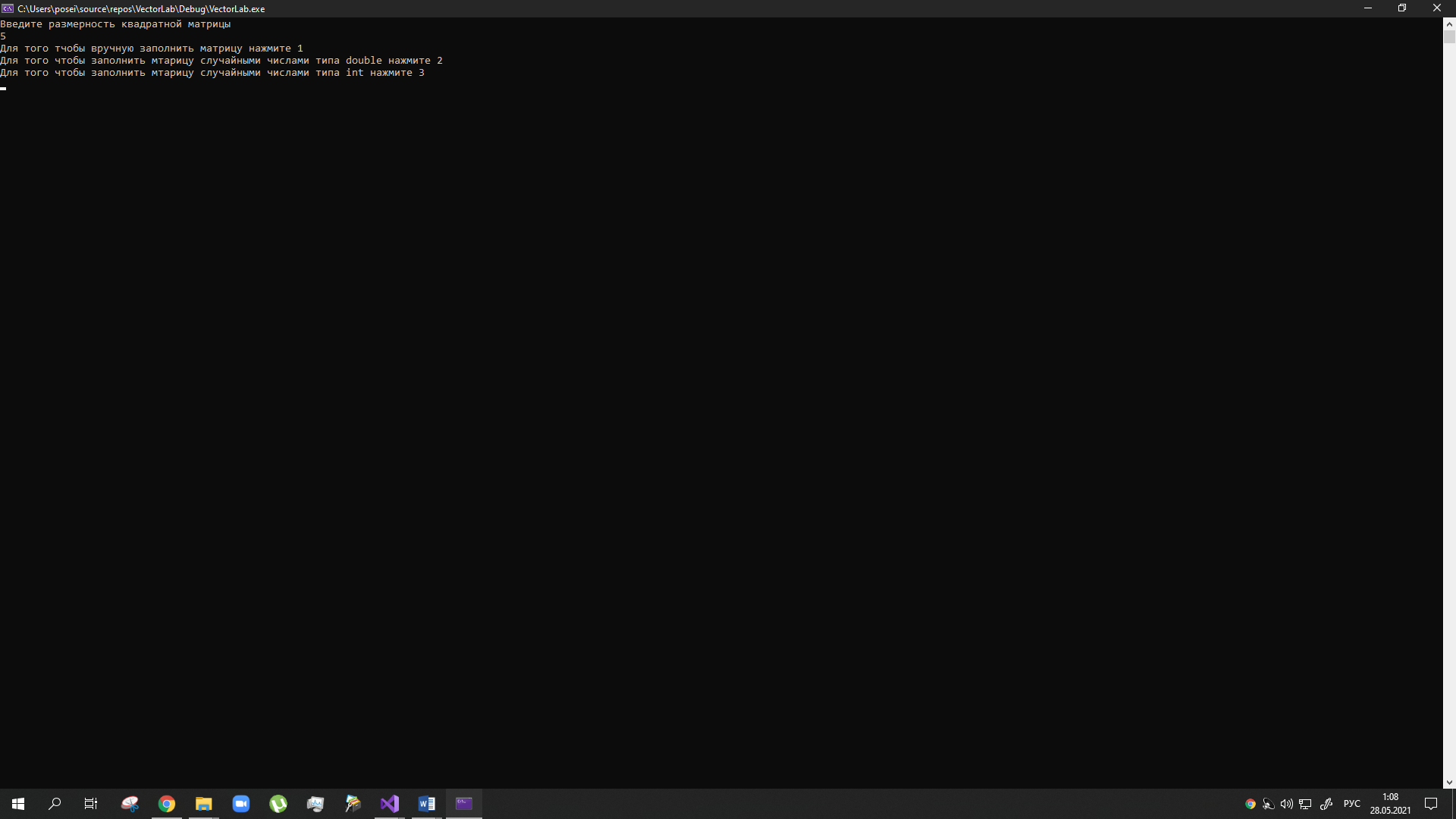
Функция main() запрашивает ввести размерность матрицы после чего создается расширенная матрица для заданной размерности – квадратная матрица + 1 столбец. Далее предлагается выбор из трех вариантов наполнения матрицы:

1. Ручной ввод элементов матрицы, в этом варианте расчёты производятся в типе данных double. В этом случае пользователь должен ввести коэффициенты построчно.
2. Наполнение матрицы случайными элементам типа double.
3. Наполнение матрицы случайными элементами типа int.

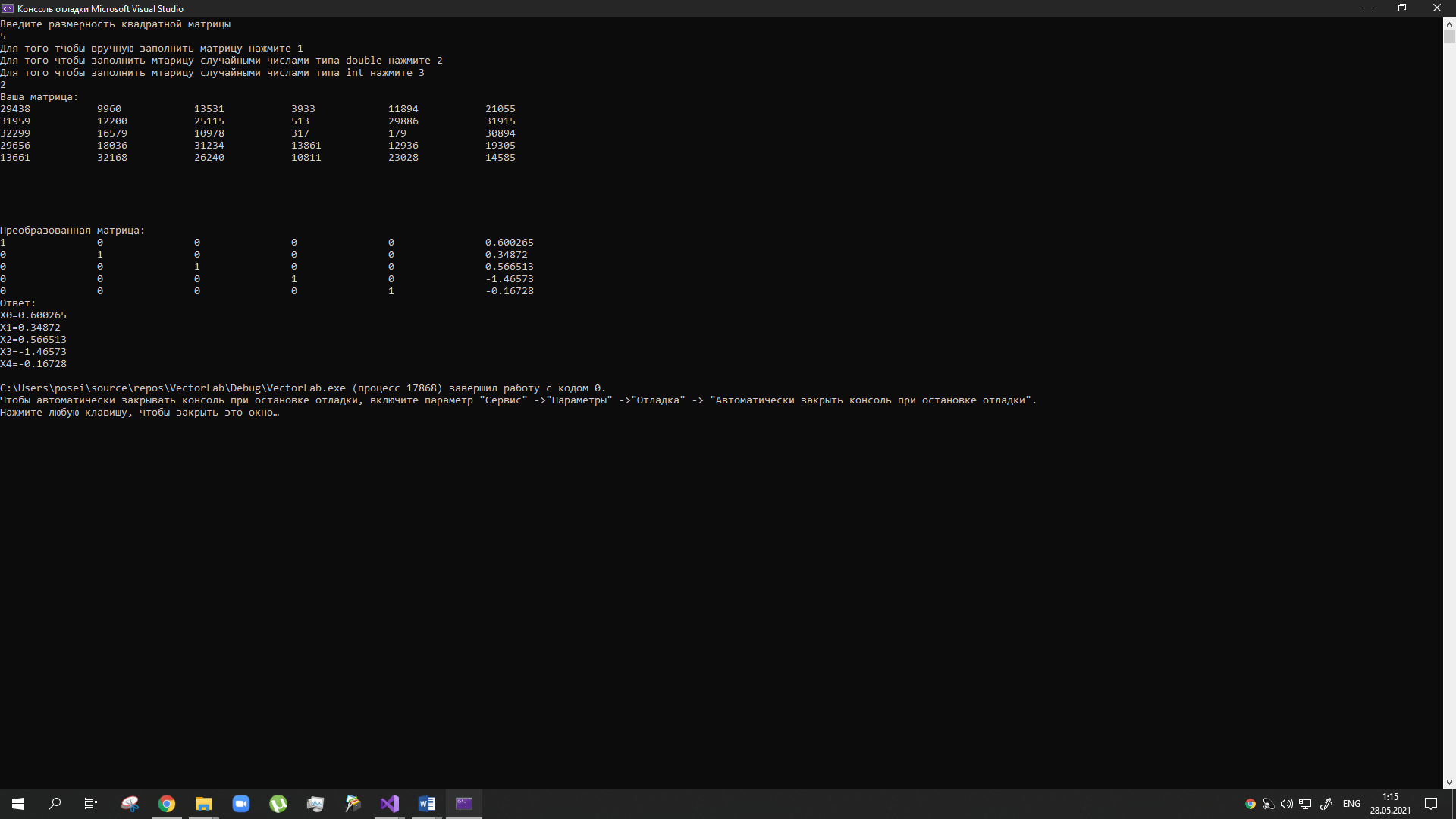
Для выбранного варианта наполнения матрицы применяется функция printM(), которая выводит на экран матрицу в ее изначальном виде, далее к матрице применяется функция J\_G – собственно сам алгоритм Жордана-Гаусса, затем функцией printM() на экран выводится решенная матрица, к ней применяется функция checkM() – если есть единственные решения – выводится стобец решений, в других случаях выводится сообщение о том что система имеет бесконечно много решений или несовместна.

# Руководство пользователя

Сначала необходимо ввести размерность квадратной матриц которую вы хотите решить, затем выбрать один из трех вариантов наполнения:



Затем выводятся исходная матрица, преобразованная матрица и один из возможных исходов решения:



# Описание программной реализации

В проекте VectorLab содержится два файла VectorLav.cpp – реализация функции main(), и matrix\_templates.h -реализация шаблонов вектора и матрицы, а так же остальных использующихся в проекте функций.

Шаблонный класс вектор (vector), представляeт из себя класс динамического массива с соответствующими методами и перегруженными операциями.

Шаблонный класс матрица (matrix) – наследник класса вектора от векторов.

Функция J\_G – выполняет над матрицей алгоритм Жордана-Гаусса.

Функция swap – меняет местами две строки в матрице.

Функция nullify\_raw – зануляет стобец выше и ниже ведущего жлемента.

Функции fill\_manually и fill\_randomly – заполняют матрицу элементами – введенными вручную и случайными соответственно.

Функция printM – выводит матрицу на экран.

Функция checkM – проверяет систему на совместность и на бесконечные решения.

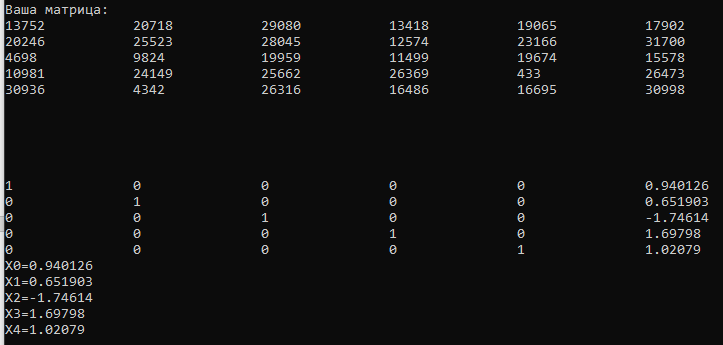
Функция printAnswer – выводит последний столбец матрицы, который в случае совместности системы является ответом.

Функция main – запрашивает размерность матрицы, способ заполнения, элементы в случае заполнения вручную. Функция J\_G решает систему, функция checkM проверяет систему на совместность и бесконечные решения. Функция printAnswer – выводит ответ.

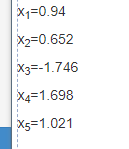
# Подтверждение корректности

Программа была проверена в сравнении с доступными онлайн калькуляторами.

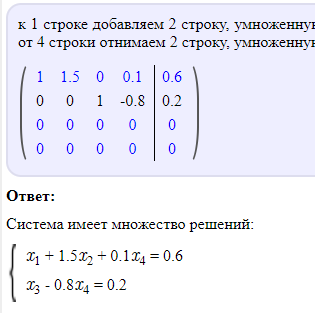
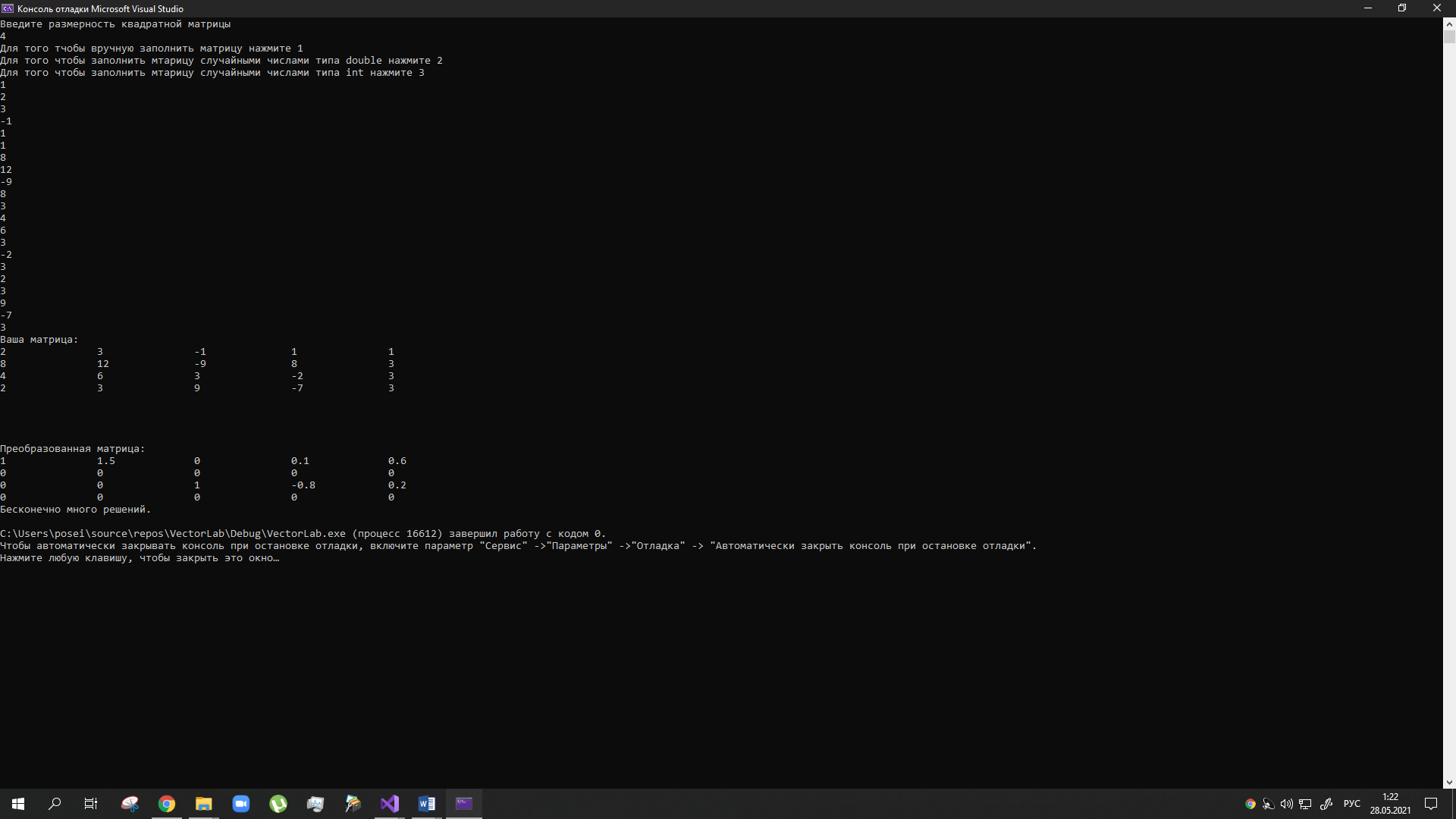
Моя программа:

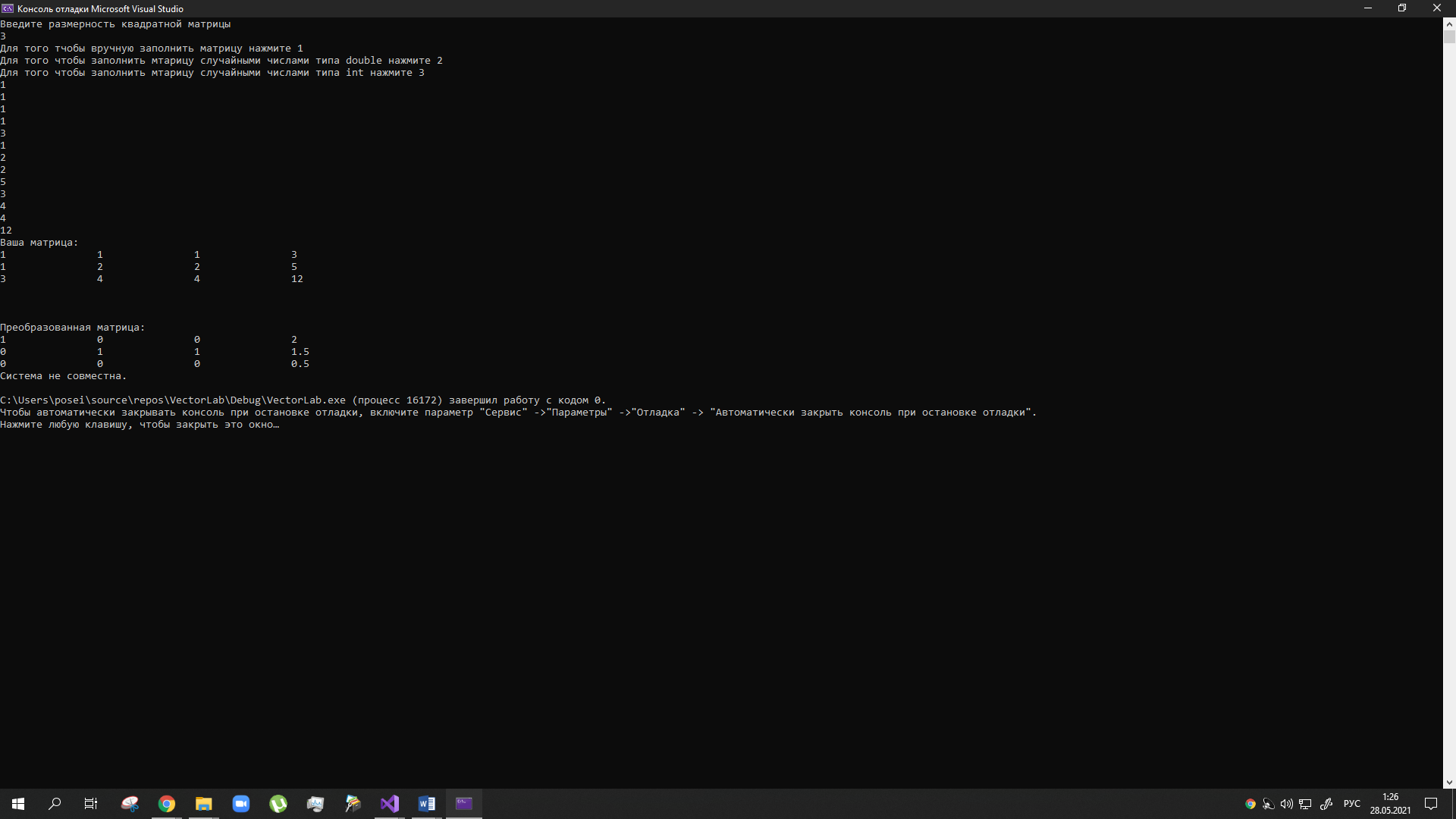
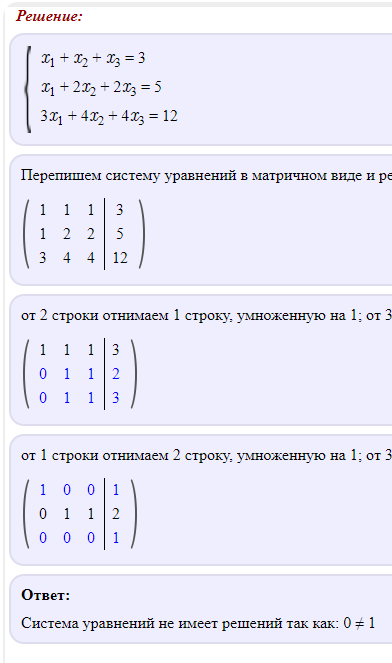


Онлайн калькулятор https://math.semestr.ru/gauss/jordan.php:



Так же я проверил программу на примерах несовместных и систем с бесконечным количеством решений:



Для проверки использовался калькулятор: https://ru.onlinemschool.com/math/assistance/equation/gaus/

# Результаты экспериментов

По данным экспериментов видно, что программа показывает довольно высокую степень точности в вычислениях. Более высокой точности возможно достичь, если приводить принимаемые значения к рациональным числам, проводить все операции с ними и выводить их в качестве ответа. Но для этого нужно реализовать класс рациональных чисел, а это не является обязательным условием выполнения работы.

К тому же программа работает довольно быстро: меньше секунды для создания и решения матрицы случайных чисел double 10\*10, ~11 секунд для размерности 100\*100, ~34 сек для размерности 200\*200.

# Заключение

Программа без ошибок выполняет свою основную задачу, при этом с удовлетворительной точностью.

# Приложение

