Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Институт информационных технологий

Факультет компьютерных технологий

Кафедра информационных систем и технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

«РЕШЕНИЕ СИСТЕМЫ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ МЕТОДОМ КВАДРАТНОГО КОРНЯ»

Студент:

гр. 281079 Веремейчик А.О.

Руководитель:

Ассистент каф. ИСиТ Болтак С.В.

Минск 2023

**АННОТАЦИЯ**

Целью выполнения курсовой работы является углубление теоретических знаний и практических навыков в области алгоритмизации задач и разработки программных средств, развитие навыков самостоятельного изучения предметной области поставленной задачи, приобретение опыта применения современной вычислительной техники для решения практических задач, приобретение практических навыков и компетенций в сфере профессиональной деятельности.

При выполнении курсовой работы необходимо было реализовать и спроектировать алгоритм решения задачи, составить пояснительную записку и описать алгоритм графическим способом.

Средством для выполнения курсовой работы является любое компьютерное устройство с установленным программным обеспечением для работы с текстом, кодом программы и блок-схемами.

Задача решена с использованием среды разработки Visual Studio 2019 и языка С++.

Итоговая пояснительная записка содержит введение, моделирование, программную реализацию и тестирование программного средства, а также блок-схему программы и алгоритма.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc135001953)

[1 МОДЕЛИРОВАНИЕ 5](#_Toc135001954)

[1.1. Описание метода прогонки для решения систем линейных алгебраических уравнений 5](#_Toc135001955)

[1.2. Оценка сложности алгоритма 7](#_Toc135001956)

[2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 10](#_Toc135001957)

[3 ТЕСТИРОВАНИЕ, ПРОВЕРКА РАБОТОСПОСОБНОСТИ И АНАЛИЗ РЕЗУЛЬТАТОВ 12](#_Toc135001958)

[ВЫВОДЫ 15](#_Toc135001959)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 16](#_Toc135001960)

# **ВВЕДЕНИЕ**

Прогонкой называется метод Томаса для решения систем линейных алгебраических уравнений с трехдиагональной матрицей. Если матрица системы обладает определенными свойствами, то метод прогонки является численно устойчивым и очень эффективным методом, который позволяет практически мгновенно решать одномерные краевые задачи, одну из которых мы рассмотрели в предыдущем разделе. Большинство корректно поставленных физических задач приводит к системе уравнений с хорошей матрицей, и в этих случаях метод прогонки проявляет слабую чувствительность как к погрешностям задания начальных условий, так и к погрешностям вычислительного характера.

Большие программы из-за своей сложности нередко содержат ошибки, которые могут стать причиной материального ущерба. В результате борьбы с проблемой сложности программного кода были выработаны три новые концепции программирования:

— объектно-ориентированное программирование (ООП);

— унифицированный язык моделирования (UML);

— специализированные средства разработки программного обеспечения.

Из всех объектно-ориентированных языков С++ является одним из широко используемых. C++ – компилируемый, статически типизированный язык программирования общего назначения.

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр. Существует множество реализаций языка C++, как бесплатных, так и коммерческих и для различных платформ. Например, на платформе x86 это GCC, Visual C++, Intel C++ Compiler, Embarcadero (Borland) C++ Builder и другие.

Темой курсового проекта является создание программного средства «Решение системы линейных алгебраических уравнений методом прогонки».

Цели курсового проекта:

* закрепить, углубить и систематизировать теоретические знания и практические умения, полученные при изучении дисциплины ОАиП;
* привить навыки самостоятельной разработки программ с помощью языка программирования С++;
* закрепить навыки разработки удобного интерфейса пользователя, системы меню, навигации, системы помощи с использованием различных сред программирования;
* привить навыки самостоятельной разработки приложений с помощью Microsoft Visual Studio;

# **МОДЕЛИРОВАНИЕ**

# **1.1. Описание метода прогонки для решения систем линейных алгебраических уравнений**

Метод прогонки - это численный метод решения систем линейных алгебраических уравнений (СЛАУ), который применяется для решения трехдиагональных матриц. Трехдиагональная матрица - это матрица, у которой все элементы, кроме элементов на трех диагоналях, равны нулю.

Метод прогонки начинается с приведения исходной системы уравнений к трехдиагональному виду путем применения последовательности прямых и обратных ходов. Затем решение системы производится путем прогонки - прохода по матрице в прямом и обратном направлениях, при котором на каждом шаге вычисляются коэффициенты прогонки и обратной прогонки.

Процесс прогонки начинается с прямого хода, при котором вычисляются коэффициенты прогонки и обновляются значения вектора решения. Затем производится обратный ход, при котором вычисляются коэффициенты обратной прогонки и обновляются значения вектора решения. Коэффициенты прогонки и обратной прогонки вычисляются рекурсивно, что позволяет сократить количество операций и ускорить процесс вычисления решения.

Метод прогонки имеет высокую точность и скорость сходимости, поэтому он широко применяется в различных областях, например, в теории упругости, теплофизике, гидродинамике и других областях науки и техники.

Для более подробного описания алгоритма метода прогонки, можно рассмотреть его на примере системы линейных уравнений вида:

a[1]\*x[1] + b[1]\*x[2] = c[1]

c[i-1]\*x[i-1] + a[i]\*x[i] + b[i]\*x[i+1] = c[i], 2 <= i <= n-1

c[n-1]\*x[n-1] + a[n]\*x[n] = c[n]

где a, b, c - заданные коэффициенты, x - неизвестные.

Первый шаг метода - прямой ход. На этом этапе вычисляются коэффициенты прогонки alpha и beta для каждого уравнения системы, а также обновляются значения вектора решения x.

Прямой ход:

alpha[1] = -b[1]/a[1]

beta[1] = c[1]/a[1]

for i in range(2, n):

m = a[i] + c[i-1]\*alpha[i-1]

alpha[i] = -b[i]/m

beta[i] = (c[i] - c[i-1]\*beta[i-1])/m

x[n] = (c[n] - c[n-1]\*beta[n-1])/(a[n] + c[n-1]\*alpha[n-1])

for i in range(n-1, 0, -1):

x[i] = alpha[i]\*x[i+1] + beta[i]

Второй шаг - обратный ход. На этом этапе вычисляются коэффициенты обратной прогонки alpha и beta для каждого уравнения системы, а затем обновляются значения вектора решения x.

Обратный ход:

alpha[n] = 0

beta[n] = (c[n] - c[n-1]\*beta[n-1])/(a[n] + c[n-1]\*alpha[n-1])

for i in range(n-1, 0, -1):

x[i] = alpha[i]\*x[i+1] + beta[i]

alpha[i] = -b[i]/(a[i] + c[i]\*alpha[i+1])

beta[i] = (c[i] - a[i]\*beta[i+1])/(a[i] + c[i]\*alpha[i+1])

x[1] = alpha[1]\*x[2] + beta[1]

Таким образом, после выполнения обратного хода, получаем решение системы линейных уравнений в виде вектора x.

В методе прогонки на каждом шаге вычисляются два коэффициента: коэффициент прогонки alpha и коэффициент обратной прогонки beta. Коэффициент прогонки alpha вычисляется по следующей формуле:

alpha[i] = -b[i]/(a[i] + c[i]\*alpha[i+1])

где a, b, c - заданные коэффициенты, i - номер уравнения в системе.

Коэффициент обратной прогонки beta вычисляется по формуле:

beta[i] = (c[i] - a[i]\*beta[i+1])/(a[i] + c[i]\*alpha[i+1])

где a, b, c - заданные коэффициенты, i - номер уравнения в системе.

На первом шаге прямого хода вычисляются начальные значения коэффициентов прогонки и обратной прогонки:

alpha[1] = -b[1]/a[1]

beta[1] = c[1]/a[1]

Затем на каждом последующем шаге прямого хода вычисляются новые значения коэффициентов прогонки и обратной прогонки:

m = a[i] + c[i-1]\*alpha[i-1]

alpha[i] = -b[i]/m

beta[i] = (c[i] - c[i-1]\*beta[i-1])/m

где m - промежуточное значение, используемое для вычисления новых значений коэффициентов.

На обратном ходе сначала вычисляется начальное значение коэффициента обратной прогонки для последнего уравнения системы:

beta[n] = (c[n] - c[n-1]\*beta[n-1])/(a[n] + c[n-1]\*alpha[n-1])

Затем на каждом последующем шаге обратного хода вычисляются новые значения коэффициентов обратной прогонки:

alpha[i] = -b[i]/(a[i] + c[i]\*alpha[i+1])

beta[i] = (c[i] - a[i]\*beta[i+1])/(a[i] + c[i]\*alpha[i+1])

После вычисления всех коэффициентов на прямом и обратном ходах, получаем решение системы линейных уравнений в виде вектора x.

Метод прогонки является очень эффективным и быстрым методом решения трехдиагональных систем линейных уравнений. Это связано с тем, что он применяет рекурсивные формулы для вычисления коэффициентов прогонки и обратной прогонки, что позволяет сократить количество операций и ускорить процесс вычисления решения.

Кроме того, метод прогонки имеет высокую точность и скорость сходимости, что делает его популярным в различных областях науки и техники, где требуется решать системы линейных уравнений. Например, метод прогонки широко применяется в теории упругости, теплофизике, гидродинамике и других областях.

Однако метод прогонки применим только к трехдиагональным матрицам. Если матрица не является трехдиагональной, то для ее решения необходимо использовать другие методы, например, метод Гаусса или метод LU-разложения.

Важно отметить, что метод прогонки может быть чувствителен к ошибкам округления, особенно при работе с матрицами, содержащими очень маленькие или очень большие значения. Поэтому при реализации метода прогонки необходимо учитывать возможность потери точности и принимать меры для ее минимизации, например, использовать повышенную точность при вычислениях или применять методы устранения особенностей матрицы.

# **1.2. Оценка сложности алгоритма**

Алгоритм прогонки (также называемый методом Томаса) - это эффективный способ решения линейной системы уравнений с трехдиагональной матрицей. Сложность алгоритма прогонки может быть оценена как O(n), где n - размерность матрицы.

Алгоритм прогонки состоит из двух фаз. На первой фазе вычисляются коэффициенты прогонки - прямой ход метода Гаусса применяемый к трехдиагональной матрице. На второй фазе решение находится обратным ходом метода Гаусса.

На первой фазе требуется выполнить n операций для прямого хода. На второй фазе также требуется выполнить n операций для обратного хода. Таким образом, общее количество операций равно 2n, что дает сложность O(n).

Таким образом, алгоритм прогонки имеет линейную сложность относительно размерности матрицы и может быть эффективно использован для решения больших систем уравнений.

# **2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

Методы проектирования алгоритмов включают: нисходящее проектирование, модульность, структурное программирование. Нисходящее проектирование предполагает последовательное разбиение исходной задачи на подзадачи до такой конкретизации, когда подзадача сможет быть реализована одним оператором выбранного для программирования языка. По ходу нисходящего проектирования та или иная подзадача может сформировать самостоятельный модуль. Тогда может быть применен принцип модульного программирования. Он обеспечивает легкость составления алгоритмов и отладки программ, легкость сопровождения и модификации, а также возможность одновременной разработки различных модулей разными специалистами с использованием разных языков программирования. При работе над модулем можно применить принцип структурного программирования. Его цель - повышение читабельности и ясности алгоритма (и программы), более высокой производительности программистов и упрощение отладки.

Данный курсовой проект был реализован на языке программирования C++ в среде разработки Microsoft Visual Studio.

В программе используется стандартная библиотека <iostream>, которая объявляет объекты, управляющие чтением из стандартных потоков и записью в них, так же использована библиотека <math.h>, это стандартная библиотека языка программирования С++, разработанная для выполнения простых математических операций. Использованы такие типы переменных, как int (для целочисленных значений) и float (для значений с плавающей запятой). Также использованы функция setlocale – для установления языкового стандарта программы, cin – для ввода значений переменных с клавиатуры, cout – для вывода значений и текста на экран. Использованы условный оператор if, который управляет условным ветвлением, цикл for, который позволяет повторить выполнение того или иного оператора или составного оператора заданное число раз.

При запуске программы выводиться список меню, и возможность ввода числа для выбора пункта меню. Меню состоит из таких выборов как ручной ввод, рандом и выход.

Ручной ввод представляет собой самостоятельный выбор размера матрицы и заполнение ее числами. При заполнении всех данных происходит вывод введенной системы на экран, далее происходит вычисление системы линейных алгебраических уравнений методом прогонки и после уже конечный посчитанный результат в виде корней системы уравнения.

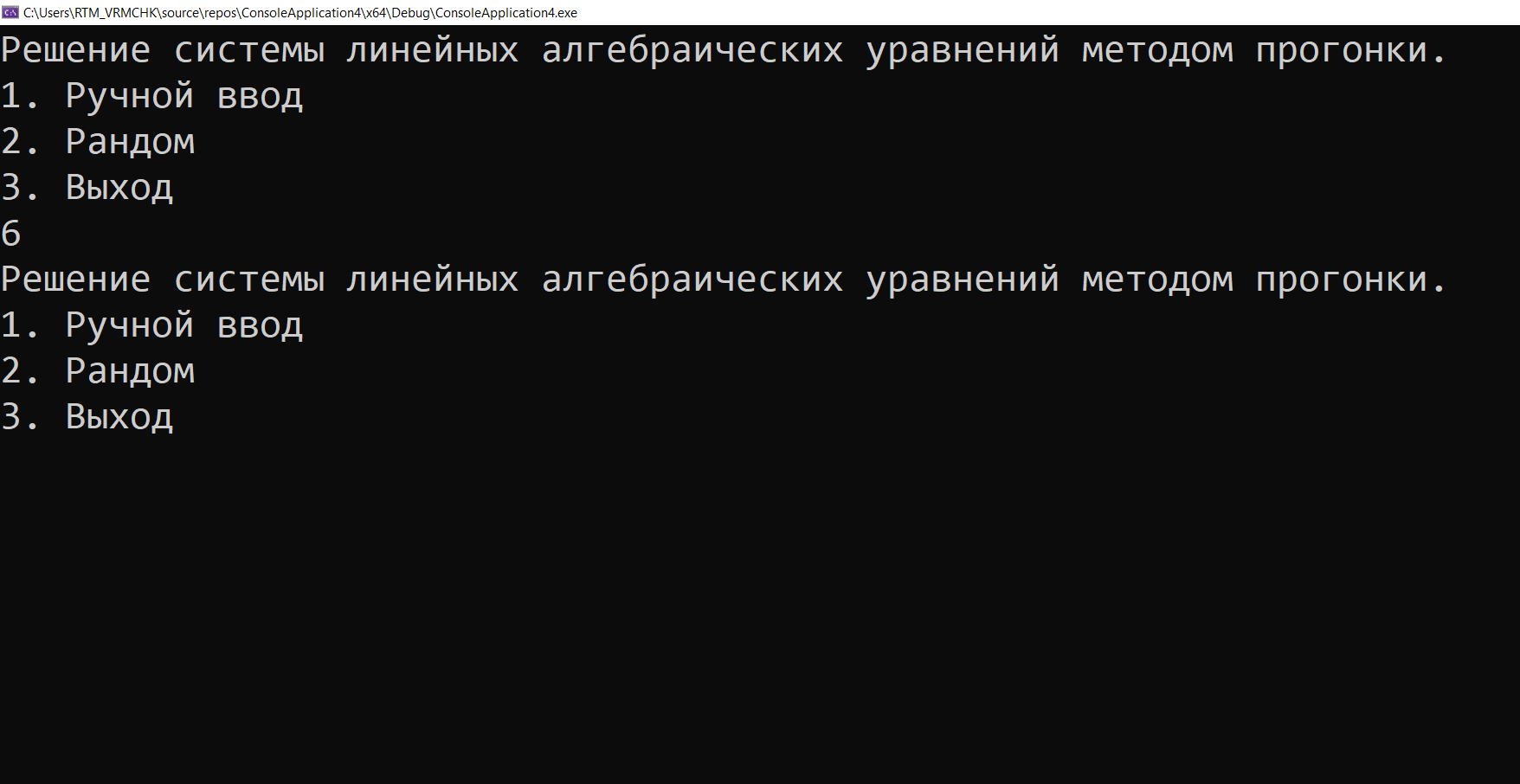
Рандом представляет собой автоматическое заполнение данных системы, но при этом пользователь выбирает сам размерность, после чего программа производит расчет и соответственно вывод результата в виде корней системы уравнения.

Выход производит полную остановку и закрытие приложения.

# **3 ТЕСТИРОВАНИЕ, ПРОВЕРКА РАБОТОСПОСОБНОСТИ И АНАЛИЗ РЕЗУЛЬТАТОВ**

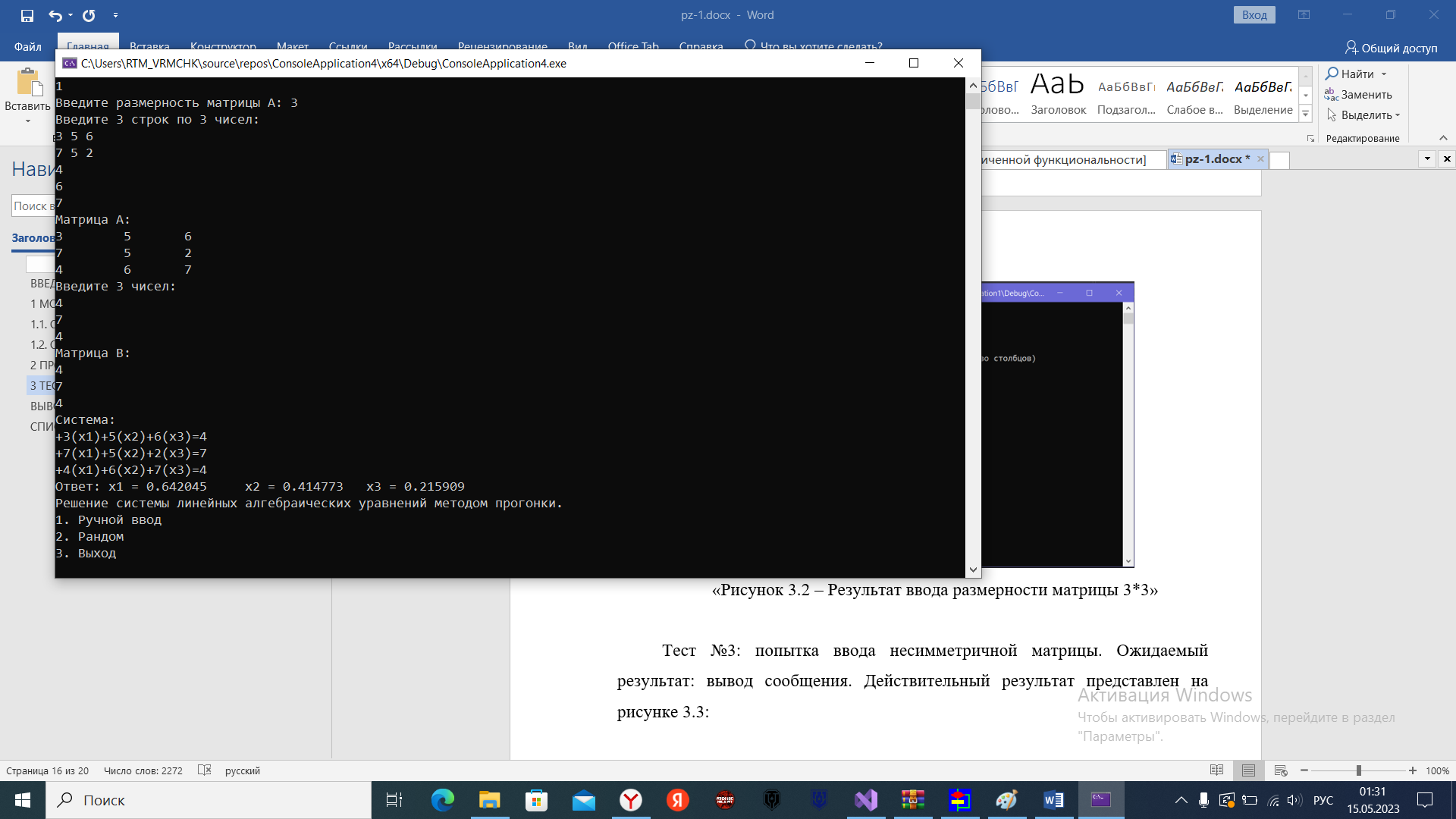
Для проверки работоспособности программы были разработаны и проведены четыре теста.

Тест №1: попытка ввода не существующего пункта меню. Ожидаемый результат: произойдет повторный вывод меню. Действительный результат представлен на рисунке 3.1:



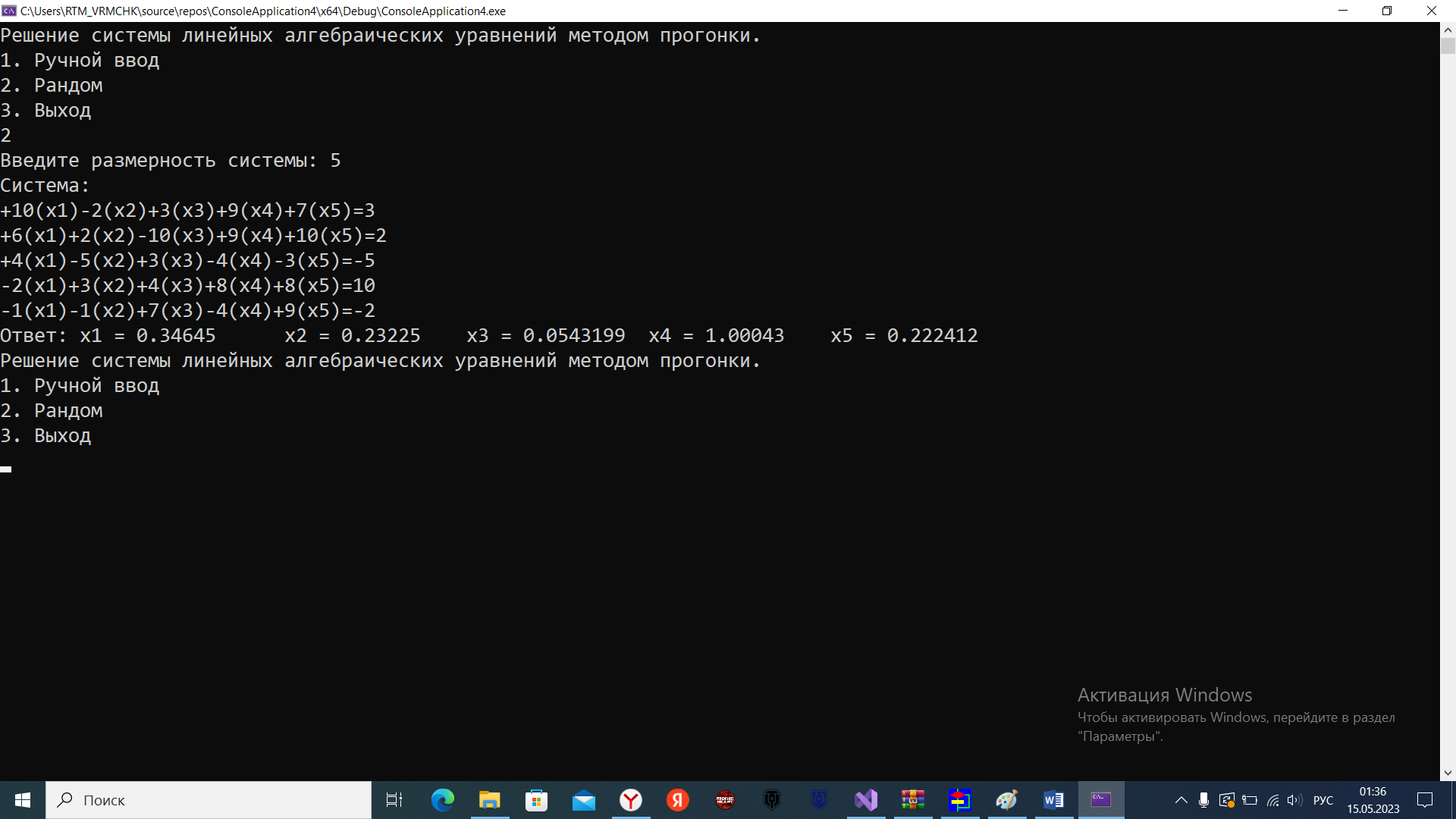
«Рисунок 3.1 – Результат ввода пункта меню»

Тест №2: попытка ввода всех элементов системы. Ожидаемый результат: вывод всех элементов системы и подсчет корней уравнения. Действительный результат представлен на рисунке 3.2:



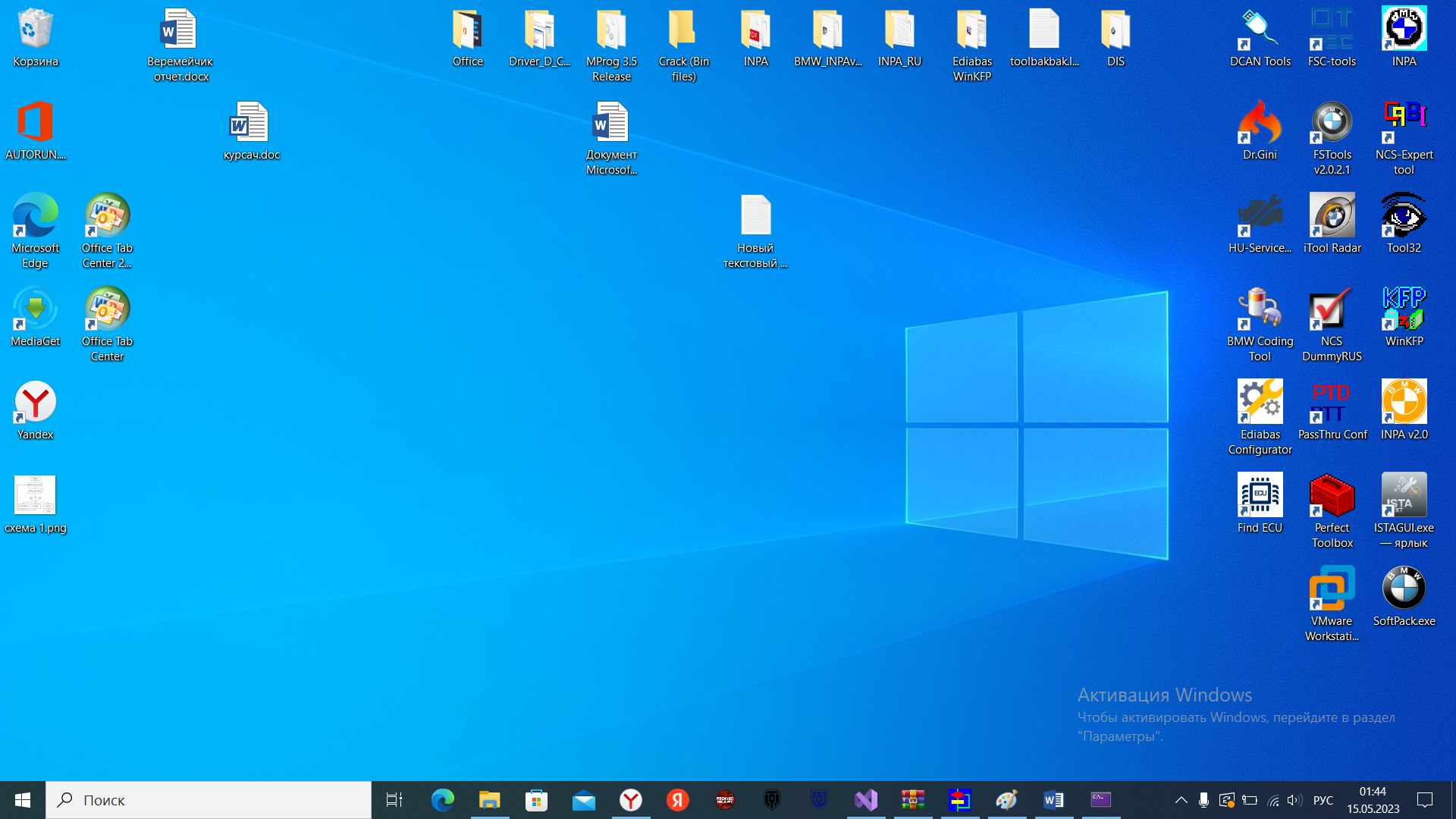
«Рисунок 3.2 – Результат ввода всех элементов системы»

Тест №3: попытка выбора пункта меню рандом и ввод размерности системы. Ожидаемый результат: Выполнение расчета системы и вывод ее на экран. Действительный результат представлен на рисунке 3.3:



«Рисунок 3.3 – Результат выбора пункта меню рандом и ввод размерности системы»

Тест №4: попытка выбора пункта меню выход. Ожидаемый результат: закрытие приложения. Действительный результат представлен на рисунке 3.4:



«Рисунок 3.4 – Результат выбора пункта выход»

# **ВЫВОДЫ**

В данной курсовой работе была реализована программа для решения системы линейных алгебраических уравнений методом прогонки посредством использования среды программирования Microsoft Visual Studio и языка C++.

В результате проделанной работы в рамках курсового проекта были усвоены знания о системах линейных алгебраических уравнений и методе проектирования алгоритма для их решения.

Разработано программное средство для решения систем линейных алгебраических уравнений методом квадратного корня.

В программе реализован следующий функционал:

1. Возможность заполнения системы вручную.
2. Возможность заполнения системы рандомно.
3. Возможность выбора пункта меню.
4. Возможность выхода из программы.
5. Решение системы линейных алгебраических уравнений.

Программа имеет удобный пользовательский интерфейс, она написана под операционную систему Windows. Работа с программой требует лишь элементарных знаний и приемов работы с ПК.

В результате выполнения курсового проекта усовершенствованы знания в области программирования консольных приложений, приобретены навыки использования научно-технической информации.

Текст основной части пояснительной записки был проверен на плагиат. Уровень уникальности текста составляет

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Павловская, Т. А. C/C++. Программирование на языке высокого уровня / Т.А. Павловская. – М.: Питер, 2013. – 464 c.

2. Т Бошемин, Б. Основы .NET / Б.Бошемин. – Вильямс,2016.

3. Хабибуллин, И. Программирование на языке высокого уровня. C/C++ / И. Хабибуллин. – М.: БХВ-Петербург, **2016**. – 512 c.

4. Амосов А.А., Дубинский Ю.А., Копченова Н.В. Вычислительные методы для инженеров: Учеб. пособие. — М.: Высш. шк., 1994. — 544 с.

5. Cleverstudents // Решение систем линейных алгебраических уравнений, методы решения, примеры [Электронный ресурс]. – 2023. – Режим доступа:http://www.cleverstudents.ru/systems/solving\_systems\_of\_linear\_equations.html?ysclid=lhnuehobj3300083596 Дата доступа: 09.05.23

# **ПРИЛОЖЕНИЕ А**

**Исходный код программы**

#include "cmath"

#include "iostream"

using namespace std;

int i, n, k, n1;

double z;

double Alpha[50][50];

double Beta[50];

double eps[50];

double X[50];

double et[50];

void print()

{

for (i = 0; i < n1; i++) {

for (k = 0; k < n1; k++)

if (Alpha[i][k] != 0 && Alpha[i][k] > 0)

cout << "+" << Alpha[i][k] << "(x" << k + 1 << ")";

else if (Alpha[i][k] != 0)

cout << Alpha[i][k] << "(x" << k + 1 << ")";

cout << "=" << Beta[i] << endl;

}

}

void clay()

{

n = n1 - 1;

eps[0] = -Alpha[0][1] / Alpha[0][0];

et[0] = Beta[0] / Alpha[0][0];

for (i = 1; i < n; i++) {

z = Alpha[i][i] + Alpha[i][i - 1] \* eps[i - 1];

eps[i] = -Alpha[i][i + 1] / z;

et[i] = (Beta[i] - Alpha[i][i - 1] \* et[i - 1]) / z;

}

X[n] = (Beta[n] - Alpha[n][n - 1] \* et[n - 1]) / (Alpha[n][n] + Alpha[n][n - 1] \* eps[n - 1]);

for (i = n - 1; i >= 0; i--)

X[i] = eps[i] \* X[i + 1] + et[i];

cout << "Ответ: ";

for (i = 0; i < n1; i++)

cout << "x" << i + 1 << " = " << X[i] << "\t ";

cout << endl;

}

void ob()

{

cout << "Введите размерность матрицы А: ";

cin >> n1;

cout << "Введите " << n1 << " строк по " << n1 << " чисел:" << endl;

for (i = 0; i < n1; i++)

for (k = 0; k < n1; k++)

cin >> Alpha[i][k];

cout << "Матрица A:" << endl;

for (i = 0; i < n1; i++) {

for (k = 0; k < n1; k++)

cout << Alpha[i][k] << "\t ";

cout << endl;

}

cout << "Введите " << n1 << " чисел:" << endl;

for (i = 0; i < n1; i++)

cin >> Beta[i];

cout << "Матрица B:" << endl;

for (i = 0; i < n1; i++)

cout << Beta[i] << endl;

cout << "Система: " << endl;

clay();

}

void randi()

{

cout << "Введите размерность системы: ";

cin >> n1;

for (i = 0; i < n1; i++)

for (k = 0; k < n1; k++)

Alpha[i][k] = rand() % 21 - 10;

for (i = 0; i < n1; i++)

Beta[i] = rand() % 21 - 10;

cout << "Система: " << endl;

print();

clay();

}

int main()

{

setlocale(LC\_ALL, "ru");

int y = 0;

while (y == 0)

{

int p = 0;

cout << "Решение системы линейных алгебраических уравнений методом прогонки." << endl << "1. Ручной ввод" << endl << "2. Рандом" << endl << "3. Выход" << endl;

cin >> p;

switch (p)

{

case 1:

ob();

break;

case 2:

randi();

break;

case 3:

exit(0);

}

}

}