Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 5

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Условный оператор»

Выполнил: ст. гр. ТУУ-111

Богомолов В.Н.

Вариант №5

06.12.2024

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

\_\_\_\_\_\_\_\_\_\_\_

(дата приёмки)

Москва – 2024 г.

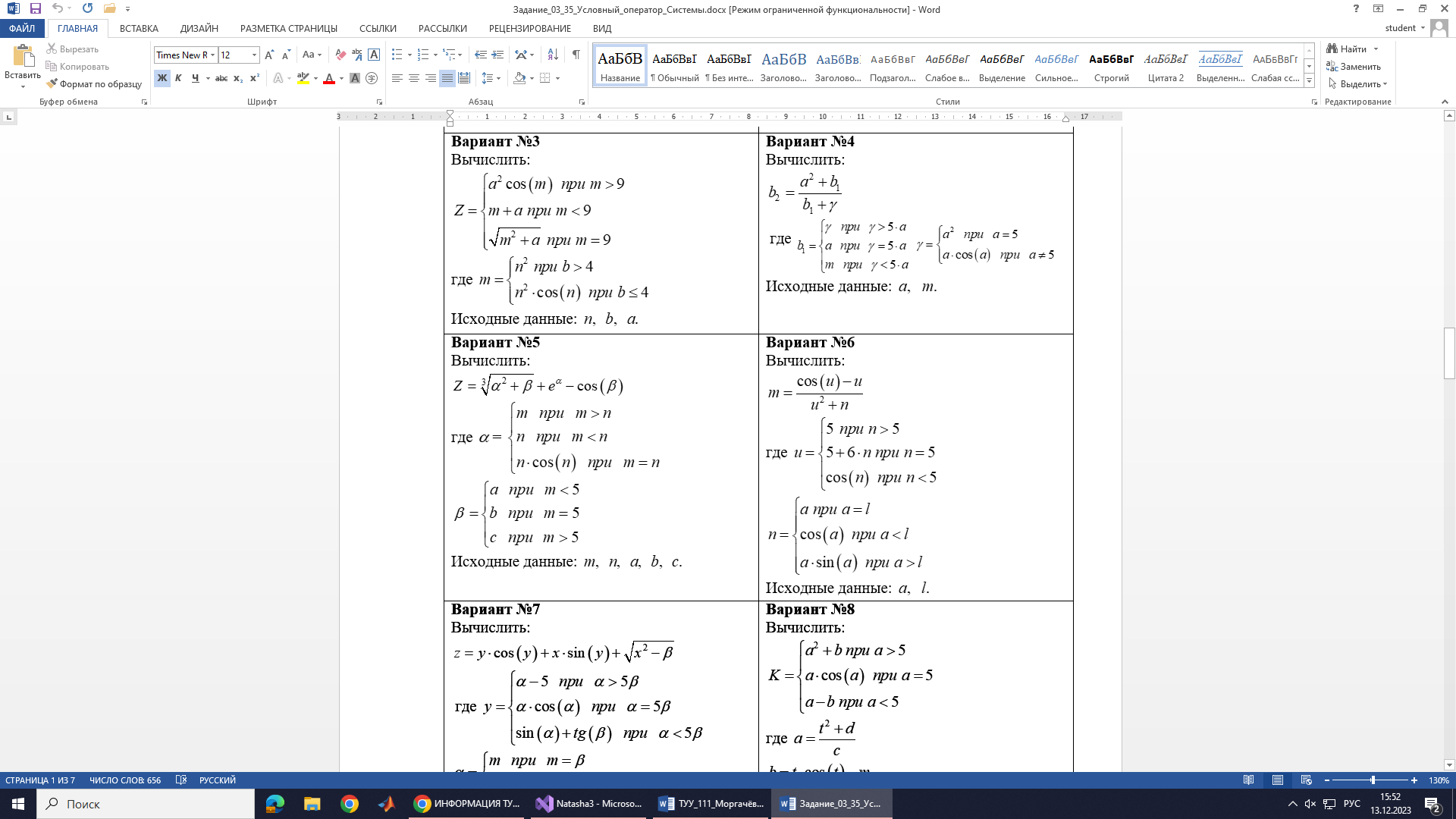
# Цель работы

Целью данной работы является освоение применения условных операторов для обработки ввода данных, а также использование метода TryParse для проверки корректности ввода.

# Формулировка задачи

Ввести контроль исходных данных. При некорректном вводе хотя бы одного из элементов, входящих во множество исходных данных, уведомлять пользователя о произошедшей ошибке и отказываться от исполнения вычислителем «ядра» составленного алгоритма.

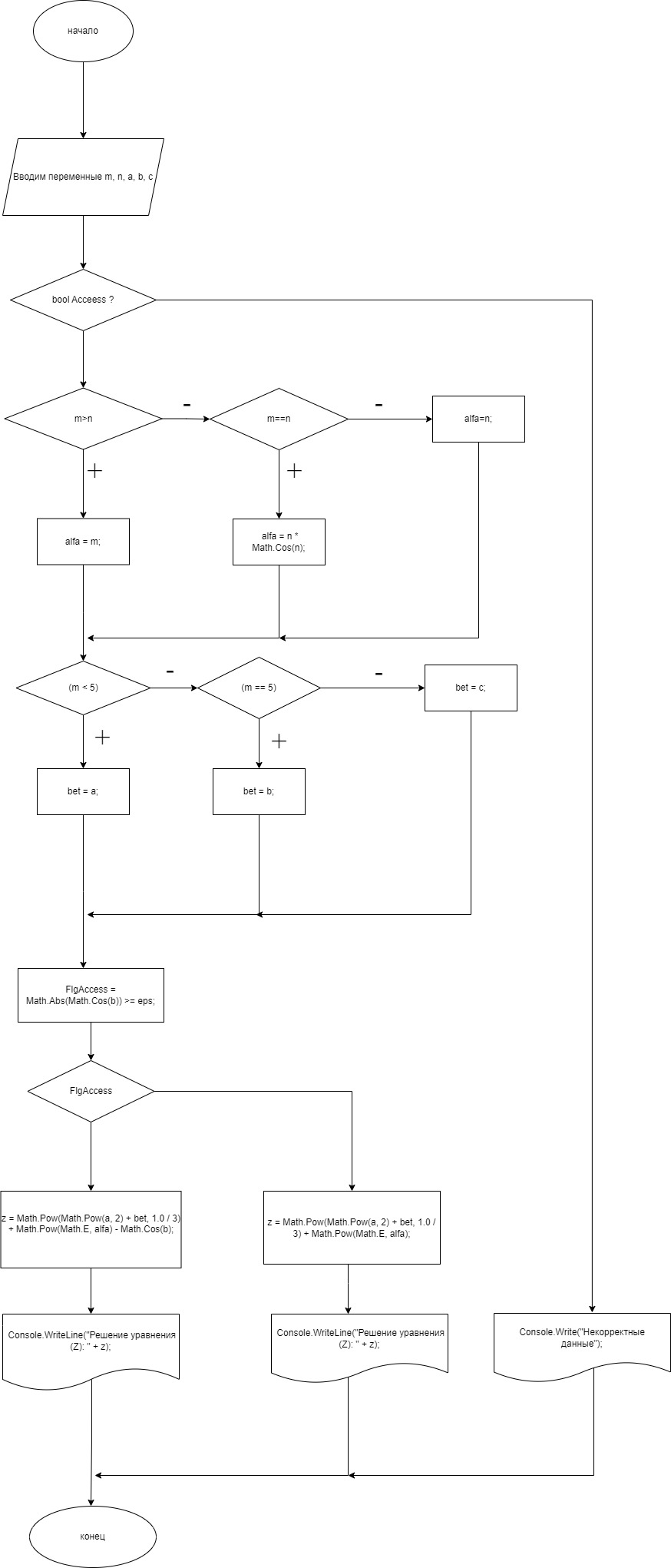
Использовать для контроля исходных данных различные состояния метода «*TryParse*», входящего в перечень доступных компонентов интересующего значащего (*valuable*) типа данных, например: *int, byte, float, double*.



2.1.1 – Вариант задания

Найти значение m, учитывая все возможные ограничения, накладываемые не только на переменные, но и на участвующие в расчёте функциональные зависимости.

# Блок-схема



3.1.1 – Блок-схема алгоритма

# Подбор тестовых примеров

* 1. **Первый ввод**

m = 2;

n = 1;

a = 34;

b = 23;

c = 7;

m > n, следовательно = m = 2

m < 5, следовательно

 =

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

4.1.1 – Первый тестовый пример

* 1. **Второй ввод**

m = 6;

n = 6;

a = 2;

b = 3;

c = 4;

m = n, следовательно = n\*cos(n)

m > 5, следовательно = 4

 =

* 1. **Третий ввод**

m = 3;

n = 4;

a = 2;

b = 3;

c = 4;

m < 5, следовательно

m < n, следовательно  = n = 4

 =

* 1. **Четвертый ввод**

m = 5;

n = 2;

a = 2;

b = 3;

c = 4;

m = 5, следовательно

m > n, следовательно = m = 5

 =

* 1. **Пятый ввод**

m = 7;

n = 5;

a = 2;

b = 0;

c = 4;

Math.Abs(Math.Cos(b)) >= eps

m = n, следовательно = n\*cos(n)

m = 5, следовательно

 =

* 1. **Шестой ввод**

m = 7;

n = 5;

a = 2;

b = 1,5;

c = 4;

Math.Abs(Math.Cos(b)) < eps

m = n, следовательно = n\*cos(n)

m = 5, следовательно

 =

* 1. **Седьмой ввод**

m = 2;

n = 1;

a = 34;

b = 23

c = 7;

m == n следовательно = m = 2

m < 5, следовательно

* 1. **Восьмой ввод**

m = 2;

n = 1;

a = 34;

b = 23

c = 7;

Введены некорректные данные

* 1. **Девятый ввод**

m = q;

n = 4;

a = 5;

b = 6;

c = 7;

Введены некорректные данные

* 1. **Десятый ввод**

m = 7;

n = w;

a = 8;

b = 9;

c = 45;

Введены некорректные данные

* 1. **Одиннадцатый ввод**

m = 1;

n = 0;

a = e;

b = 2;

c = 3;

Введены некорректные данные

* 1. **Двенадцатый ввод**

m = 4;

n = 5;

a = 6;

b = r;

c = 9;

Введены некорректные данные

* 1. **Тринадцатый ввод**

m = 6;

n = 4;

a = 5;

b = 9;

c = t;

Введены некорректные данные

# Листинг (код программы)

using System;

class Program

{

// Главная точка входа в программу

static void Main(string[] args)

{

// Объявление переменных для входных данных и расчётов

double a, b, c, m, n, alfa, bet, z, eps = 0.001;

bool FlgAccess = true; // Флаг для проверки корректности ввода

// Запрос и ввод значения m с проверкой на корректность

Console.Write("Введите число m: ");

FlgAccess &= double.TryParse(Console.ReadLine(), out m); // Если ввод некорректен, флаг станет false

Console.WriteLine();

// Запрос и ввод значения n с проверкой на корректность

Console.Write("Введите число n: ");

FlgAccess &= double.TryParse(Console.ReadLine(), out n); // Если ввод некорректен, флаг станет false

Console.WriteLine();

// Запрос и ввод значения a с проверкой на корректность

Console.Write("Введите число a: ");

FlgAccess &= double.TryParse(Console.ReadLine(), out a); // Если ввод некорректен, флаг станет false

Console.WriteLine();

// Запрос и ввод значения b с проверкой на корректность

Console.Write("Введите число b: ");

FlgAccess &= double.TryParse(Console.ReadLine(), out b); // Если ввод некорректен, флаг станет false

Console.WriteLine();

// Запрос и ввод значения c с проверкой на корректность

Console.Write("Введите число c: ");

FlgAccess &= double.TryParse(Console.ReadLine(), out c); // Если ввод некорректен, флаг станет false

Console.WriteLine();

// Проверка на корректность ввода всех данных

if (FlgAccess)

{

// Определение значения alfa в зависимости от условий

if (m > n)

{

alfa = m; // Если m больше n, то alfa = m

}

else if (m == n)

{

alfa = n \* Math.Cos(n); // Если m равно n, то alfa = n \* cos(n)

}

else

{

alfa = n; // Если m меньше n, то alfa = n

}

// Определение значения bet в зависимости от условий

if (m < 5)

{

bet = a; // Если m меньше 5, то bet = a

}

else if (m == 5)

{

bet = b; // Если m равно 5, то bet = b

}

else

{

bet = c; // Если m больше 5, то bet = c

}

// Проверка условия на значение косинуса и расчёт z

if (Math.Abs(Math.Cos(b)) >= eps)

{

// Если |cos(b)| >= eps, то выполняется полное вычисление

z = Math.Pow(alfa \* alfa + bet, 1.0 / 3.0) + Math.Pow(Math.E, alfa) - Math.Cos(bet);

}

else

{

// Если условие не выполнено, то выполняется упрощённый расчёт

z = Math.Pow(alfa \* alfa + bet, 1.0 / 3.0) + Math.Pow(Math.E, alfa);

}

// Вывод результата на экран

Console.WriteLine("Решение уравнения (Z): " + z);

}

else

{

// Если ввод некорректен, выводится сообщение об ошибке

Console.WriteLine("Введены некорректные данные");

}

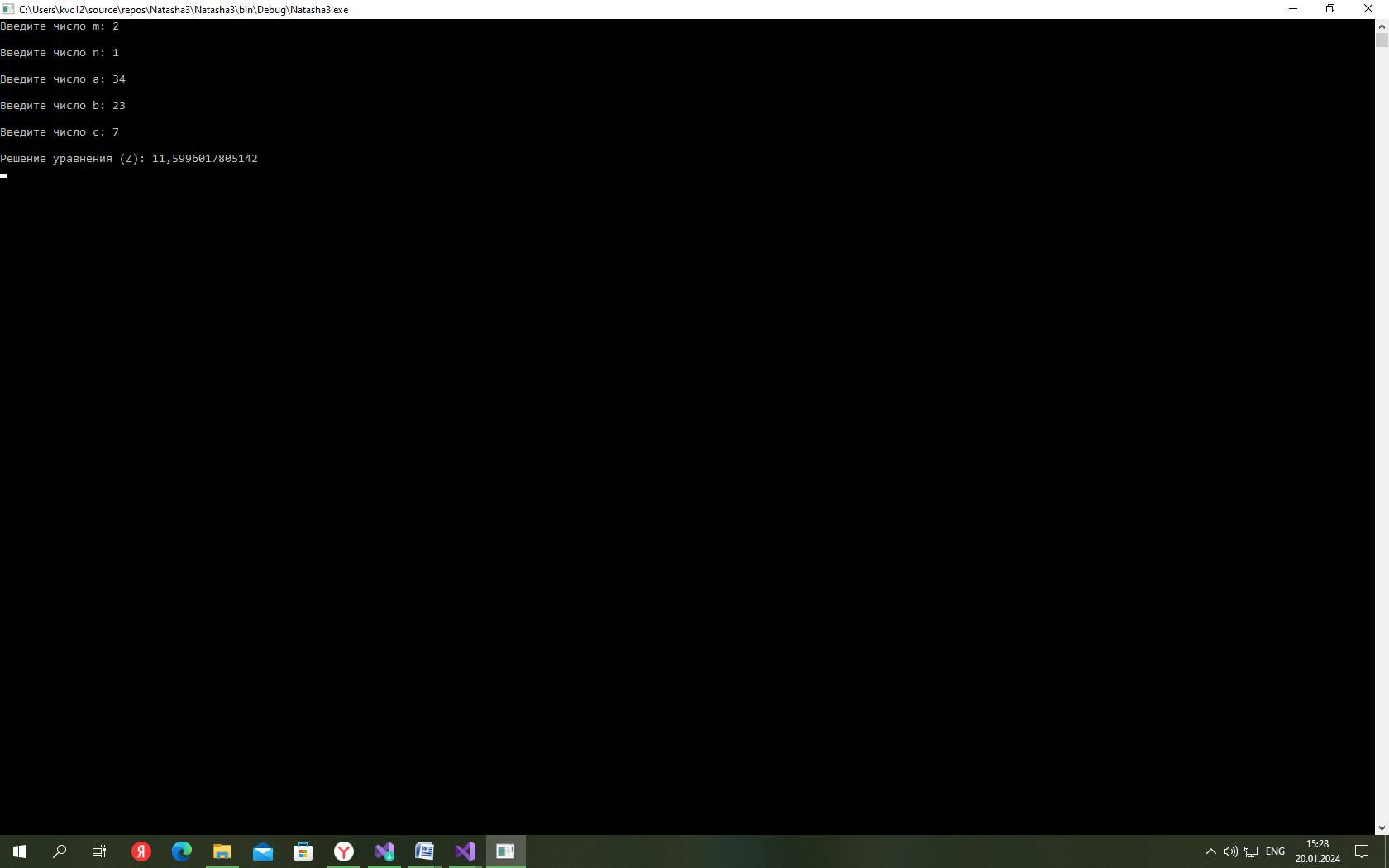
// Ожидание нажатия клавиши для завершения программы

Console.ReadKey(true);

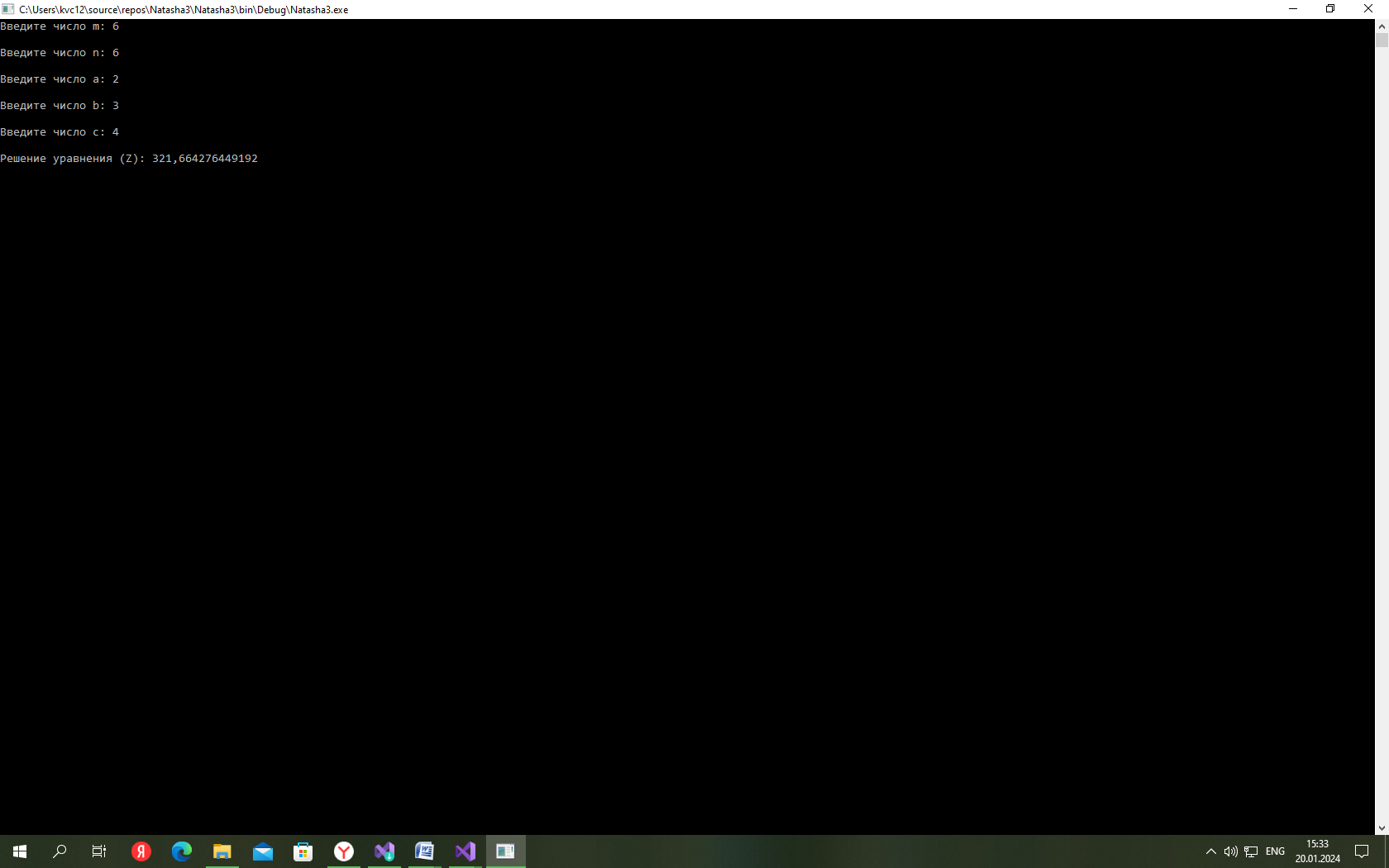
}

}

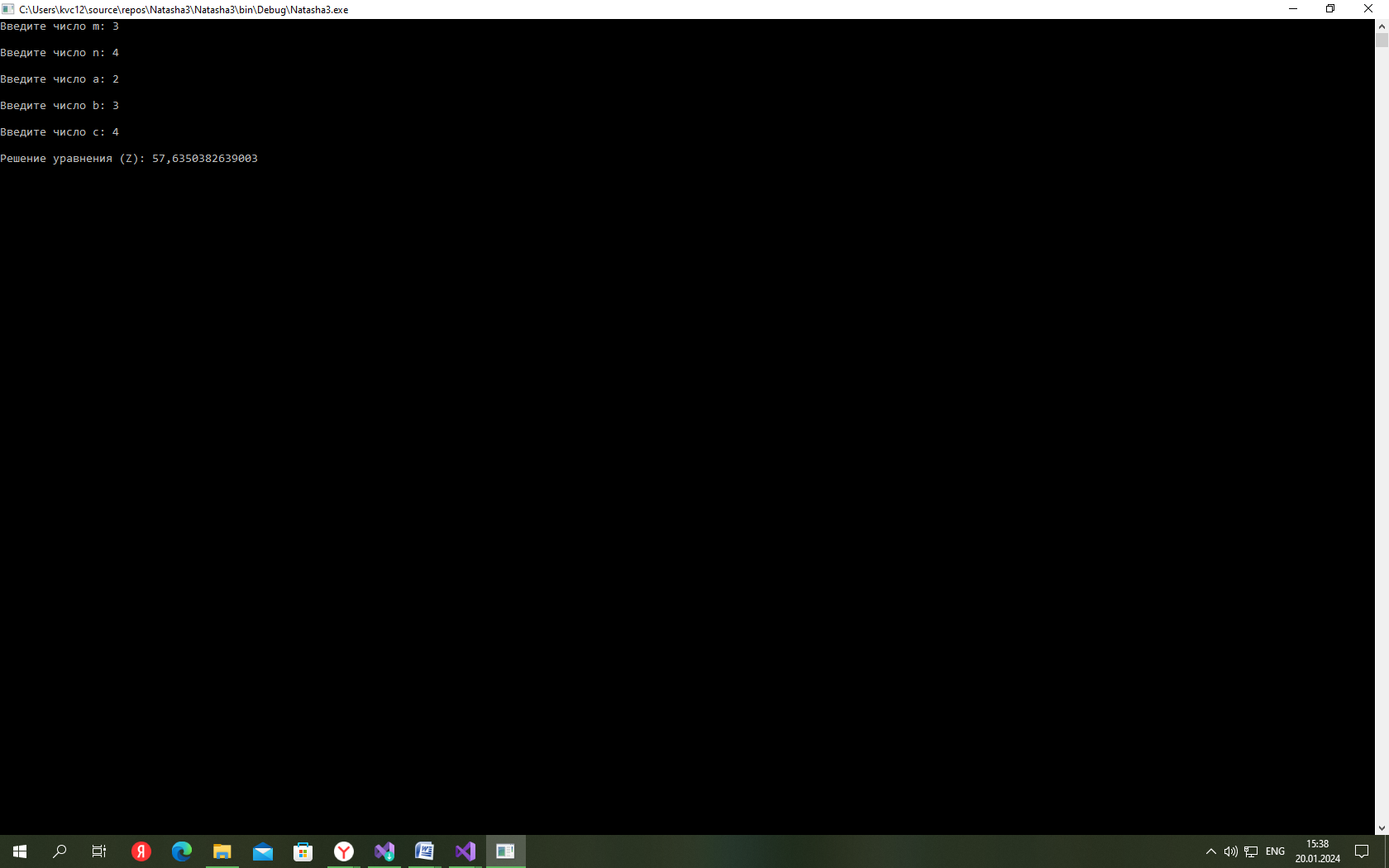
# Тестирование (расчёт тестовых примеров ПК)



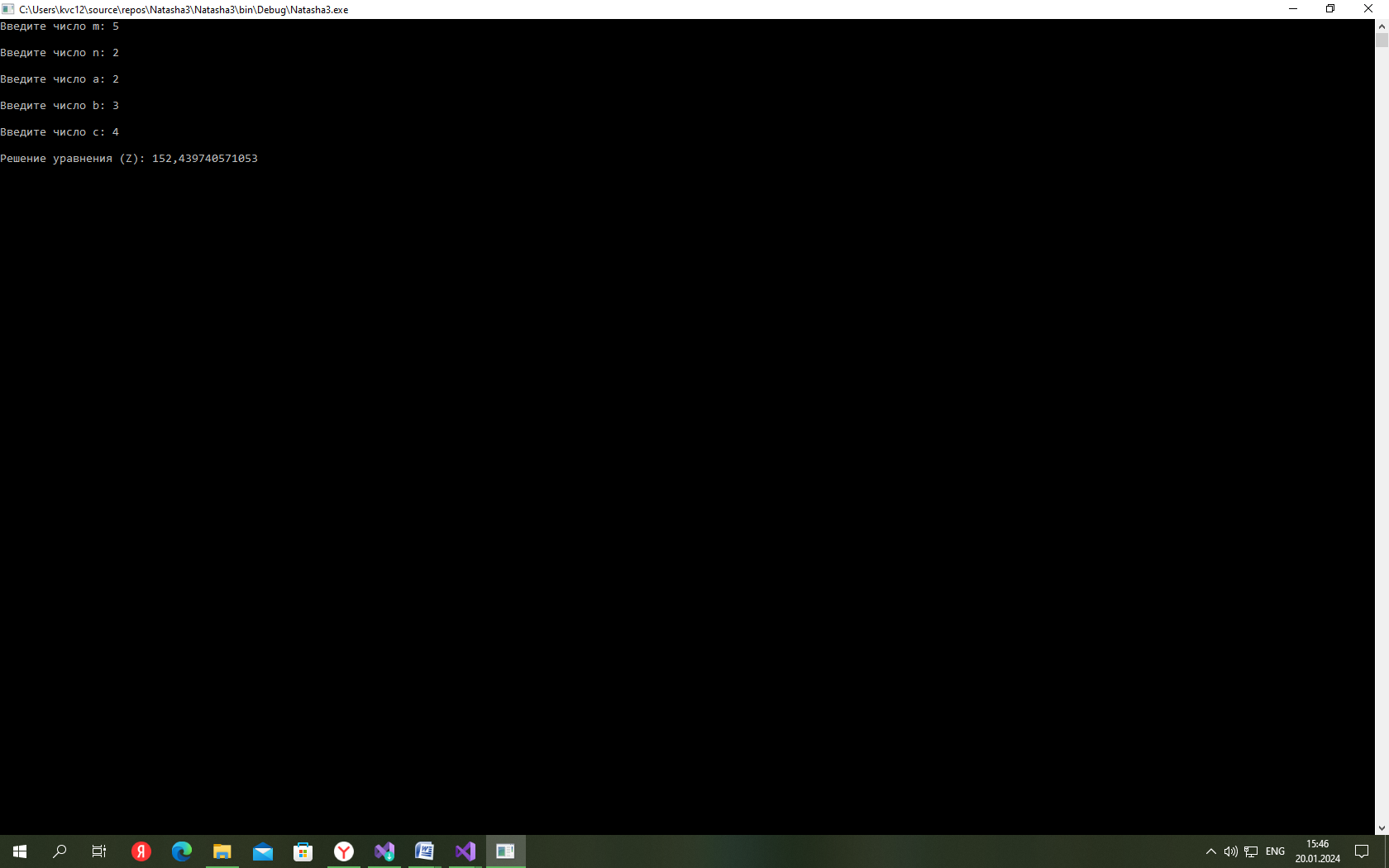
6.1.1 – Тестирование №1

****

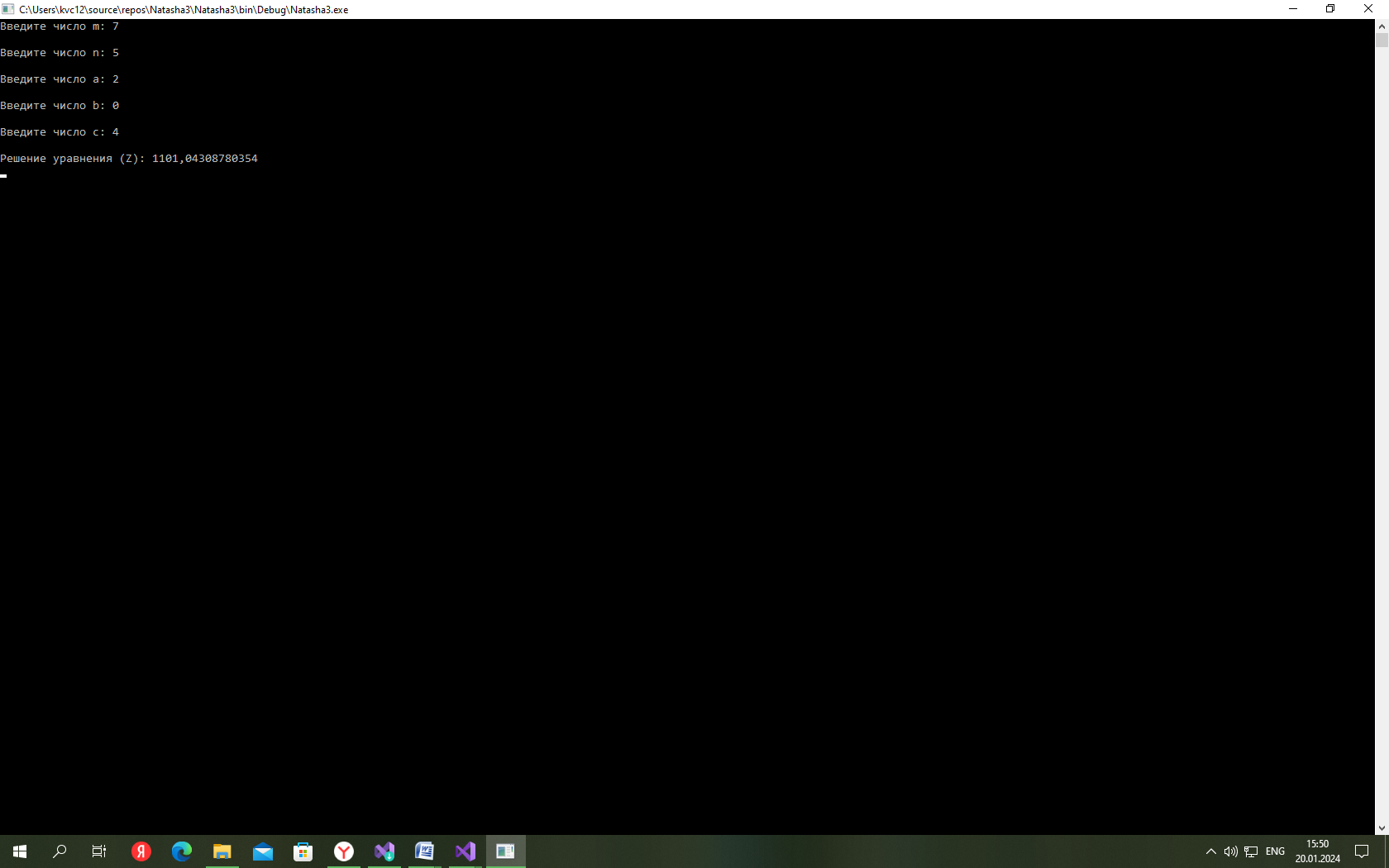
6.1.2 – Тестирование №2

****

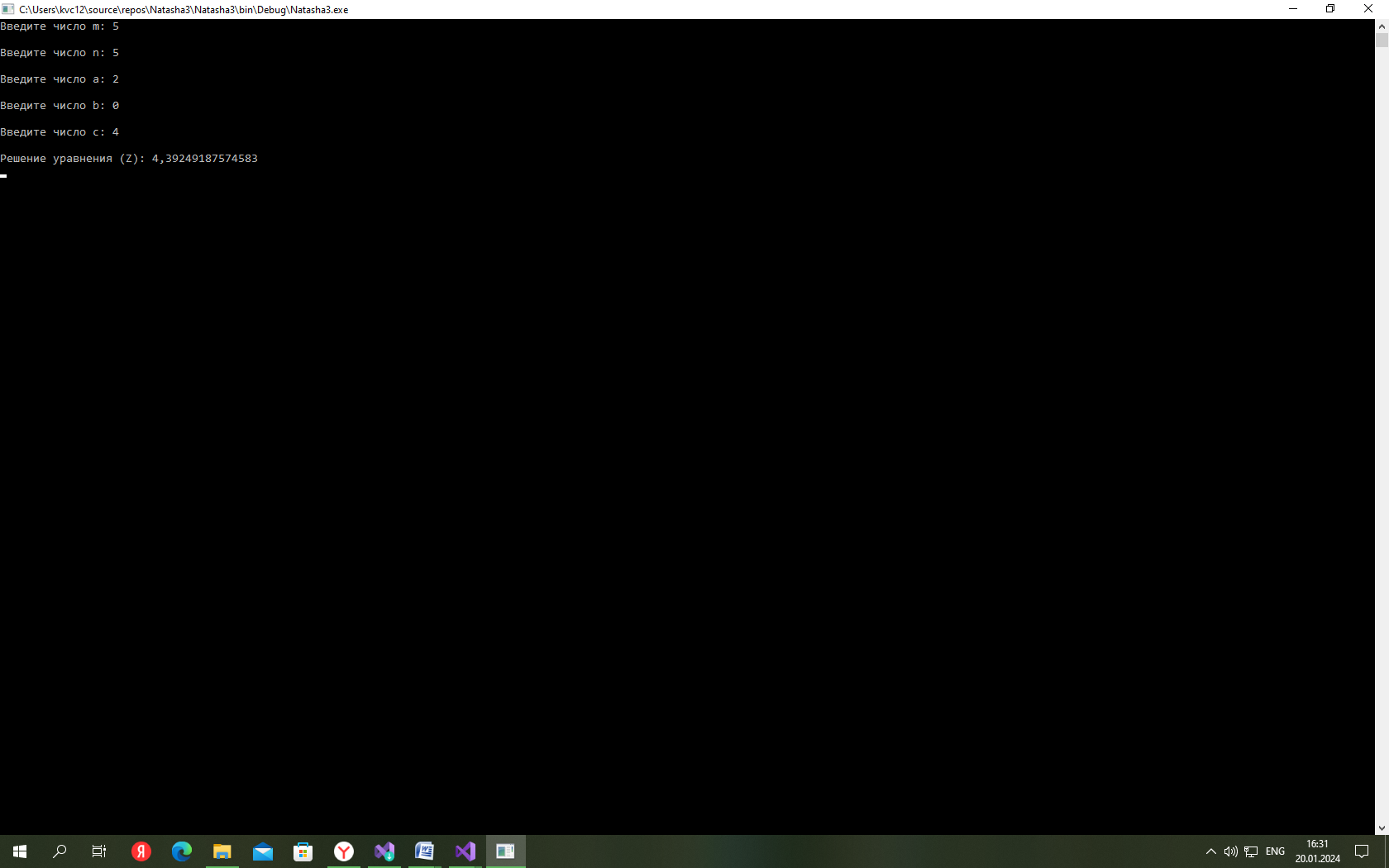
6.1.3 – Тестирование №3

****

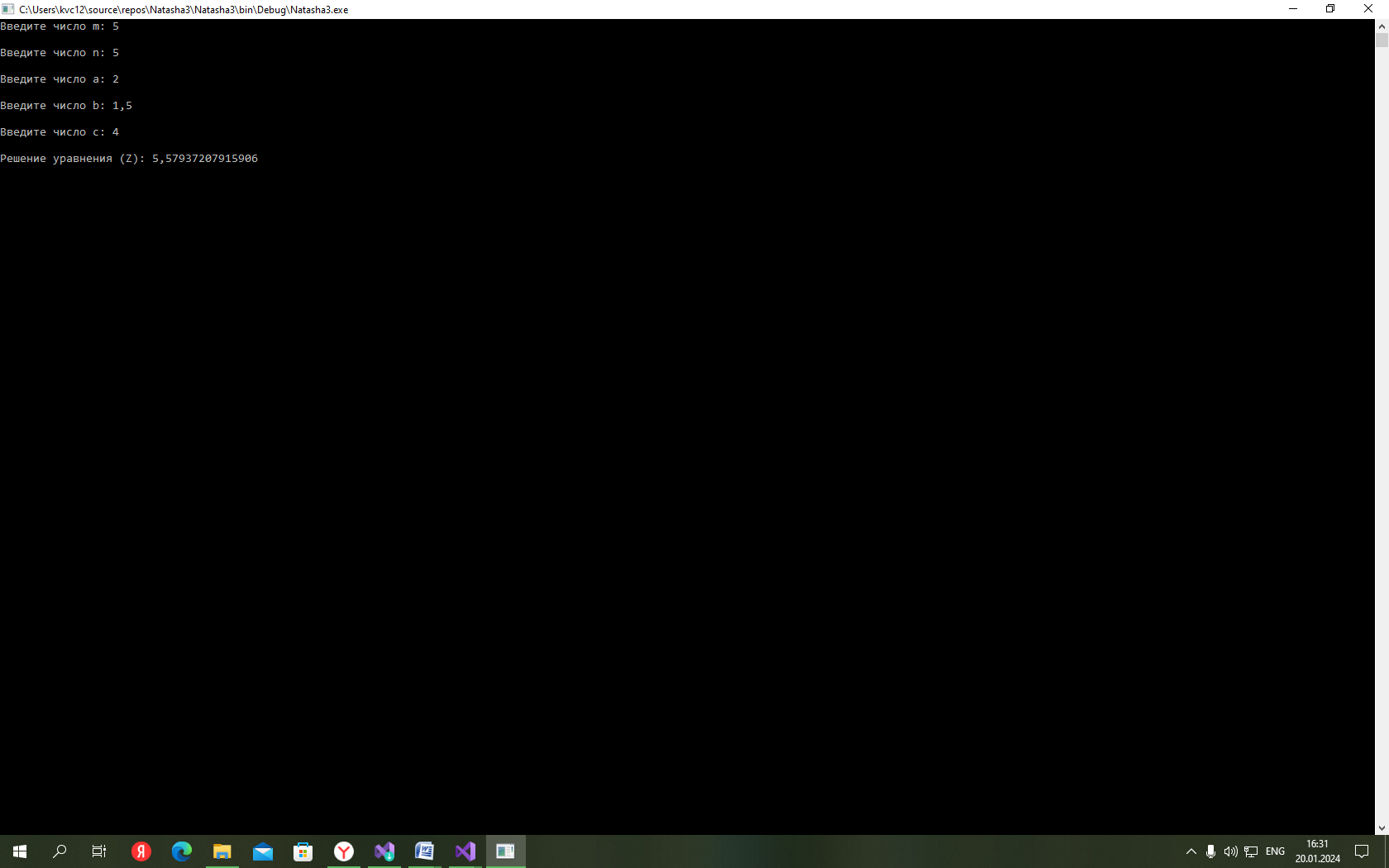
6.1.4 – Тестирование №4

****

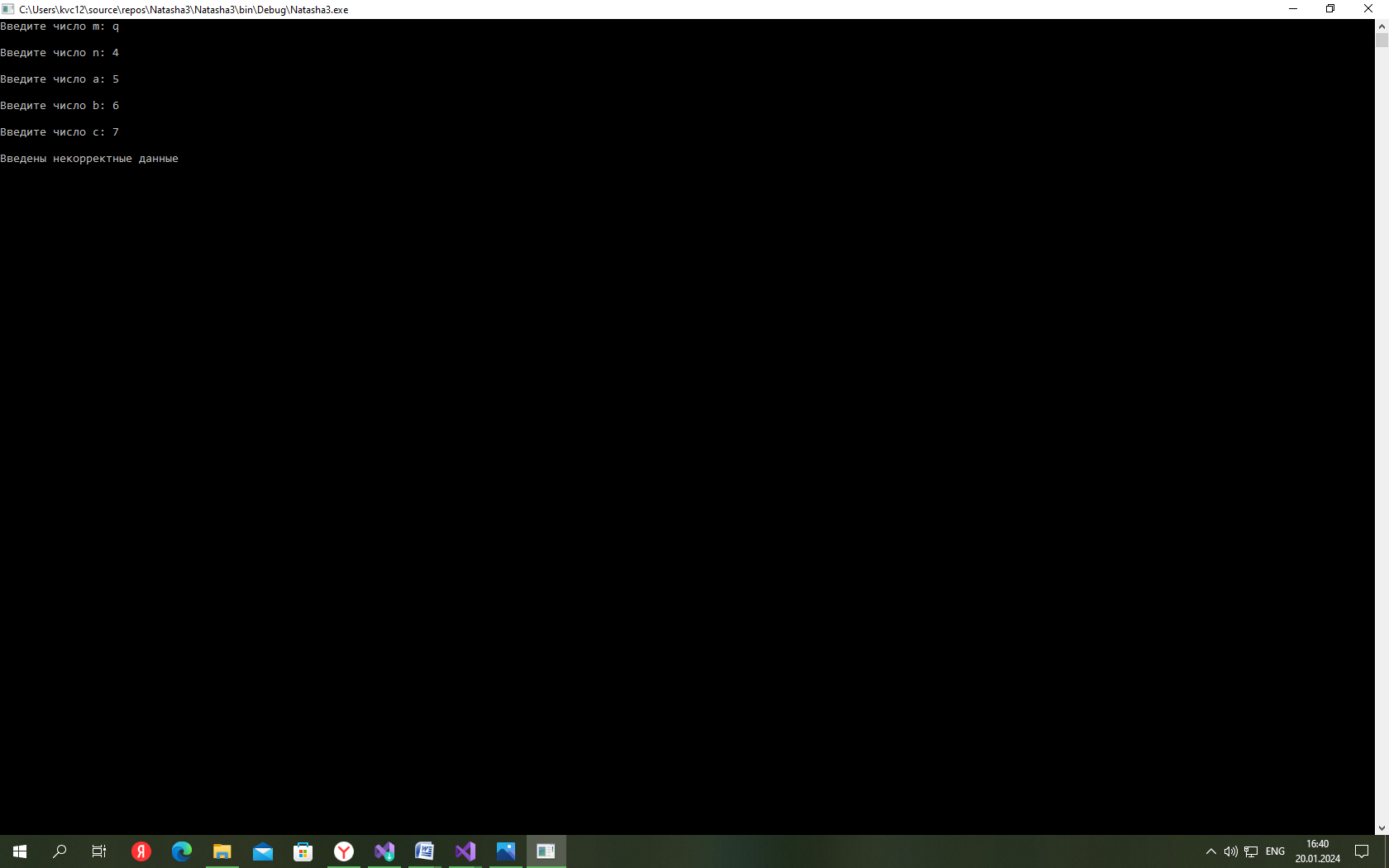
6.1.5 – Тестирование №5

****

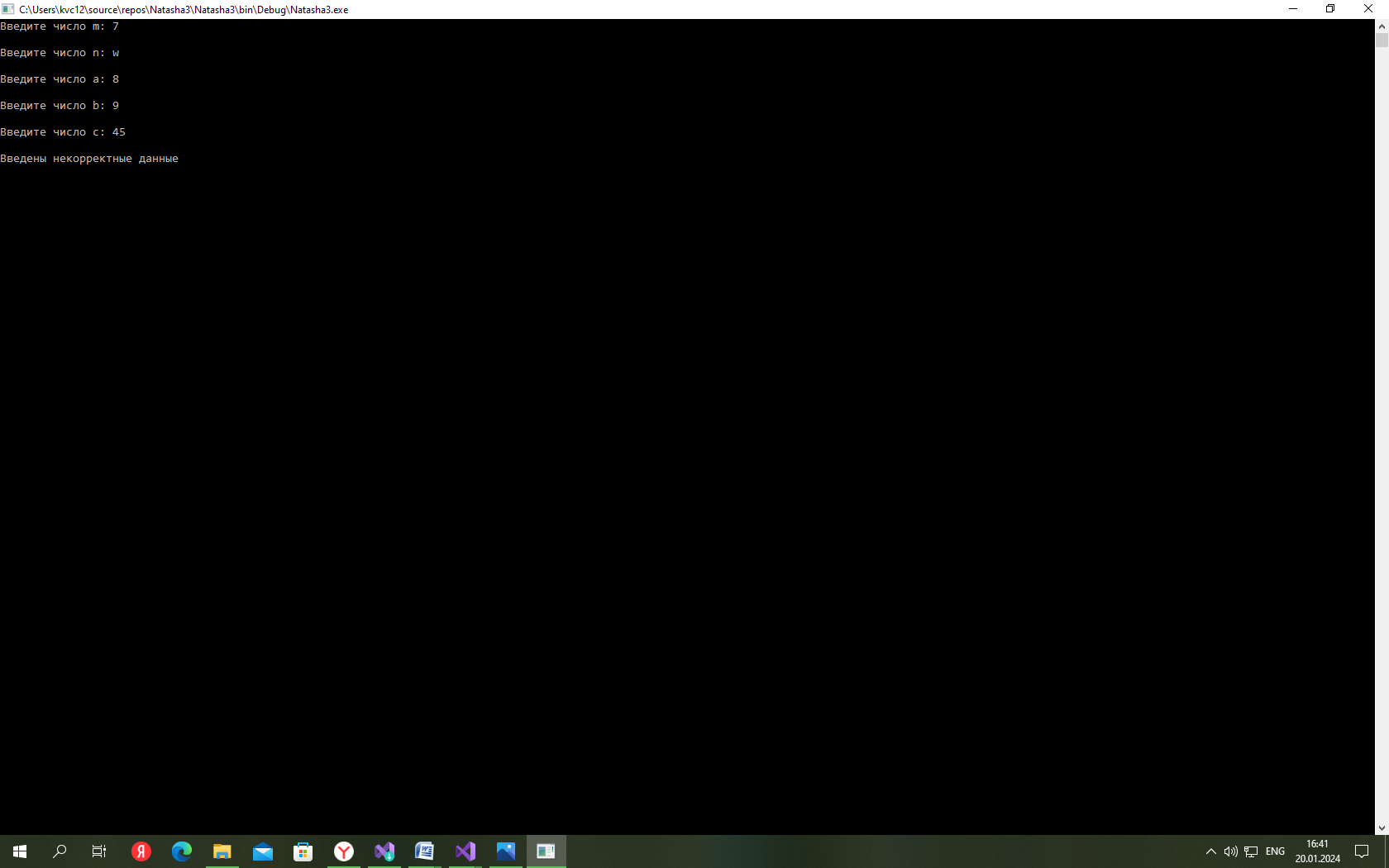
6.1.6 – Тестирование №6

****

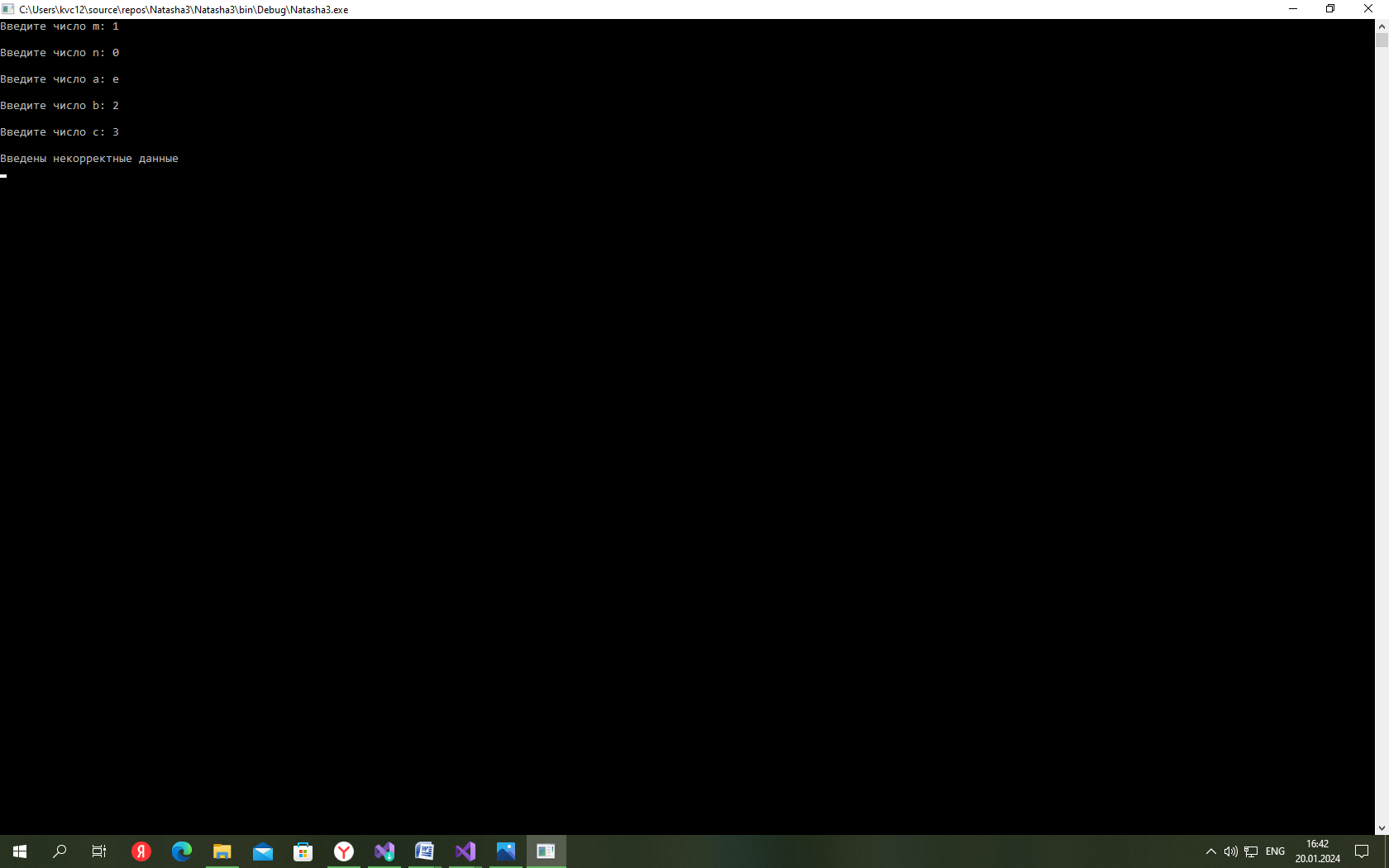
6.1.7 – Тестирование №7

****

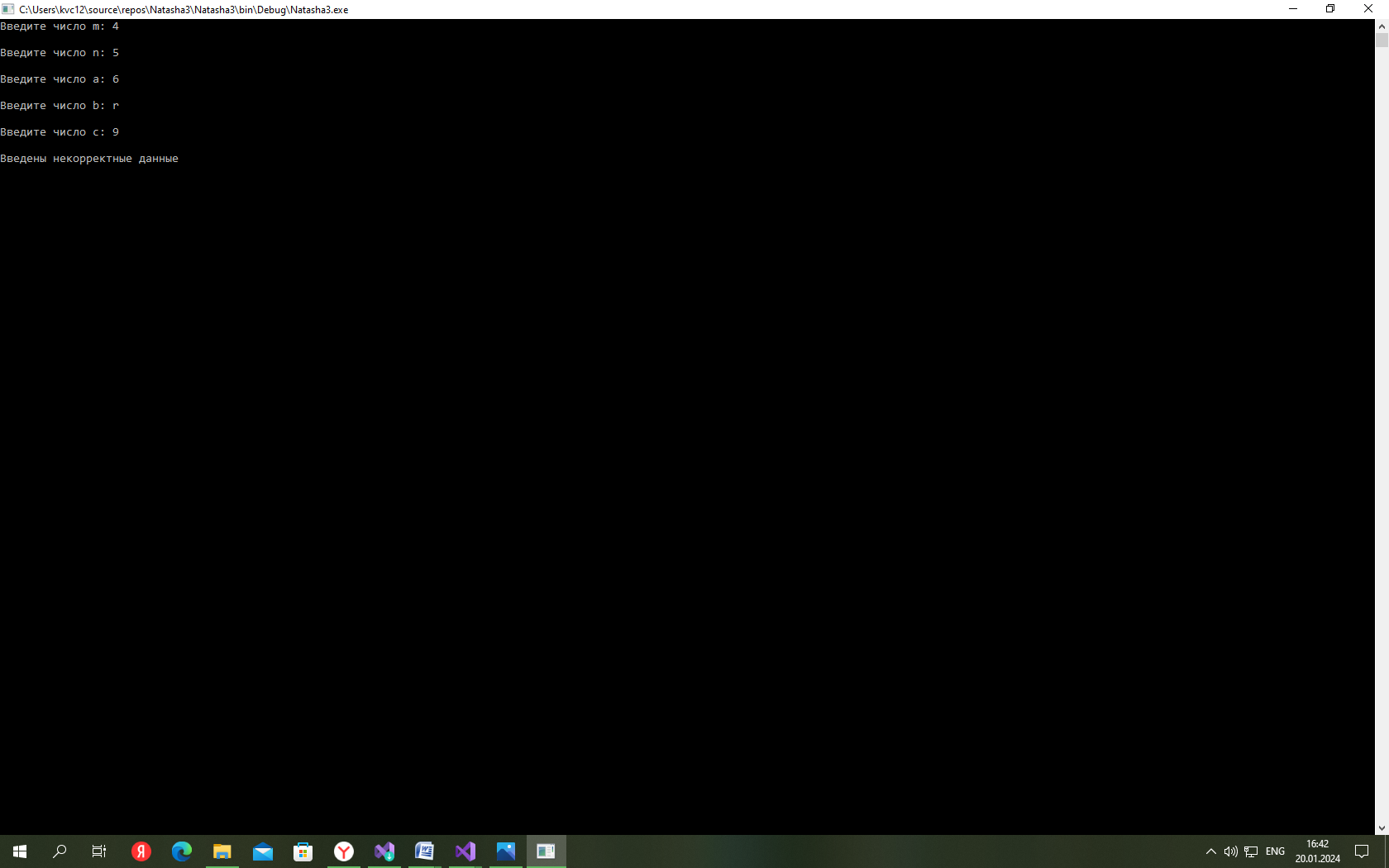
6.1.8 – Тестирование №8

****

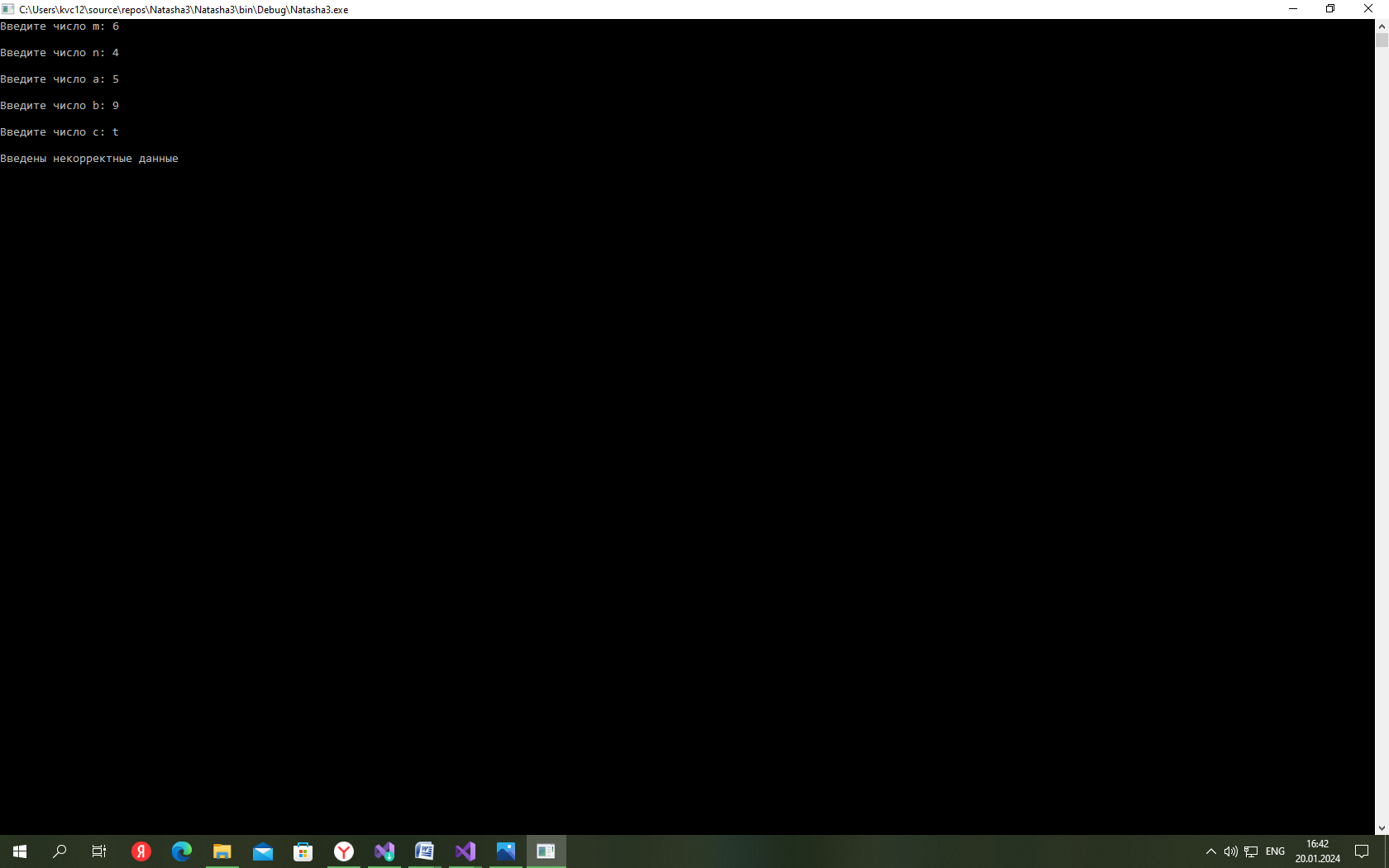
6.1.9 – Тестирование №9

****

6.1.10 – Тестирование №10

****

6.1.11 – Тестирование №11

****

6.1.12 – Тестирование №12

# Вывод

В ходе выполнения лабораторной работы была решена задача, связанная с использованием условных операторов для проверки корректности ввода данных и выполнения вычислений. Программа, с использованием метода TryParse, корректно обрабатывает вводимые пользователем значения и, в случае ошибок ввода, сообщает об этом и завершает выполнение. Работа помогла укрепить знания о принципах работы с условиями и обработке ошибок в программировании, а также продемонстрировала важность правильной проверки данных для обеспечения надёжности программ.