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.

13.01.2025

(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.

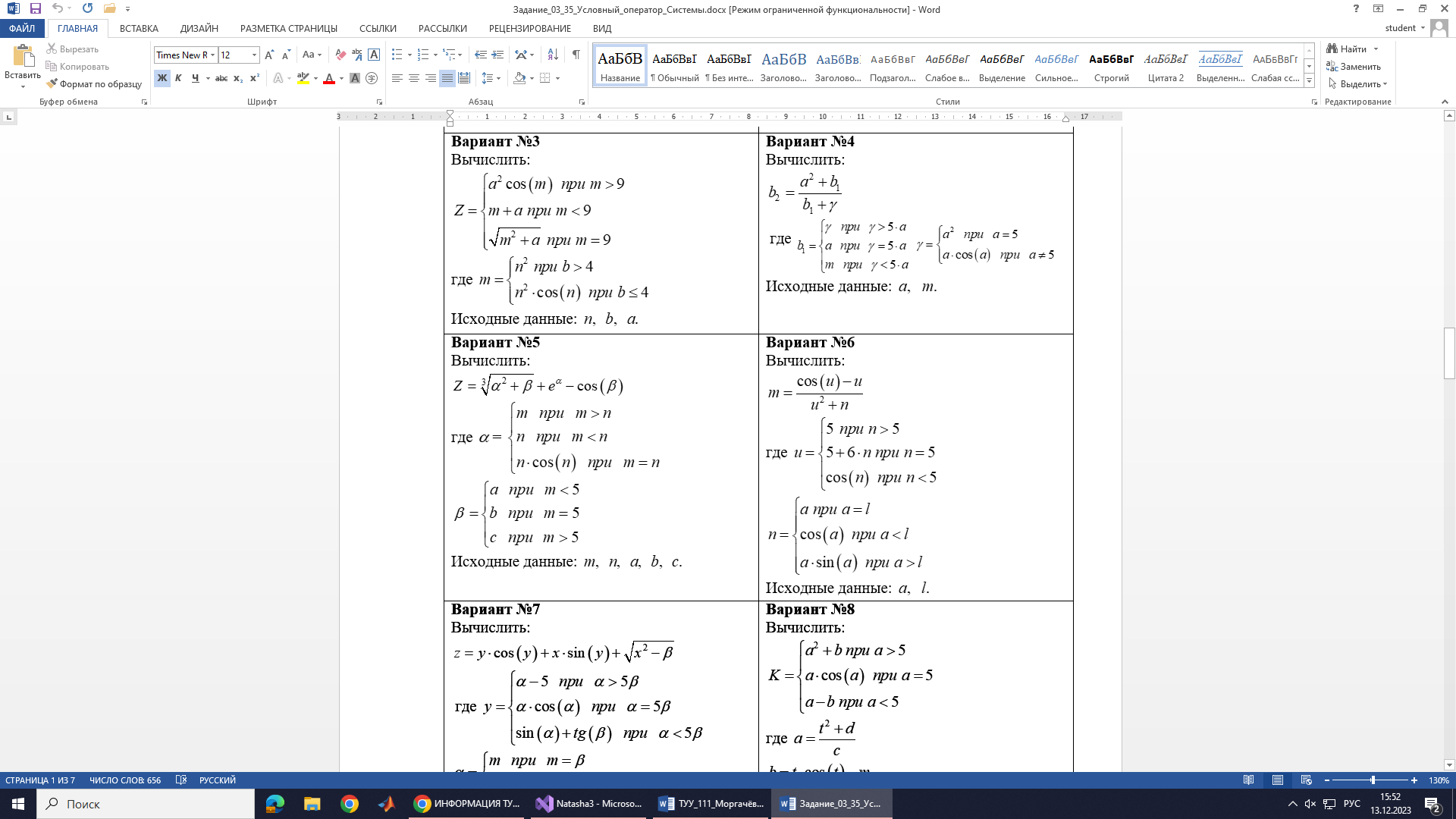
# Формулировка задачи

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*.

* 1. **Задание 1**

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



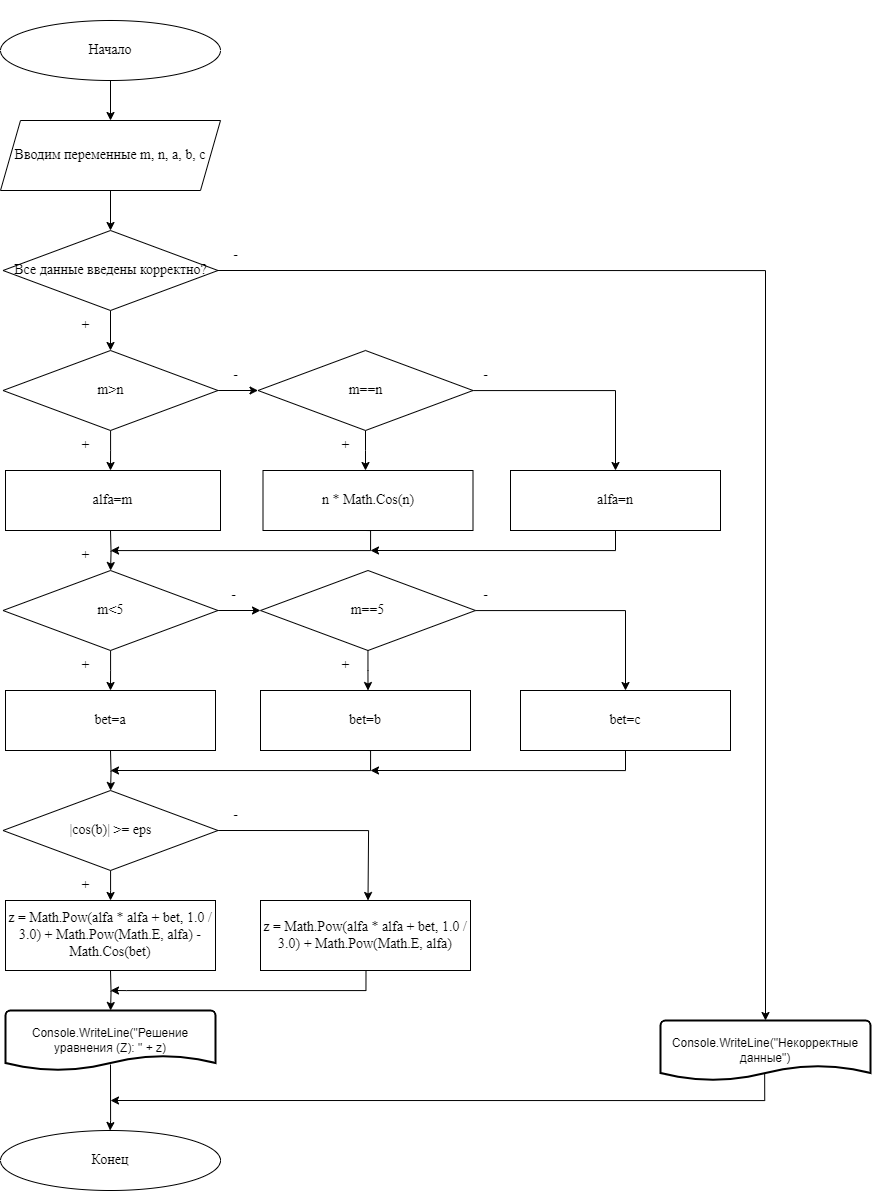
2.1.1 – Вариант задания

* 1. **Задание 2**

Необходимо разработать программу на языке C#, которая принимает на вход координаты трех точек A(x1​, y1​), B(x2​, y2​), C(x3​, y3​) на координатной плоскости и определяет, лежат ли эти точки на одной прямой. Для этого следует воспользоваться геометрическим свойством: три точки коллинеарны, если площадь треугольника, образованного этими точками, равна нулю.

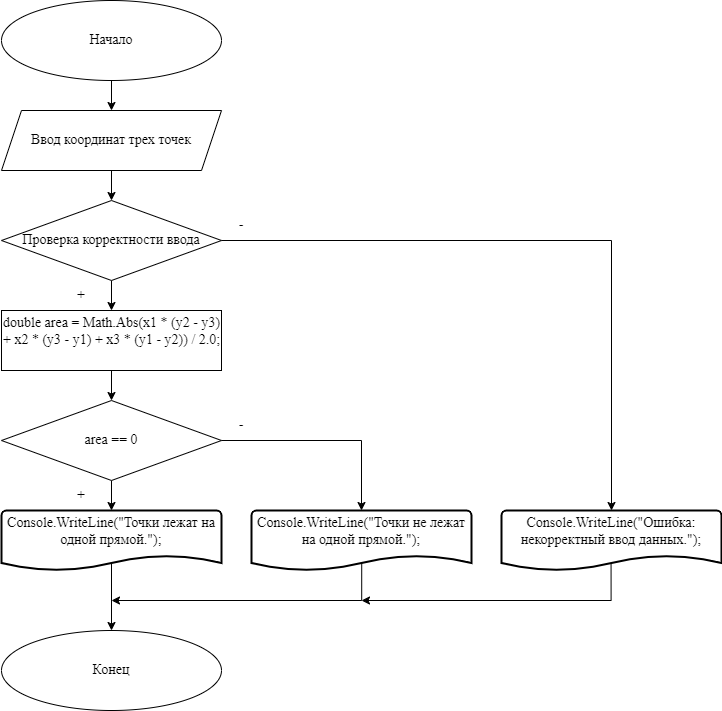
# Блок-схема

* 1. **Задание 1**



3.1.1 – Блок-схема алгоритма

* 1. **Задание 2**

****

3.2.1 – Блок-схема алгоритма

# Подбор тестовых примеров

* 1. **Задание 1**

1. 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

2) m = 6;

n = 6;

a = 2;

b = 3;

c = 4;

m = n, следовательно = n\*cos(n)

m > 5, следовательно = 4

 =

3) m = 3;

n = 4;

a = 2;

b = 3;

c = 4;

m < 5, следовательно

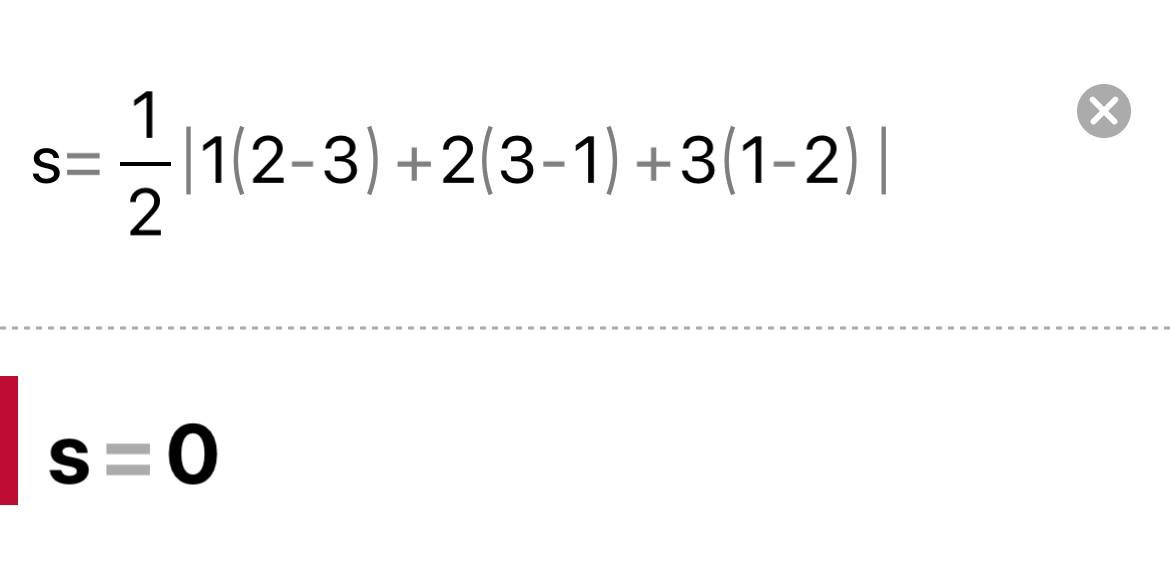
m < n, следовательно  = n = 4

 =

* 1. **Задание 2**

1. Входные данные: А(1, 1), В(2, 2), С(3, 3)

Формула площади:



4.2.1 – Первый тестовый пример

Описание: Площадь треугольника равна 0, следовательно, точки лежат на одной прямой.

Ожидаемый результат: "Точки лежат на одной прямой."

1. Входные данные: А(1, t), В(2, 2), С(3, 3)

Описание: В данном случае, если t не задано как число, ввод данных для точки A считается некорректным.

Ожидаемый результат: Сообщение об ошибке: "Ошибка: некорректный ввод координат точки A."

# Листинг (код программы)

* 1. **Задание 1**

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);

}

}

* 1. **Задание 2**

using System;

class Program

{

static void Main()

{

// Блок ввода данных

Console.WriteLine("Введите координаты точки A (x1, y1):");

if (!double.TryParse(Console.ReadLine(), out double x1) || !double.TryParse(Console.ReadLine(), out double y1))

{

Console.WriteLine("Ошибка: некорректный ввод координат точки A.");

return;

}

Console.WriteLine("Введите координаты точки B (x2, y2):");

if (!double.TryParse(Console.ReadLine(), out double x2) || !double.TryParse(Console.ReadLine(), out double y2))

{

Console.WriteLine("Ошибка: некорректный ввод координат точки B.");

return;

}

Console.WriteLine("Введите координаты точки C (x3, y3):");

if (!double.TryParse(Console.ReadLine(), out double x3) || !double.TryParse(Console.ReadLine(), out double y3))

{

Console.WriteLine("Ошибка: некорректный ввод координат точки C.");

return;

}

// Блок расчетов

double area = Math.Abs(x1 \* (y2 - y3) + x2 \* (y3 - y1) + x3 \* (y1 - y2)) / 2.0;

// Блок вывода результата

if (area == 0)

{

Console.WriteLine("Точки лежат на одной прямой.");

}

else

{

Console.WriteLine("Точки не лежат на одной прямой.");

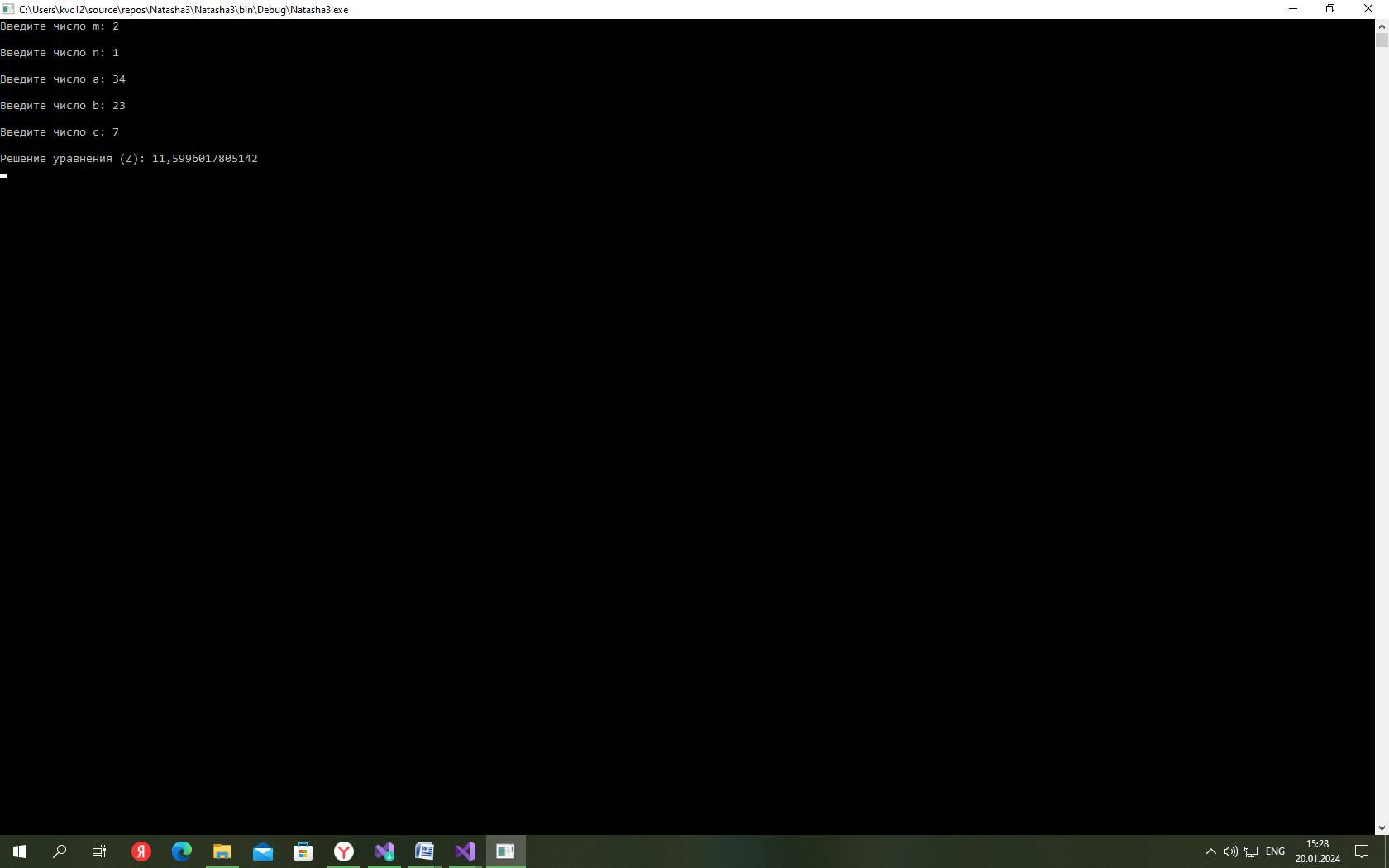
}

}

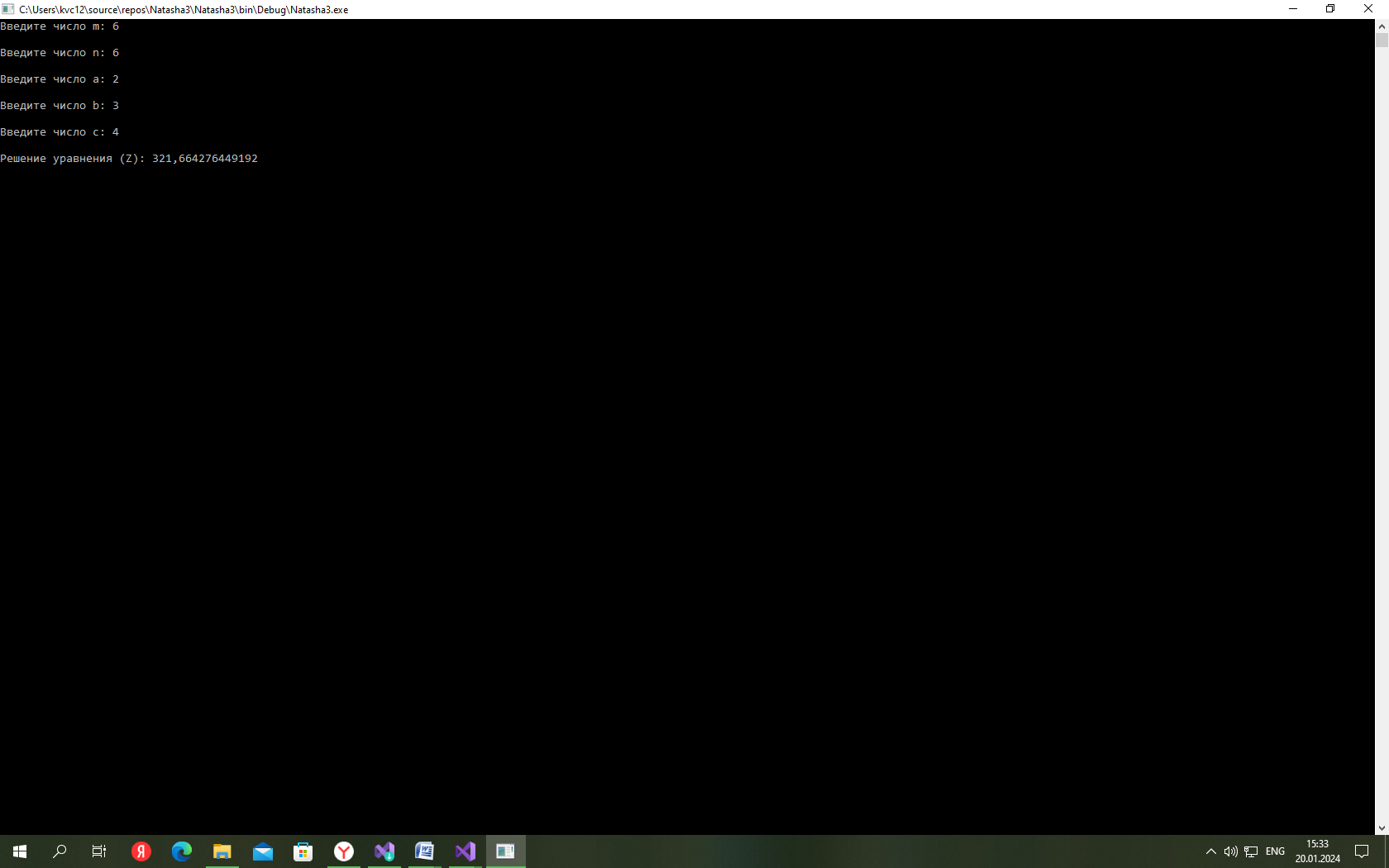
}

# Тестирование (расчёт тестовых примеров ПК)

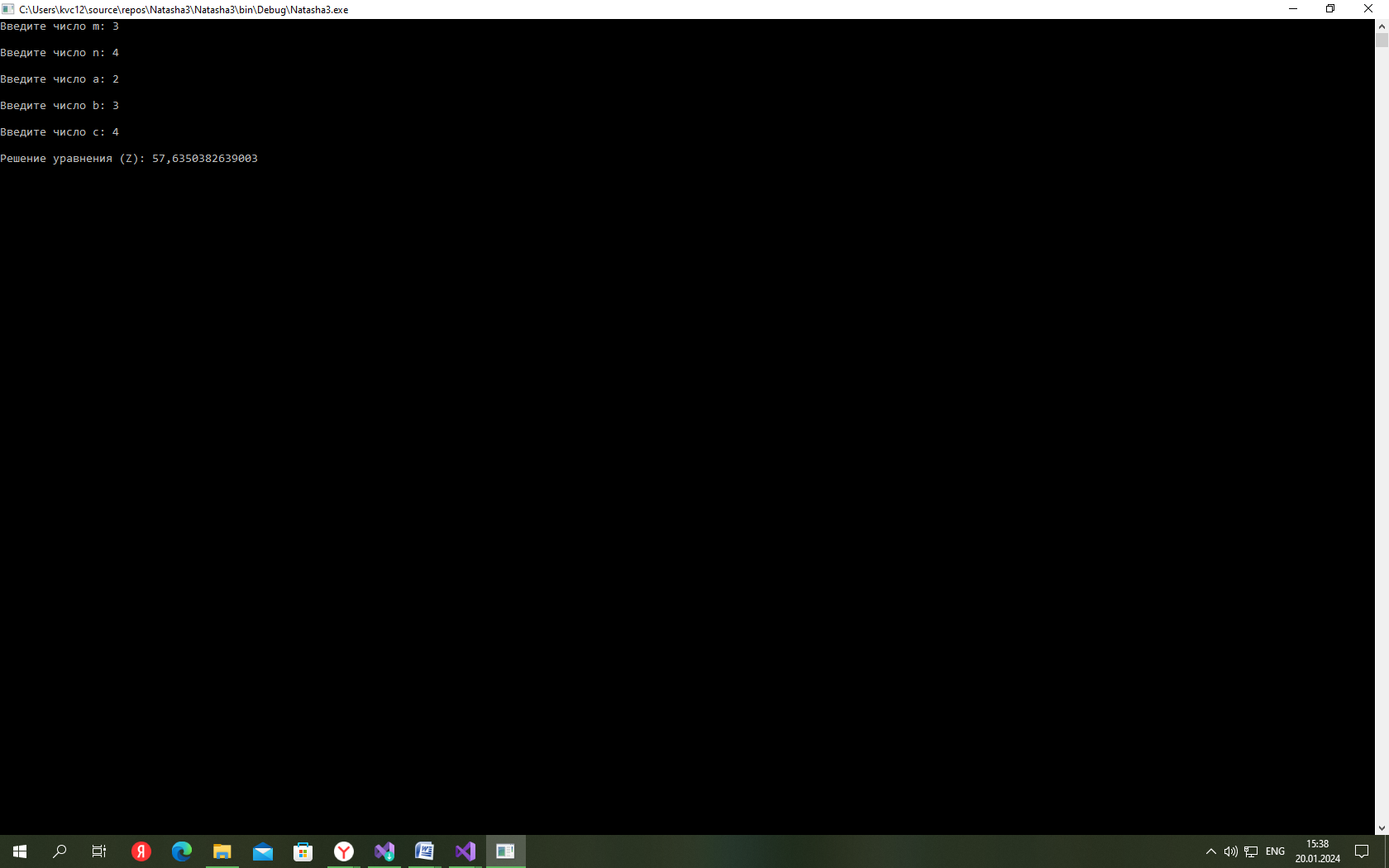
# Задание 1



6.1.1 – Тестирование №1

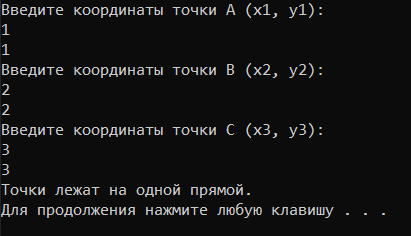
****

6.1.2 – Тестирование №2

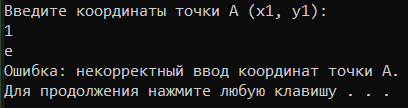
****

6.1.3 – Тестирование №3

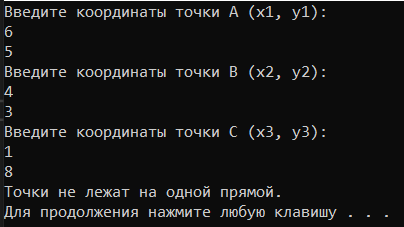
* 1. **Задание 2**



6.2.1 – Тестирование №1



6.2.2 – Тестирование №2



6.2.3 – Тестирование №3

# 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.