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 № 7

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Циклы с пред- и постусловием»

Выполнил: ст. гр. ТУУ-111

Ванчаев Д.С.

Вариант №10

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

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

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

(дата приёмки)

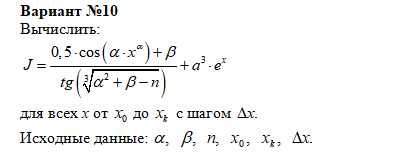
Москва – 2025 г.

1. **Цель работы**

Решить поставленную задачу программирования по разделу «Циклы с пред- и постусловием». Не использовать операторы циклов по известному диапазону (*for*) и известному множеству (*foreach*) значений.

1. **Формулировка задачи**

Вычислить:



1. **Блок - схема алгоритма**

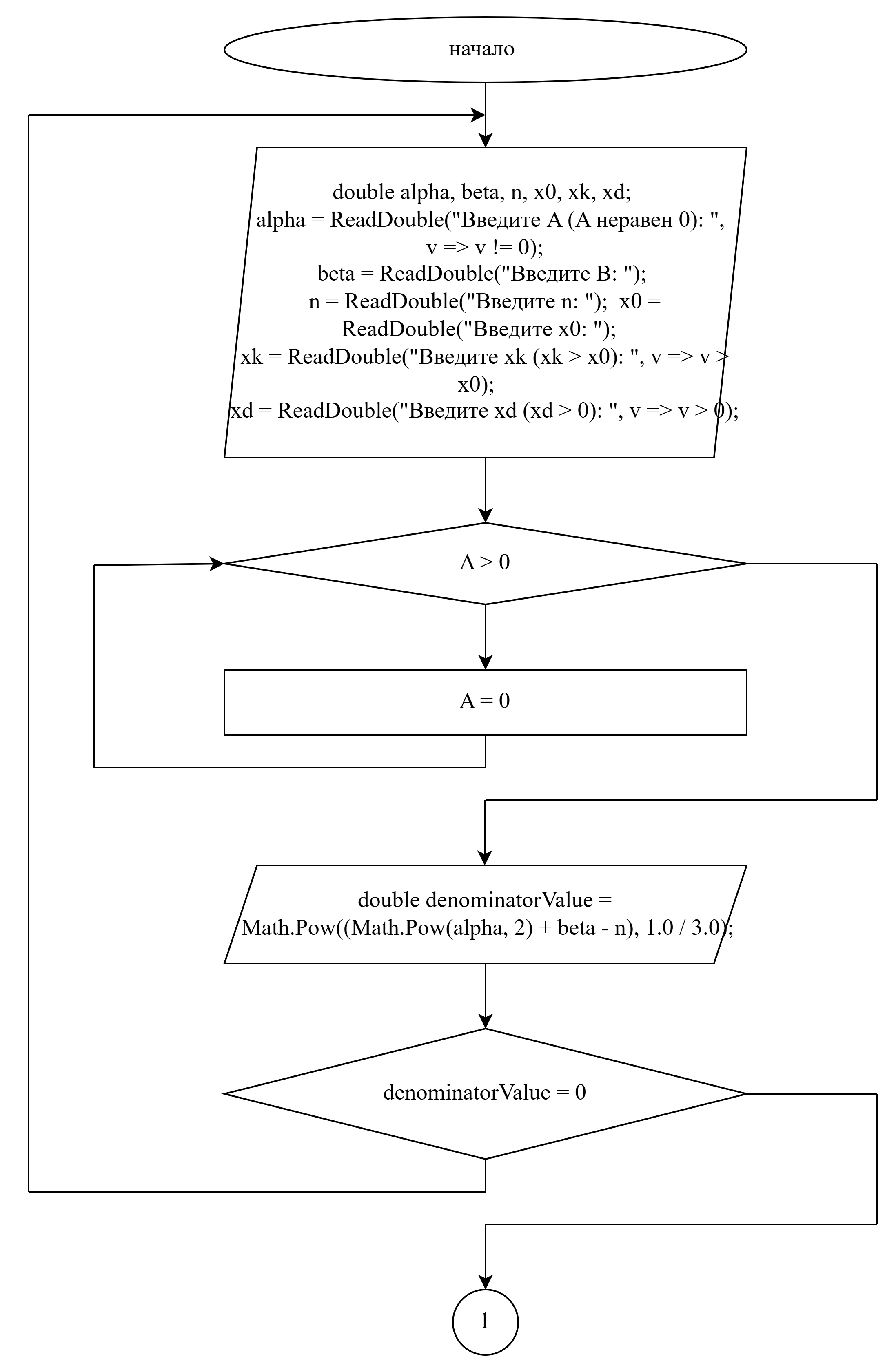


Рисунок 3.1 - Блок схема алгоритма

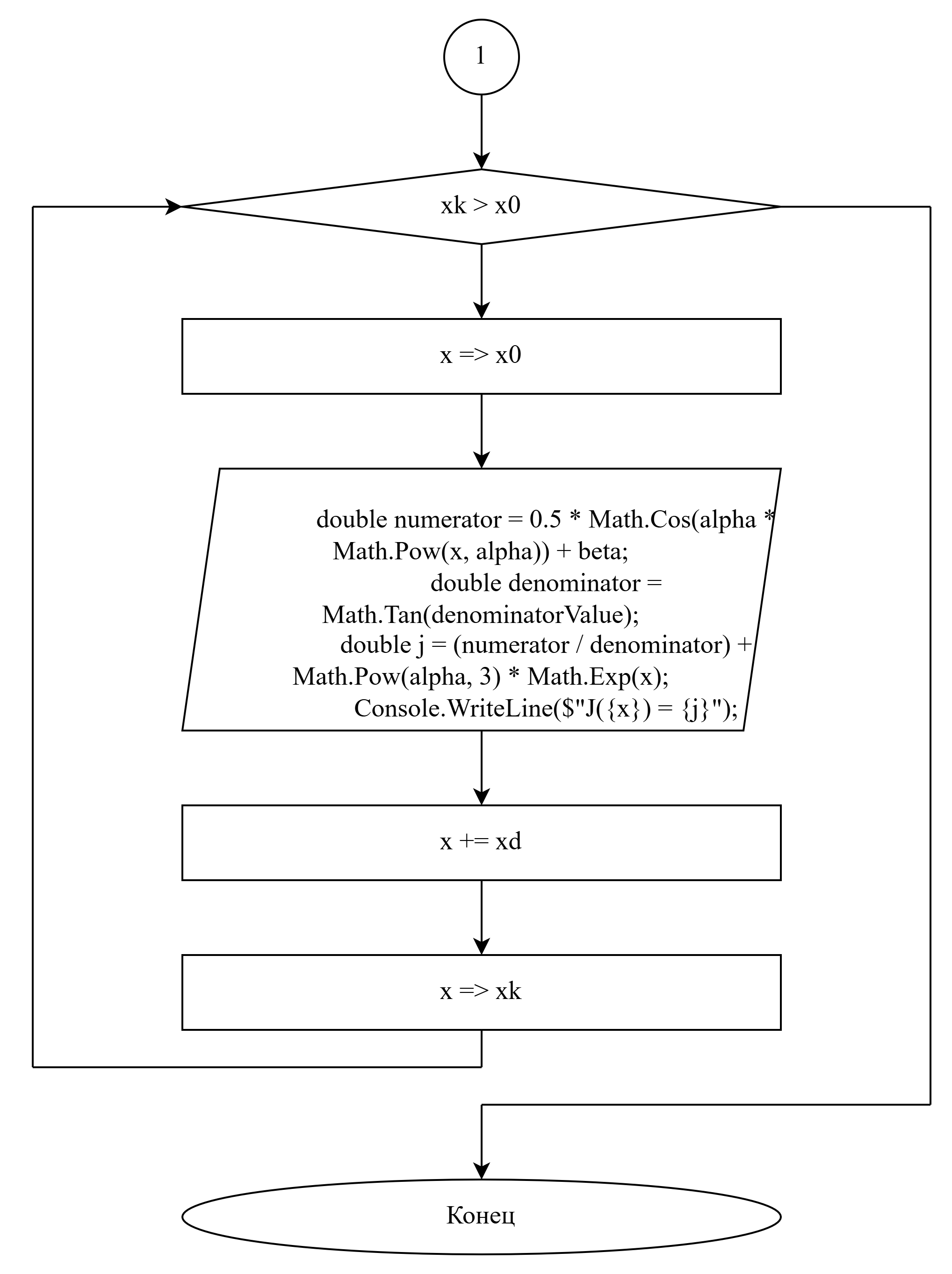


Рисунок 3.2 - Блок схема алгоритма

1. **Подбор тестовых примеров**

Введите A (A неравен 0): 1

Введите B: 2

Введите n: 1111

Введите x0: -0,5

Введите xk (xk > x0): 0,5

Введите xd (xd > 0): 0,01

Вывод : нет решений

2)

Введите A (A неравен 0): 0,1

Введите B: 21

Введите n: 2

Введите x0: -0,5

Введите xk (xk > x0): 0,5

Введите xd (xd > 0): 0,01

Вывод :

От -0,5 до 0 нет чисел

3)

Введите A (A неравен 0): 1

Введите B: 1

Введите n: 2

Введите x0: -0,5

Введите xk (xk > x0): 0,5

Введите xd (xd > 0): 0,01

Вывод :

Не определено (деление на ноль)

4)

Введите A (A неравен 0): 11

Введите B: 21

Введите n: 1

Введите x0: -0,5

Введите xk (xk > x0): 0,5

Введите xd (xd > 0): 0,01

Результаты:

J(-0,5) = 795,757024751974

J(-0,49) = 803,87044594543

J(-0,48) = 812,065408875175

J(-0,47) = 820,342732881138

J(-0,46) = 828,70324559268

J(-0,45) = 837,147782994601

J(-0,44) = 845,677189499044

J(-0,43) = 854,292318021856

J(-0,42) = 862,994030062325

J(-0,41) = 871,783195785558

J(-0,4) = 880,660694106951

J(-0,39) = 889,627412778378

J(-0,38) = 898,684248475846

J(-0,37) = 907,832106888419

J(-0,36) = 917,071902808317

J(-0,35) = 926,404560222083

J(-0,34) = 935,831012402796

J(-0,33) = 945,352202003271

J(-0,32) = 954,969081150256

J(-0,31) = 964,682611539597

J(-0,3) = 974,49376453238

J(-0,29) = 984,403521252057

J(-0,28) = 994,412872682545

J(-0,27) = 1004,52281976732

J(-0,26) = 1014,73437350951

J(-0,25) = 1025,048555073

J(-0,24) = 1035,46639588455

J(-0,23) = 1045,9889377369

J(-0,22) = 1056,61723289302

J(-0,21) = 1067,35234419128

J(-0,2) = 1078,19534515176

J(-0,19) = 1089,14732008358

J(-0,18) = 1100,20936419337

J(-0,17) = 1111,38258369475

J(-0,16) = 1122,66809591899

J(-0,15) = 1134,06702942671

J(-0,14) = 1145,58052412077

J(-0,13) = 1157,20973136023

J(-0,12) = 1168,9558140755

J(-0,11) = 1180,81994688464

J(-0,0999999999999997) = 1192,80331621083

J(-0,0899999999999997) = 1204,90712040097

J(-0,0799999999999997) = 1217,13256984558

J(-0,0699999999999997) = 1229,48088709978

J(-0,0599999999999997) = 1241,9533070056

J(-0,0499999999999997) = 1254,55107681541

J(-0,0399999999999997) = 1267,27545631671

J(-0,0299999999999997) = 1280,12771795803

J(-0,0199999999999997) = 1293,10914697625

J(-0,00999999999999969) = 1306,22104152511

J(3,08780778723872E-16) = 1319,46471280496

J(0,0100000000000003) = 1332,84148519399

J(0,0200000000000003) = 1346,35269638058

J(0,0300000000000003) = 1359,99969749709

J(0,0400000000000003) = 1373,78385325503

J(0,0500000000000003) = 1387,70654208145

J(0,0600000000000003) = 1401,76915625684

J(0,0700000000000003) = 1415,97310205433

J(0,0800000000000003) = 1430,31979988033

J(0,0900000000000003) = 1444,8106844166

J(0,1) = 1459,44720476365

J(0,11) = 1474,23082458572

J(0,12) = 1489,16302225711

J(0,13) = 1504,24529101003

J(0,14) = 1519,47913908393

J(0,15) = 1534,86608987631

J(0,16) = 1550,40768209506

J(0,17) = 1566,10546991237

J(0,18) = 1581,96102312009

J(0,19) = 1597,97592728677

J(0,2) = 1614,15178391615

J(0,21) = 1630,49021060739

J(0,22) = 1646,99284121677

J(0,23) = 1663,6613260211

J(0,24) = 1680,49733188275

J(0,25) = 1697,50254241635

J(0,26) = 1714,67865815711

J(0,27) = 1732,02739673092

J(0,28) = 1749,5504930261

J(0,29) = 1767,24969936689

J(0,3) = 1785,12678568867

J(0,31) = 1803,183539715

J(0,32) = 1821,42176713633

J(0,33) = 1839,84329179067

J(0,34) = 1858,44995584591

J(0,35) = 1877,2436199841

J(0,36) = 1896,22616358759

J(0,370000000000001) = 1915,39948492699

J(0,380000000000001) = 1934,76550135115

J(0,390000000000001) = 1954,32614947912

J(0,400000000000001) = 1974,08338539405

J(0,410000000000001) = 1994,03918483933

J(0,420000000000001) = 2014,19554341689

J(0,430000000000001) = 2034,55447678789

J(0,440000000000001) = 2055,11802087597

J(0,450000000000001) = 2075,88823207344

J(0,460000000000001) = 2096,86718745062

J(0,470000000000001) = 2118,05698496917

J(0,480000000000001) = 2139,45974369994

J(0,490000000000001) = 2161,07760404658

1. **Листинг**

using System;

namespace Zadanie7

{

class Program

{

static void Main()

{

double alpha, beta, n, x0, xk, xd;

// Ввод данных с проверками

alpha = ReadDouble("Введите A (A неравен 0): ", v => v != 0);

beta = ReadDouble("Введите B: ");

n = ReadDouble("Введите n: ");

x0 = ReadDouble("Введите x0: ");

xk = ReadDouble("Введите xk (xk > x0): ", v => v > x0);

xd = ReadDouble("Введите xd (xd > 0): ", v => v > 0);

// Вычисление и вывод значений функции J

Console.WriteLine("\nРезультаты:");

for (double x = x0; x <= xk; x += xd)

{

double denominatorValue = Math.Pow((Math.Pow(alpha, 2) + beta - n), 1.0 / 3.0);

// Проверяем, что аргумент тангенса не ведет к бесконечности

if (Math.Tan(denominatorValue) == 0)

{

Console.WriteLine($"J({x}) не определено (деление на ноль)");

continue;

}

double numerator = 0.5 \* Math.Cos(alpha \* Math.Pow(x, alpha)) + beta;

double denominator = Math.Tan(denominatorValue);

double j = (numerator / denominator) + Math.Pow(alpha, 3) \* Math.Exp(x);

Console.WriteLine($"J({x}) = {j}");

}

}

// Метод для безопасного ввода числа с условием

static double ReadDouble(string message, Func<double, bool> condition = null)

{

double value;

while (true)

{

Console.Write(message);

if (double.TryParse(Console.ReadLine(), out value) && (condition == null || condition(value)))

{

return value;

}

Console.WriteLine("Ошибка! Введите корректное значение.");

}

}

}

}

1. **Расчёт тестовых примеров на ПК:**

**1)**

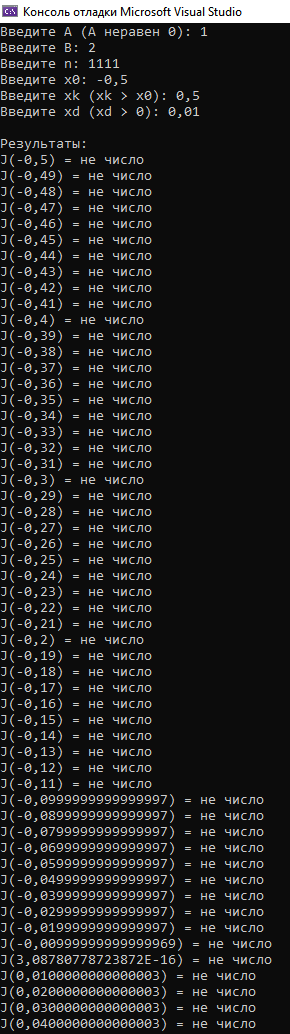


Рисунок 6.1 - Результат вычислений сделанных на ПК

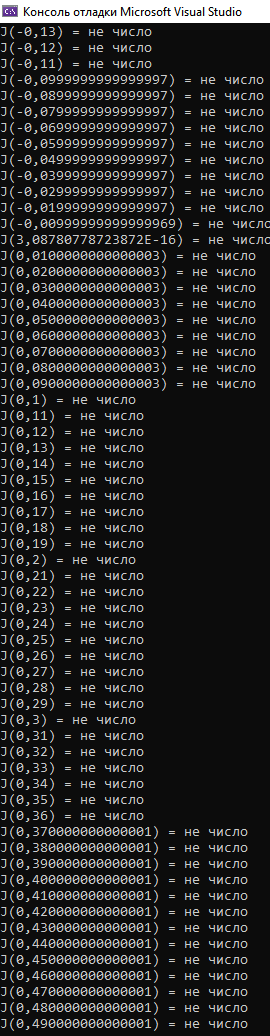


Рисунок 6.1.2 - Результат вычислений сделанных на ПК

**2)**

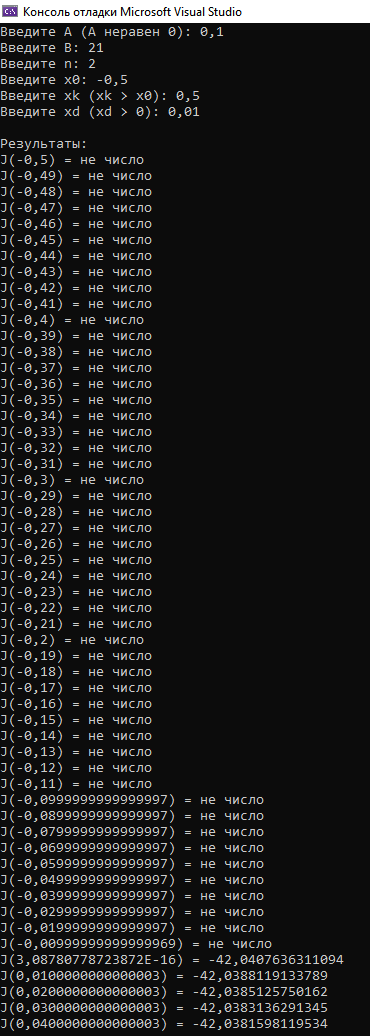


Рисунок 6.2 - Результат вычислений сделанных на ПК

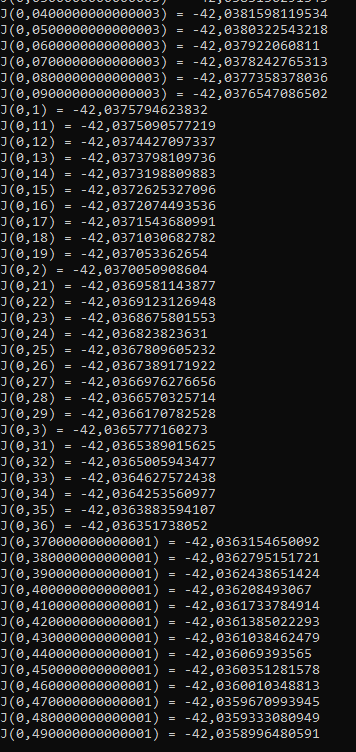


Рисунок 6.2.2 - Результат вычислений сделанных на ПК

**3)**

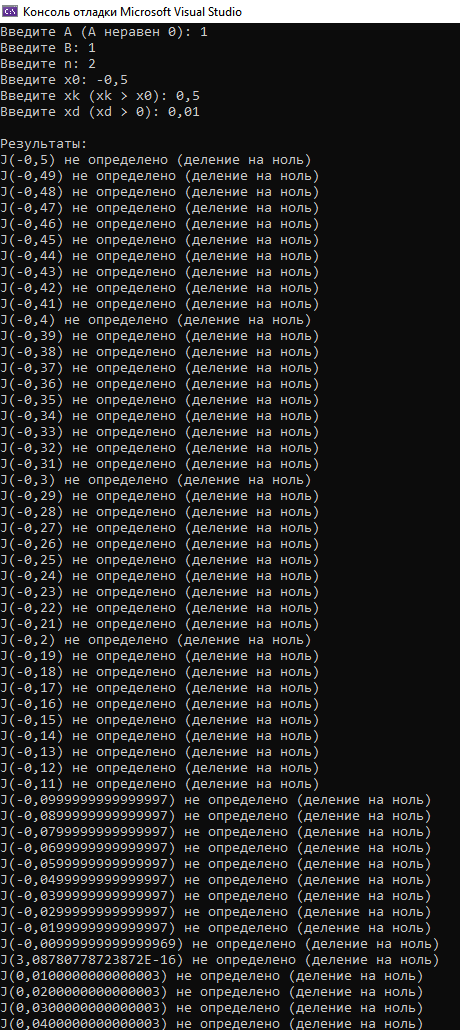


Рисунок 6.3 - Результат вычислений сделанных на ПК

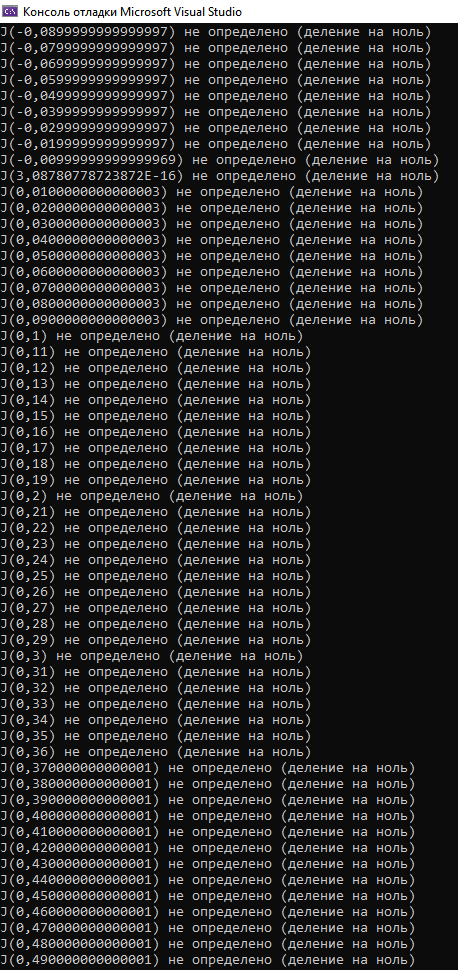


Рисунок 6.3.2 - Результат вычислений сделанных на ПК

**4)**

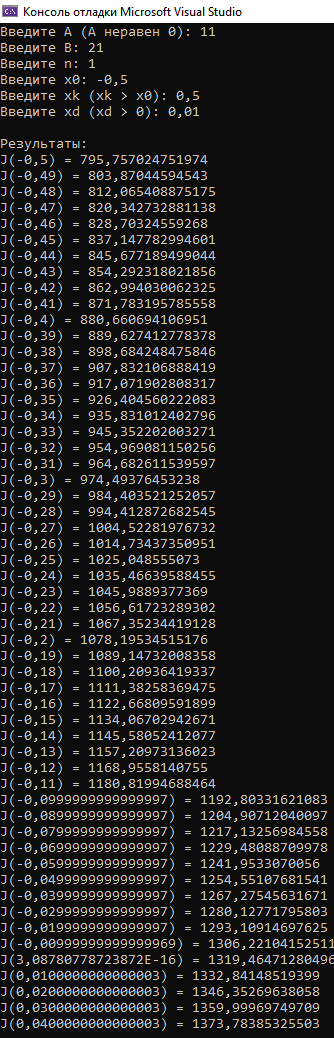


Рисунок 6.4 - Результат вычислений сделанных на ПК

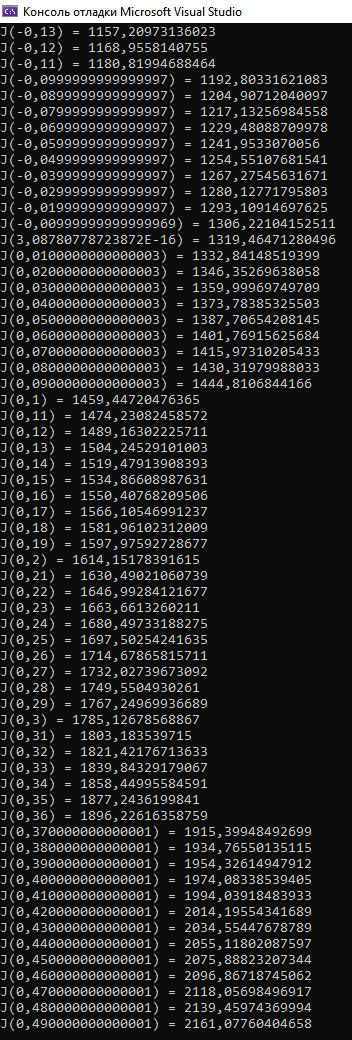


Рисунок 6.4.2 - Результат вычислений сделанных на ПК

1. **Вывод**

Работа позволила освоить использование циклов с предусловием для пошаговых вычислений. Программа выполняет корректные расчёты при заданных условиях и обрабатывает ошибки ввода данных.