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 № 6

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Циклы с пред и постусловием»

Выполнил: ст. гр. ТУУ-211

Воронов А.В.

Вариант №3

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

(дата приёмки)

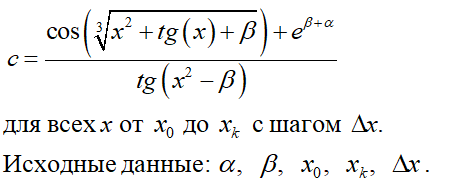
Москва – 2024 г.

1. **Цель работы:**

Научиться решать поставленную задачу, используя циклы с пред и постусловием, в интегрированной среде разработки (*IDE*) *Microsoft Visual Studio* на языке *Visual C#.*

1. **Формулировка задачи:**

Вычислить:



* 1. **Блок-схема алгоритма основной программы:**

НАЧАЛО

Ввод:  
alpha, beta, x0, xk, dx

x = x0

нет

x <= xk?

да

znamenatelArgument = x \* x - beta

(Math.Abs(znamenatelArgument % Math.PI - Math.PI / 2) < 1e-6)

нет

да

Вывод:  
знаменатель стремится к бесконечности

chislitel = Math.Cos(Math.Pow(x \* x + Math.Tan(x) + beta, 1.0 / 3.0)) + Math.Exp(beta + alpha);  
znamenatel = Math.Tan(znamenatelArgument);

znamenatel == 0

нет

да

Вывод:  
знаменатель равен нулю

c = chislitel / znamenatel

Вывод: c

x = x + dx

КОНЕЦ

* 1. **Блок-схема алгоритма метода ввода данных:**

Прием параметров:  
prompt, positive

НАЧАЛО

valid = false

Вывод: prompt  
Ввод: input

double.TryParse(input, out value) = true?

нет

Вывод: “Некорректный ввод”

да

positive && value <= 0

да

Вывод: “Значение должно быть положительным”

нет

valid = true

valid = false

да

нет

Возврат значения:  
value

КОНЕЦ

1. **Подбор тестовых примеров:**

А) Пример-1.

Входные данные:  
Alpha=1, beta = 2, x0 = 0.5, xk = 2.5, deltaX = 0.5.  
Ожидаемый результат: Программа должна вычислить значение c для каждого x от 0.5 до 2.5 с шагом 0.5. Результаты должны быть числовыми значениями.

Б) Пример-2.

Входные данные:  
Alpha=0.5, beta = 1.5, x0 = 0.1, xk = 1.0, deltaX = 0.2.  
Ожидаемый результат: Программа должна вычислить значение c для каждого x от 0.1 до 1.0 с шагом 0.2. Результаты должны быть числовыми значениями.

В) Пример-3.

Входные данные:  
Alpha=100, beta = 50, x0 = 10, xk = 20, deltaX = 2.  
Ожидаемый результат: Программа должна вычислить значение c для каждого x от 10 до 20 с шагом 2. Результаты должны быть числовыми значениями. Тест покажет, как программа обрабатывает ситуации, когда экспоненциальные функции и тангенс могут давать большие значения.

Г) Пример-4. Проверка на некорректный ввод.  
  
Входные данные:  
Alpha=2, beta = 1, x0 = 0, xk = 2, deltaX = -0.5.  
Ожидаемый результат: Программа должна вывести сообщение о том, что шаг должен быть положительным, и заставить пользователя ввести корректное значение для шага, прежде чем продолжить выполнение программы.

1. **Листинг (код) программы:**

using System;

class Program

{

static void Main()

//основная часть программы

{

//объявляем еременные для исходных данных

double alpha, beta, x0, xk, dx;

//ввод данных с контролем ввода

//для ввода используем отдельный метод, где выполняются проверки ввода

alpha = GetValidatedInput("Введите значение alpha (действительное число): ");

beta = GetValidatedInput("Введите значение beta (действительное число): ");

x0 = GetValidatedInput("Введите значение x0 (начальное значение, действительное число): ");

xk = GetValidatedInput("Введите значение xk (конечное значение, действительное число): ");

dx = GetValidatedInput("Введите значение deltaX (шаг, положительное действительное число): ", positive: true);

//цикл по значениям x от x0 до xk с шагом delta-x

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

{

//проверяем ограничения для тангенса в знаменателе

double znamenatelArgument = x \* x - beta;

if (Math.Abs(znamenatelArgument % Math.PI - Math.PI / 2) < 1e-6)

{

Console.WriteLine($"Для x = {x} знаменатель стремится к бесконечности (тангенс в точке асимптоты). Пропускаем это значение.");

continue;

}

//вычисляем числитель и знаменатель

double chislitel = Math.Cos(Math.Pow(x \* x + Math.Tan(x) + beta, 1.0 / 3.0)) + Math.Exp(beta + alpha);

double znamenatel = Math.Tan(znamenatelArgument);

//проверка на деление на ноль (на случай непредвиденных ситуаций)

if (znamenatel == 0)

{

Console.WriteLine($"Для x = {x} знаменатель равен нулю. Пропускаем это значение.");

continue; //если знаменатель дроби равен нулю, то пропускаем текущее значение x

}

//вычисляем значение c

double c = chislitel / znamenatel;

//вывод результата на экран

//используем форматированный вывод (4 и 6 знаков после запятой)

Console.WriteLine($"При x = {x:F4}, значение c = {c:F6}");

}

}

//метод для проверки корректного ввода данных

static double GetValidatedInput(string prompt, bool positive = false)

{

double value;

bool valid = false;

//цикл для повторного запроса данных до получения корректного значения

do

{

Console.Write(prompt);

string input = Console.ReadLine();

//проверяем, корректен ли ввод

if (double.TryParse(input, out value))

{

//если значение должно быть положительным, проверяем это условие

if (positive && value <= 0)

{

Console.WriteLine("Значение должно быть положительным. Попробуйте снова.");

}

else

{

valid = true; //ввод корректный

}

}

else

{

Console.WriteLine("Некорректный ввод. Пожалуйста, введите действительное число.");

}

} while (!valid);

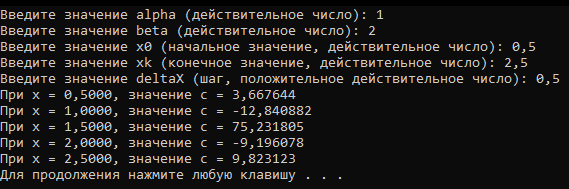
return value; //возвращаем введенное значение

}

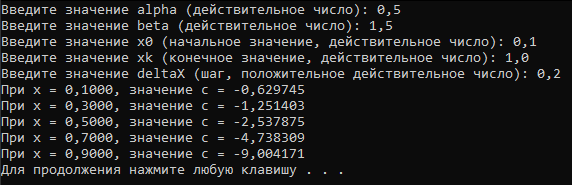
}

**6. Расчет тестовых примеров на ПК:**

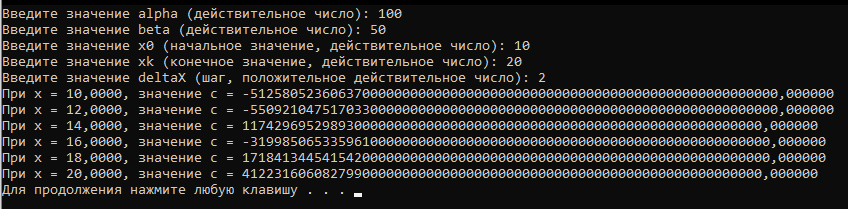
1)

****

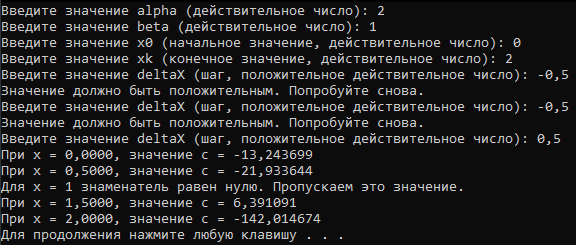
2)

****

3)

****

4)

****

1. **Вывод:**

Решил поставленную задачу, используя циклы с пред и постусловием, в интегрированной среде разработки (*IDE*) *Microsoft Visual Studio* на языке *Visual C#.* В основной части программы используется цикл for с предусловием. В методе ввода данных с клавиатуры используется цикл do-while с постусловием.