ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
О ПРАКТИЧЕСКОЙ РАБОТЕ №3

По дисциплине «Введение в языки программирования»

Выполнил: ст. гр. ТКИ – 112

Павлючков Д. Д.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[**Задание 3–1** 3](#_Toc91707704)

[**Условие задания** 3](#_Toc91707705)

[**Блок-схема алгоритма** 4](#_Toc91707706)

[**Программа на языке С++** 5](#_Toc91707707)

[**Решение тестовых примеров на С++** 6](#_Toc91707708)

[**Решение тестовых примеров в Excel** 7](#_Toc91707709)

[**Задание 3–2** 8](#_Toc91707710)

[**Условие задания** 8](#_Toc91707711)

[**Блок-схема алгоритма** 9](#_Toc91707712)

[**Программа на языке С++** 10](#_Toc91707713)

[**Решение тестовых примеров на С++** 11](#_Toc91707714)

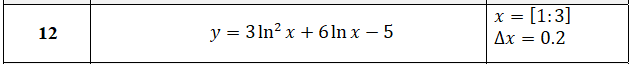
[**Решение тестовых примеров в Excel** 12](#_Toc91707715)

# **Задание 3–1**

## **Условие задания**

Протабулировать заданную в таблице функцию. Использовать данные в таблице значения шага и интервала в качестве ввода пользователя для решения тестового примера. При невозможности расчёта функции в конкретной точке выводить её значение и надпись, означающую отсутствие решения. При решении данного задания в MatLab необходимо построить график.

Таблица



## **Блок-схема алгоритма**

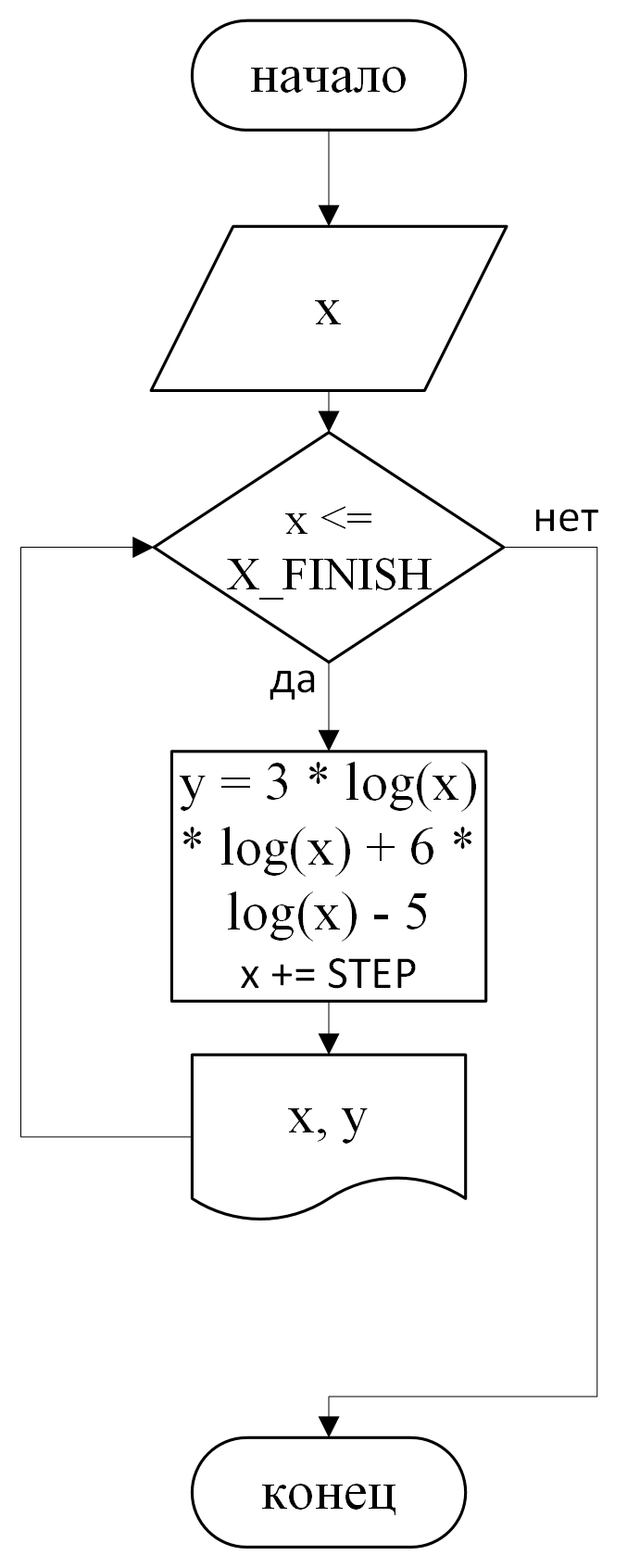


Рисунок 1 – Блок-схема 3-1

## 

## **Программа на языке С++**

#include <iostream>

#include <cmath>

#include <iomanip>

/\*\*

\* \brief Рассчитывает значение функции в заданной точке

\* \param x Заданная точка

\* \return Возвращает значение функции в заданной точке

\*/

double calculateFunc(double x);

/\*\*

\* \brief Проверяет, существует ли функция в заданной точке

\* \param x Заданная точка

\* \return Возвращает true, если значение функции в заданной точке существует

\*/

bool isCalculated(double x);

int main()

{

const double X\_START = 1;

const double X\_FINISH = 3;

const double STEP = 0.2;

double x = X\_START;

while (x <= X\_FINISH)

{

if (isCalculated(x))

{

const double y = calculateFunc(x);

std::cout << "x = " << std::setw(5) << std::left << std::setprecision(5) << x << " y = " << y << "\n";

}

else

{

std::cout << "x = " << x << " y = не существует" << "\n";

}

x += STEP;

}

return 0;

}

double calculateFunc(const double x)

{

return 3 \* log(x) \* log(x) + 6 \* log(x) - 5;

}

bool isCalculated(const double x)

{

return x > 0;

}

## **Решение тестовых примеров на С++**

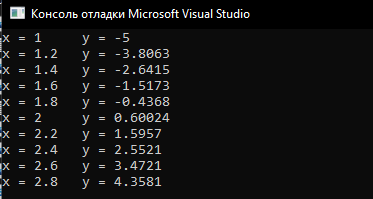


Рисунок 2 – Решение тестового примера для задания 3-1 на С++

## **Решение тестовых примеров в Excel**

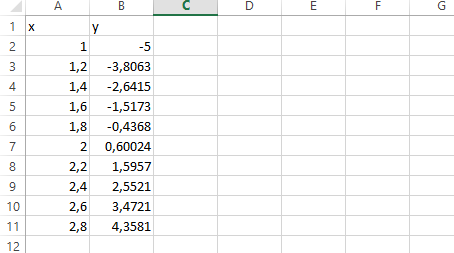


Рисунок 3 - Решение тестового примера задания 3-1 в Excel

# **Задание 3–2**

## **Условие задания**

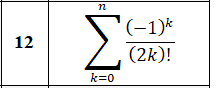
Составьте две программы:

a) вычислить сумму первых n членов последовательности (k = 1, 2, 3 ..., n).

b) вычислить сумму всех членов последовательности, не меньших заданного числа e.

Помните о проверке пользовательского ввода. Все результаты вывести на экран. Отчёт дополнить блок-схемой. При вычислении факториалов рекомендуется отказаться от использования рекурсивных методов.

Таблица 2



## **Блок-схема алгоритма**

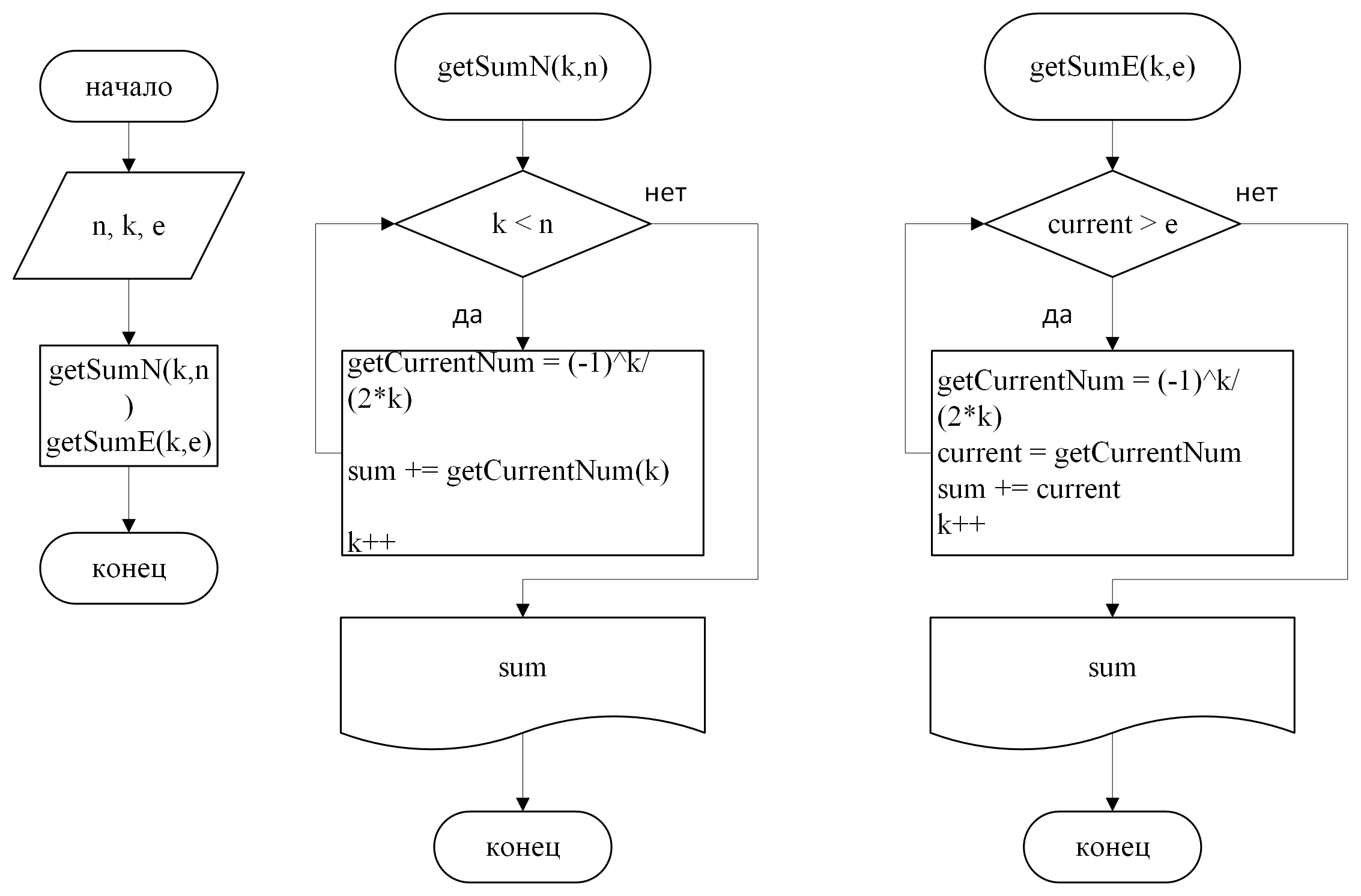


Рисунок 4 – Блок-схема 3-2

## 

## **Программа на языке С++**

#include <iostream>

using namespace std;

/\*\*

\* \brief Функция для нахождения текущего элемента.

\* \param k Переменная элемента ряда.

\* \return Возвращает значение текущего элемента.

\*\*/

double getCurrentNum(int k);

/\*\*

\* \brief Функция для вычисления суммы первых n элементов ряда.

\* \param n Количество элементов ряда.

\* \param k Переменная элемента ряда.

\* \return Возвращает сумму первых n элементов ряда.

\*\*/

double getSumN(int k, int n);

/\*\*

\*\brief Функция для вычисления суммы членов ряда не меньших числа e.

\*\param e Заданное число.

\*\param k Переменная элемента ряда.

\*\return Возвращает сумму членов ряда не меньших числа e.

\*\*/

double getSumE(int k, double e);

/\*\*

\* \brief Точка входа в программу

\* \return В случае успеха возвращает 0

\*/

int main()

{

int k = 1;

int n;

double e = 0.0;

cout << "Введите количество членов ряда: ";

cin >> n;

cout << "\nВведите значение e: ";

cin >> e;

cout << "\nСумма первых n членов: " << getSumN(k, n);

cout << "\nСумма членов не меньших числа е: " << getSumE(k, e) << endl;

return 0;

}

double getCurrentNum(const int k)

{

return (pow(-1, k)) / (2 \* k);

}

double getSumN(int k, int n)

{

double sum = 0;

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

{

sum += getCurrentNum(k);

k++;

}

return sum;

}

double getSumE(int k, double e)

{

double sum = 0;

double current = getCurrentNum(k);

while (current > e)

{

sum += current;

k++;

current = getCurrentNum(k);

}

return sum;

}}

## 

## **Решение тестовых примеров на С++**

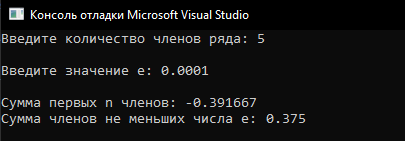


Рисунок 5 – Решение тестового примера для задания 3-2

## **Решение тестовых примеров в Excel**

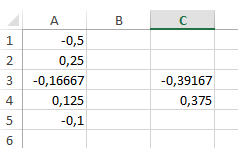


Рисунок 6 – Решение тестового примера для задания 3-2 в Excel