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 – 111

Самохвалова М. 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 1

|  |  |  |
| --- | --- | --- |
| Вариант | Функция | Константы |
| 1 |  |  |

## **Блок-схема алгоритма**

2 > x > 3

x = x + step

y

Начало

x, step

Конец

True

False

Рисунок 1 – Блок-схема 3-1

## **Программа на языке С++**

#define \_USE\_MATH\_DEFINES // for C++

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

/\*\*

\* \brief Возможность расчета функции в точке.

\* \param точка x - Входной параметр .

\* \return true, если функция определена в заданной точке x.

\*/

bool IsCalculated(const double x);

/\*\*

\* \brief Расчет функции в точке x.

\* \param точка x - Входной параметр.

\* \return Значение функции в точке x

\*/

double Calculation(const double x);

/\*\*

\* \brief Точка входа в программу.

\* \return Код ошибки (0 - успех).

\*/

int main()

{

const auto leftBound = 2.0;

const auto rightBound = 3.0;

const auto step = 0.1;

auto x = leftBound;

cout << " x" << setw(15) << "y\n";

while ((x < rightBound) || (abs(x - rightBound) < step))

{

if (IsCalculated(x))

{

const auto y = Calculation(x);

cout << setw(10) << setprecision(2) << x << setw(15)

<< setprecision(5) << y << '\n';

}

else

{

cout << setw(10) << setprecision(2) << x << setw(15)

<< "not defined \n";

}

x += step;

}

return 0;

}

double Calculation(const double x)

{

return 3 \* sin(sqrt(x)) + 0.39 \* x - 3.8;

}

bool IsCalculated(const double x)

{

return x > 0;

}

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

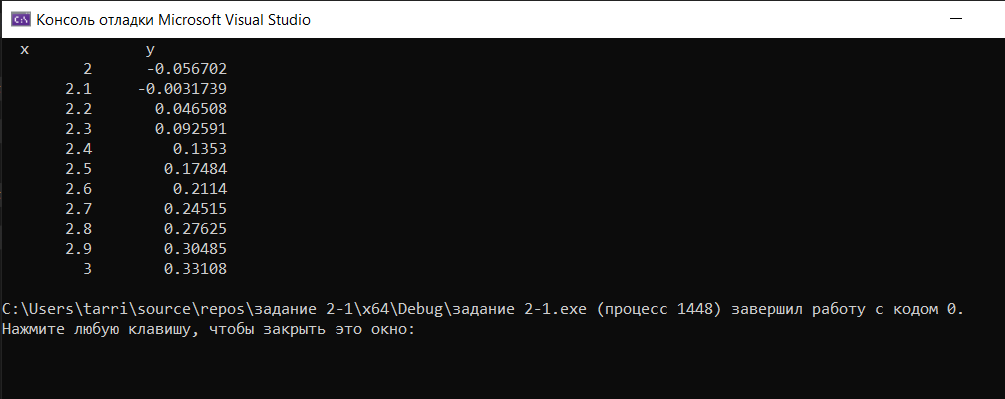


Рисунок 2 – Решение тестового примера для задания 3-1 на С++

## **Решение тестовых примеров в Excel**

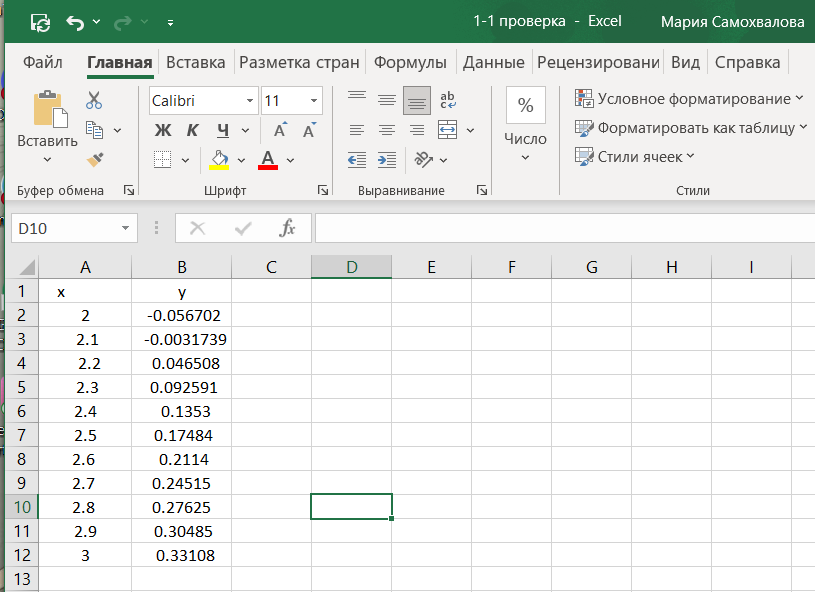


Рисунок 3 - Решение тестового примера задания 3-1 в Excel

# **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

|  |  |
| --- | --- |
| Номер варианта | Ряд |
| 1 |  |

## **Блок-схема алгоритма**

Начало

k, n, e

False

k < n

cur = 0

k = 0

True

getCurrent = 1.0 / (k + 2) \* (k + 3)

cur = cur + getCurrent

sum1 = sum1 + cur

k = k +1

abs(cur)< e

False

getCurrent = 1.0 / (k + 2) \* (k + 3)

cur = cur \* getCurrent

sum2 = sum2+ cur

k = k +1

True

sum1

sum2

Конец

Рисунок 4 – Блок-схема 3-2

## **Программа на языке С++**

#include <iostream>

using namespace std;

/\*\*

\* \brief функция определения текущего элемента ряда.

\* \param k - переменная для вычисления элемента ряда.

\* \return значение текущего элемент ряда.

\*\*/

double getCurrent(const int k);

/\*\*

\* \brief функция вычисления суммы первых n элементов ряда.

\* \param n - количество элементов ряда.

\* \param k - переменная для вычисления элемента ряда.

\* \return сумма первых n элементов ряда.

\*\*/

double getSumm1(int n, int k);

/\*\*

\*\brief вычисление суммы членов ряда не меньших числа e.

\*\param e - константа по условию.

\*\param k - переменная для вычисление элемента ряда.

\*\return сумма членов ряда не меньших числа e.

\*\*/

double getSumm2(double e, int k);

/\*\*

\*\brief точка входа в программу.

\*\return 0 в случае успеха.

\*/

int main()

{

int n;

double e = 0.0;

int k = 0;

cout << "Введите количество членов ряда" << endl;

cin >> n;

cout << "Введите e ";

cin >> e;

cout << getSumm1(n, k) << endl;

cout << getSumm2(e, k);

return 0;

}

double getCurrent(const int k) {

return 1.0 / ((k + 2) \* (k + 3));

}

double getSumm1(int n, int k) {

double sum;

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

sum = sum + getCurrent(k);

k = k + 1;

}

return sum;

}

double getSumm2(double e, int k) {

double cur, sum;

cur = getCurrent(k);

while (cur > e) {

sum = sum + cur;

k++;

cur = getCurrent(k);

}

return sum;

}

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

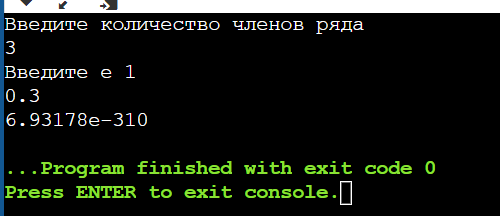


Рисунок 5 – Решение тестового примера для задания 3-2

## **Решение тестовых примеров в Excel**

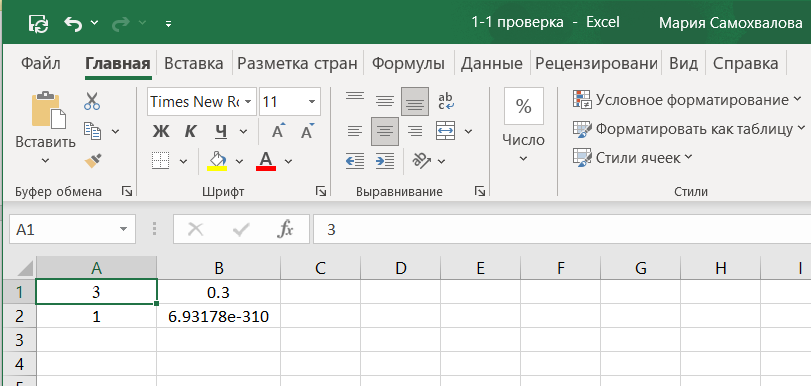


Рисунок 6 – Решение тестового примера для задания 3-2 в Excel