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

Соколов И.Д.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2022

Оглавление

[Задание 3–1 3](#_Toc85287969)

[Формулировка задания 3](#_Toc85287970)

[Блок-схема алгоритма 4](#_Toc85287971)

[Программа на языке С++ 5](#_Toc85287972)

[Решение тестовых примеров на С++ 6](#_Toc85287973)

[Решение тестовых примеров в Excel 7](#_Toc85287974)

[Задание 3–2 8](#_Toc85287975)

[Формулировка задания 8](#_Toc85287976)

[Блок-схема алгоритма 9](#_Toc85287977)

[Программа на языке С++ 10](#_Toc85287978)

[Решение тестовых примеров на С++ 12](#_Toc85287979)

[Решение тестовых примеров в Excel 13](#_Toc85287980)

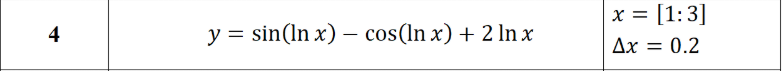
**Задание 3-1**

**Формулировка 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**

****

**Блок-схема алгоритма**

Начало

x, step

x>=0

**True**

**False**

y = sin(log(x)) - cos(log(x)) + 2.0 \* (log(x))

x=x + step

y

Конец

Рисунок 1 – Блок-схема алгоритма

**Программа на языке C++**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | #include <cmath> | |  | #include <iostream> | |  | #include <iomanip> | |  |  | |  | using namespace std; | |  |  | |  | /\*\* | |  | \* \brief Математическая функция, рассчитывающая значение функции y. | |  | \* \param x - константа | |  | \* \return Возвращает значение функции y. | |  | \*\*/ | |  | double getY(double x); | |  |  | |  | /\*\* | |  | \* \brief Проверка y при заданном х | |  | \* \param x Значение x | |  | \* \return true, y существует при заданном х | |  | \*/ | |  | bool isCalculated(double x); | |  |  | |  | /\*\* | |  | \* \brief Точка входа в программу. | |  | \* \return Возвращает 0 в случае успешного выполнения. | |  | \*/ | |  | int main() | |  | { | |  | double x = 1.0; | |  | const double step = 0.2; | |  | const double rightBorder = 3.0; | |  |  | |  | while(x < rightBorder) | |  | { | |  | if (isCalculated(x)) | |  | { | |  | const double y = getY(x); | |  | std::cout << "x = " << std::setw(10) << std::left << std::setprecision(4) << x << " y = " << y << "\n"; | |  | } | |  | else | |  | { | |  | cout << "При x = " << x << " y не существует" << "\n"; | |  | } | |  |  | |  | x+= step; | |  | } | |  | return 0; | |  | } | |  |  | |  | double getY(const double x) | |  | { | |  | return sin(log(x)) - cos(log(x)) + 2.0 \* (log(x)); | |  | } | |  | bool isCalculated(const double x) | |  | { | |  | return x >= 0; | |  | } | |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Решение тестовых примеров на C++**

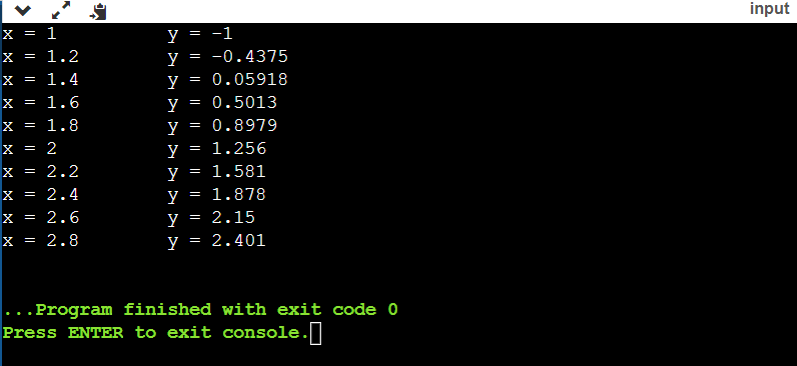
****

Рисунок 2 – Решение тестовых примеров на языке C++

**Решение тестовых примеров в Excel**

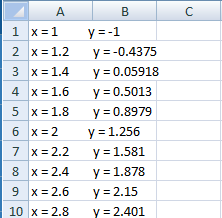
****

Рисунок 3 – Решение тестовых примеров в Excel

**Задание 3-2**

**Формулировка задания**

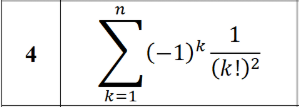
Составьте две программы:

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**

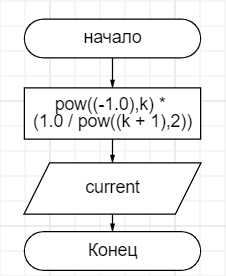
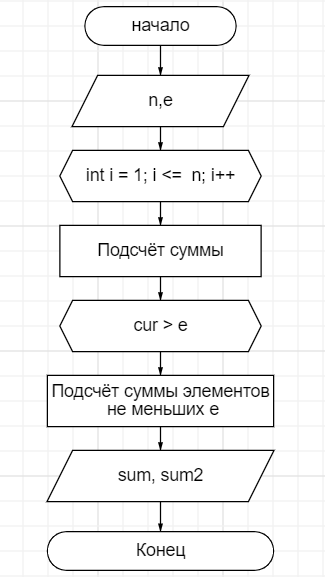


Рисунок 4 – Блок-схема алгоритма

**Программа на языке C++**

|  |  |
| --- | --- |
|  | #include <iostream> |
|  | #include <cmath> |
|  |  |
|  | using namespace std; |
|  |  |
|  | /\*\* |
|  | \* \brief функция определения текущего элемента ряда. |
|  | \* param k - переменная для вычисления элемента ряда. |
|  | \* \return значение текущего элемент ряда. |
|  | \*/ |
|  | double getCurrent(const size\_t 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 getSumm(const int n, size\_t 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 getSummE(const double e, size\_t k); |
|  | /\*\* |
|  | \* \brief точка входа в программу. |
|  | \* return 0 в случае успеха. |
|  | \*/ |
|  | int main() |
|  | { |
|  | int n; |
|  | double e = 0.0; |
|  | size\_t k = 1; |
|  | cout<<"Введите количество членов ряда"<< endl; |
|  | cin >> n; |
|  | cout<< "Введите e "; |
|  | cin>>e; |
|  | cout<< getSumm(n, k)<< endl; |
|  | cout<< getSummE(e, k); |
|  | return 0; |
|  | } |
|  | double getCurrent(const size\_t k){ |
|  | return pow((-1.0),k) \* (1.0 / pow((k + 1),2)); |
|  | } |
|  | double getSumm(const int n, size\_t k){ |
|  | double sum; |
|  | for (int i = 1; i <= n; i++){ |
|  | sum = sum + getCurrent(k); |
|  | k=k+1; |
|  | } |
|  | return sum; |
|  | } |
|  | double getSummE(const double e, size\_t k){ |
|  | double cur, sum; |
|  | cur = getCurrent(k); |
|  | while (cur > e){ |
|  | sum = sum + cur; |
|  | k++; |
|  | cur = getCurrent(k); |
|  | } |
|  | return sum; |
|  | } |

**Решение тестовых примеров на C++**

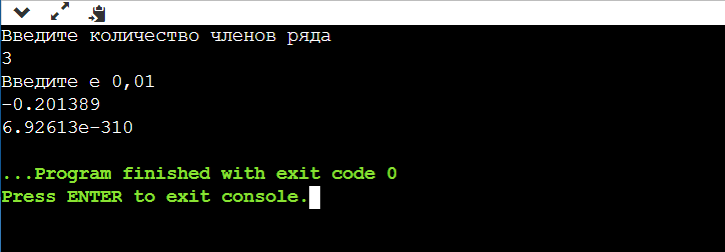
****

Рисунок 5 – Решение тестовых примеров на языке C++

**Решение тестовых примеров в Excel**

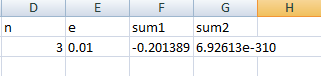
****

Рисунок 6 – Решение тестовых примеров в Excel