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 ЛАБОРАТОРНОЙ РАБОТЕ №3.1

По дисциплине «Языки программирования»

ВАРИАНТ 3

Выполнил: ст. гр. ТКИ-142

Величко Иван Сергеевич

Проверил: к.т.н., доц. Васильева М. А.

(Проверил: к.т.н, доц. Балакина Е. П.)

Москва 2023

1. Формулировка задания

Протабулировать заданную в таблице функцию (Таблица 1). 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.

Таблица 1 – Исходные данные

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Функция** | **Константы** |
| 3 |  | 𝑥=[0.4:1]  ∆𝑥=0.05 |

1. Блок-схема алгоритма

Ниже представлены:

Блок-схема основного алгоритма (Рисунок 1).

Блок-схема функции check\_step (Рисунок 2).

Блок-схема функции calc(Рисунок 3.1)

Блок-схема функции calc(Рисунок 3.2)

Блок-схема функции input(Рисунок 4)

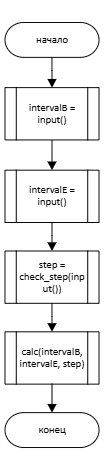


Рисунок 1 – Блок-схема основного алгоритма

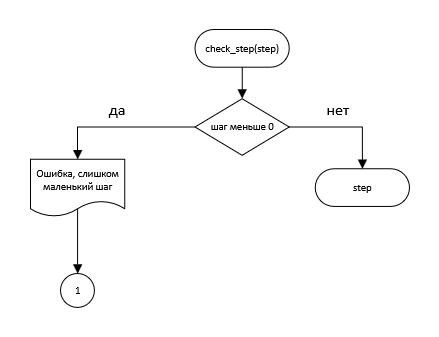


Рисунок 2 – Блок-схема функции check\_step

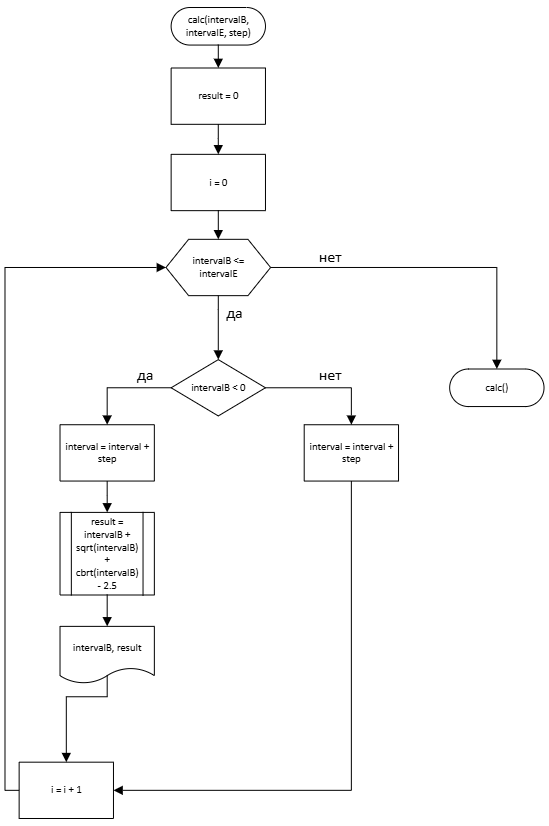


Рисунок 3 – Блок-схема функции calc()



Рисунок 4 – Блок-схема функции input()

1. Текст программы на языке C

#include <stdio.h>

#include <errno.h>

#include <math.h>

#include <float.h>

#include <stdlib.h>

/\*\*

\* @brief принимает числовое значение из stdin

\* @return числовое значение полученное из stdin

\*/

double input();

/\*\*

\* @brief проверяет что шаг является положительным числом

\* @param step параметр принимающий значение шага

\* @return числовое значение параметра step

\*/

double check\_step(const double step);

/\*\*

\* @brief табулирует функцию в пределах ОДЗ

\* @param intervalB параметр принимающий значение начала интервала

\* @param intervalE параметр принимающий значение конца интервала

\* @param step параметр принимающий значение шага

\*/

void calc(double intervalB, const double intervalE, const double step);

/\*\*

\* @brief - точка входа в программу

\* @return 0 при успешном выполнении программы

\*/

int main() {

puts("Пожалуйста введите значения начала и конца интервала:");

const double intervalB = input();

const double intervalE = input();

puts("Пожалуйста введите значение шага табулирования:");

const double step = check\_step(input());

calc(intervalB, intervalE, step);

return 0;

}

double input() {

double value = 0.0;

int s = scanf\_s("%lf", &value);

if (s != 1) {

errno = EIO;

perror("Ошибка, не числовое значение\n");

exit(EXIT\_FAILURE);

}

return value;

}

double check\_step(const double step) {

if (step < DBL\_EPSILON) {

errno = EIO;

perror("Ошибка, слишком маленький шаг\n");

exit(EXIT\_FAILURE);

}

return step;

}

void calc(double intervalB, const double intervalE, const double step) {

double result = 0;

for (; intervalB <= intervalE + DBL\_EPSILON; intervalB += step) {

if (intervalB < 0) // ОДЗ

puts("Значение не может быть посчитанно");

result = intervalB + sqrt(intervalB) + cbrt(intervalB) - 2.5;

printf("Для x = %0.2f y = %0.2f\n", intervalB, result);

}

}

1. Результаты выполнения программы

Результаты выполнения программы представлены ниже (Рисунок 3).

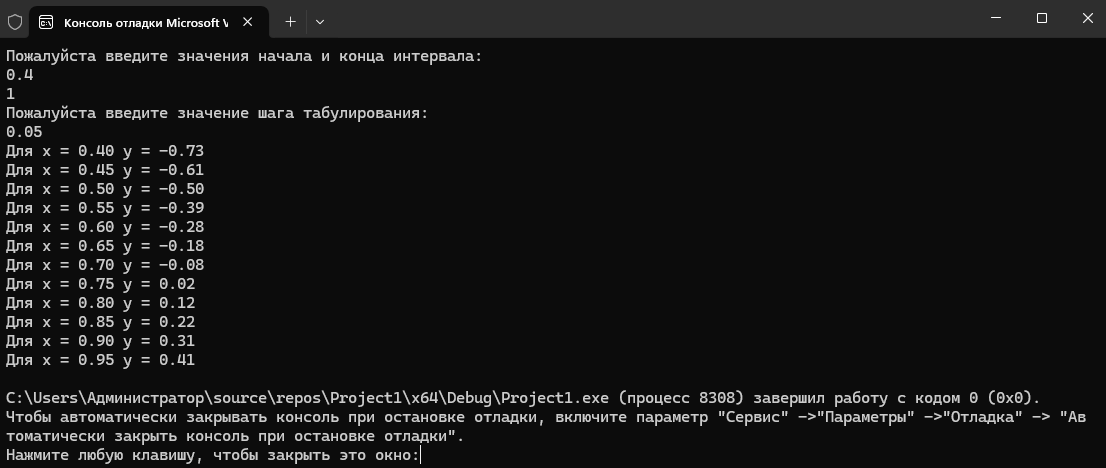


Рисунок 5 – Результаты выполнения программы

1. Выполнение тестовых примеров

Результат табулирования и построения графика функции в программе MATHLAB приведен ниже (Рисунок 5).

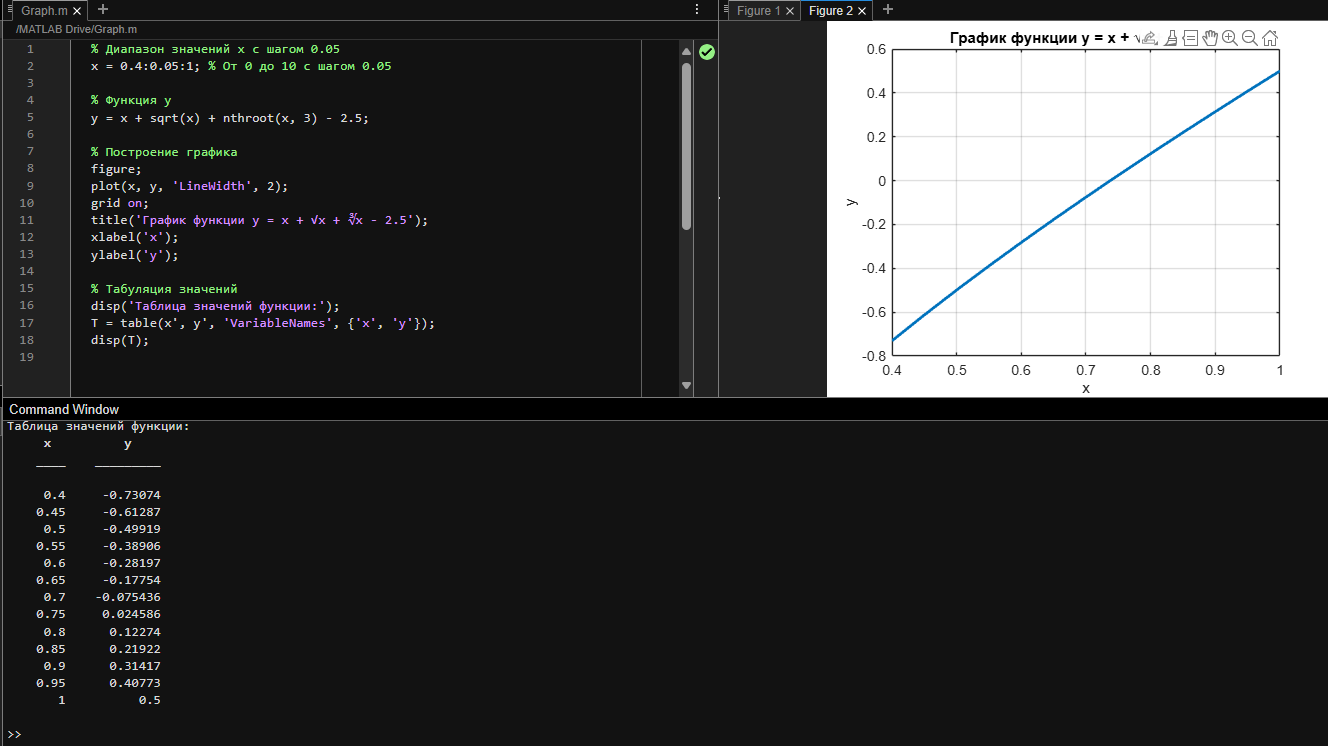


Рисунок 6 – Результат расчета сопротивления параллельного соединения

1. Отметка о выполнении задания в веб-хостинге системы контроля версий

Отметка о выполнение задания представлена ниже (Рисунок 6)

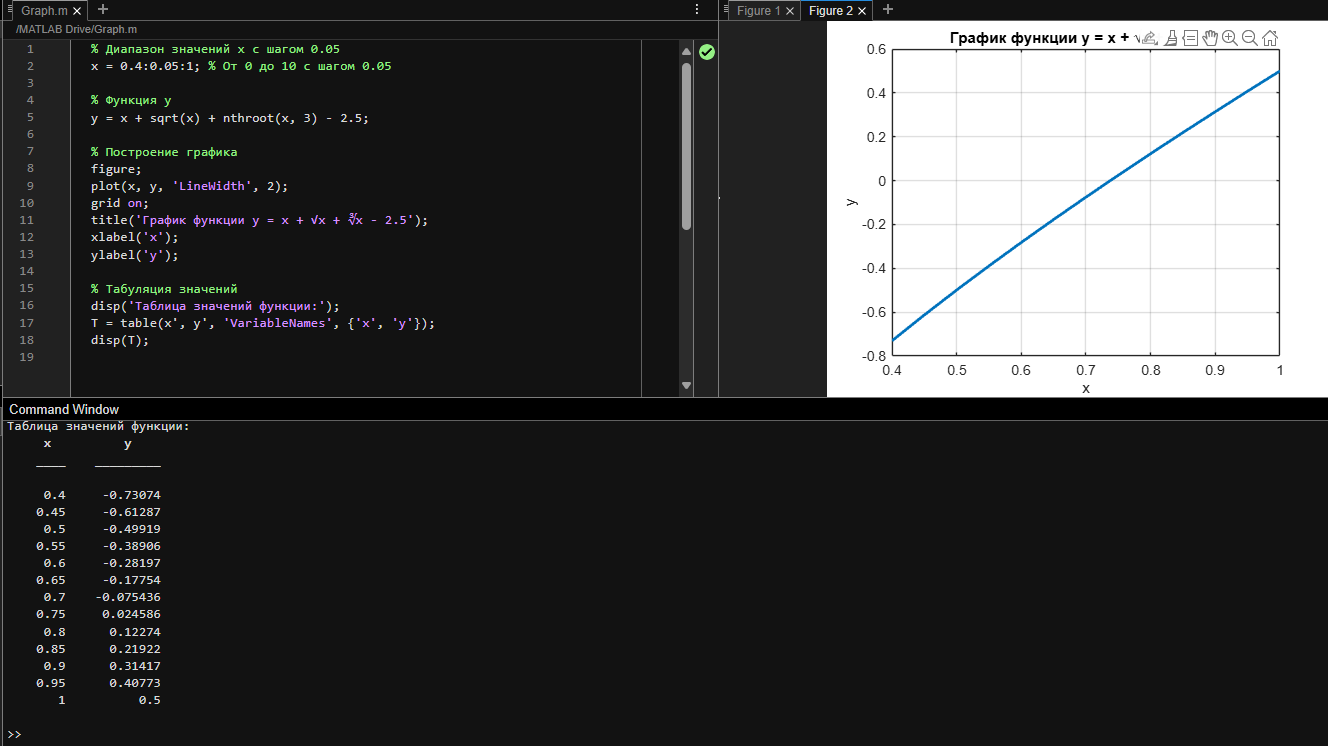


Рисунок 7 – Отметка о выполнении задания