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 №1

По дисциплине «Языки программирования»

ВАРИАНТ 7

Выполнил: ст. гр. ТКИ - 141

Костин Роман Сергеевич

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

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

Москва 2024

1. Решение задачи 1.1
   1. Формулировка задачи

Создать консольное приложение, вычисляющее значения переменных по представленным в таблице формулам. Расчёт примера осуществить по заданным константам. Вывести на экран значения исходных данных, а также результат вычислений.

Таблица  - Условие задачи 1.1

|  |  |  |
| --- | --- | --- |
| Вариант | Формулы | Константы |
| 7 |  | x=0.5  y=0.05  z=0.7 |

* 1. Блок–схема алгоритма

Блок-схема алгоритма представлена ниже (Рисунок 1, Рисунок 2, Рисунок 3)

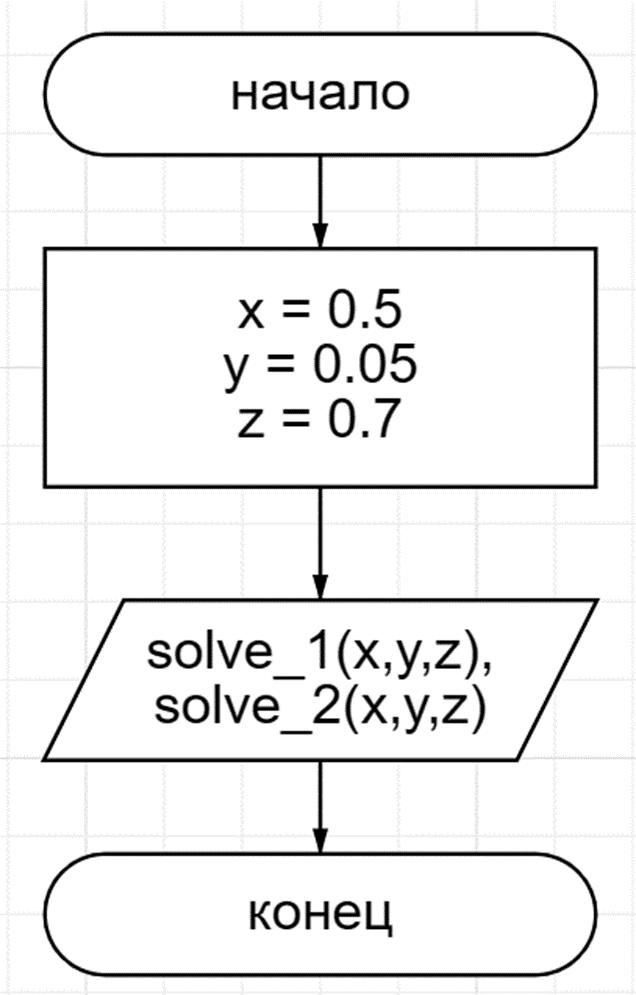


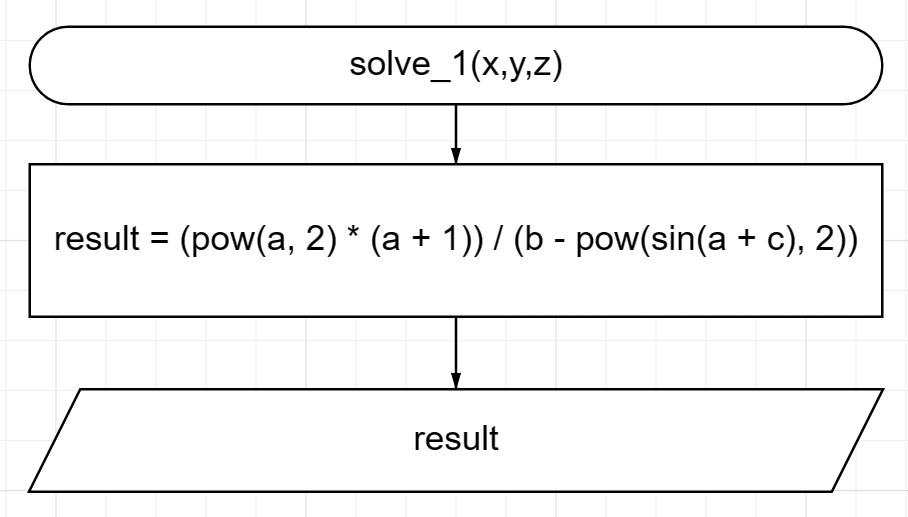
Рисунок  - блок-схема функции main()

Рисунок  - блок-схема функции solve\_1()

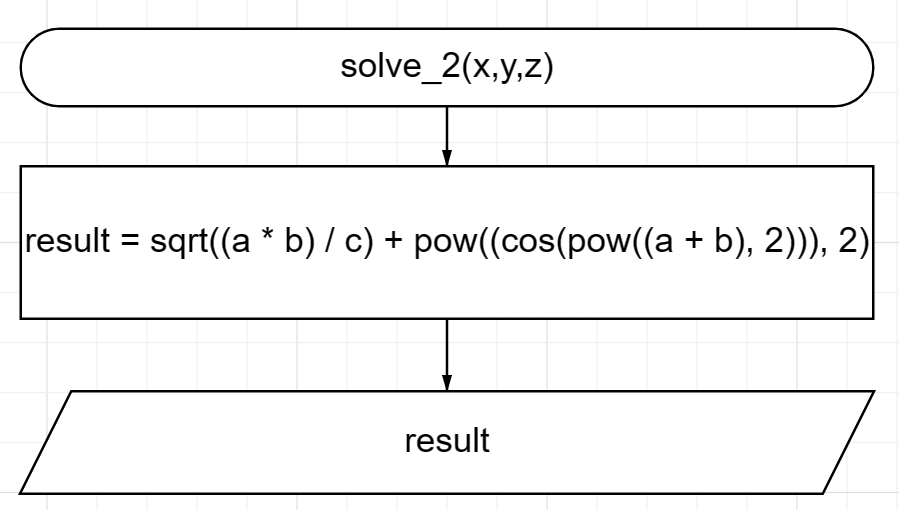


Рисунок  – блок-схема функции solve\_2

* 1. Код задания 1.1

#include <stdio.h>

#include <math.h>

/\*\*

\*@brief function gives solution for the equation "a"

\*@param a some constant

\*@param b some constant

\*@param c some constant

\*@return solution for the equation "a"

\*/

double solve\_1(double a, double b, double c);

/\*\*

\*@brief function gives solution for the equation "b"

\*@param a some constant

\*@param b some constant

\*@param c some constant

\*@return solution for the equation "b"

\*/

double solve\_2(double a, double b, double c);

//@brief entry point into the program

int main()

{

double a = 0.5, b = 0.05, c = 0.7;

printf\_s("%lf\n%lf\n", solve\_1(a, b, c), solve\_2(a, b, c));

return 0;

}

double solve\_1(double a, double b, double c)

{

double result = (pow(a, 2) \* (a + 1)) / (b - pow(sin(a + c), 2));

return result;

}

double solve\_2(double a, double b, double c)

{

double result = sqrt((a \* b) / c) + pow((cos(pow((a + b), 2))), 2);

return result;

}

* 1. Решение различных тестовых примеров на C

Ниже представлен вывод программы 1.1 в C (Рисунок 4)



Рисунок  – Вывод программы 1.1

* 1. Решение различных тестовых примеров в Excel

Ниже представлено решение задания 1.1 в Excel (Рисунок 5)

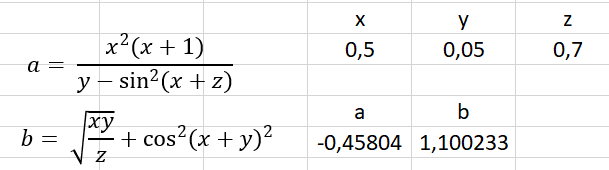


Рисунок  – Решение задания 1.1 в Excel

* 1. Подтверждение approve 1.1

Ниже предоставлено доказательство того, что задание 1.1 было принято (Рисунок 6)

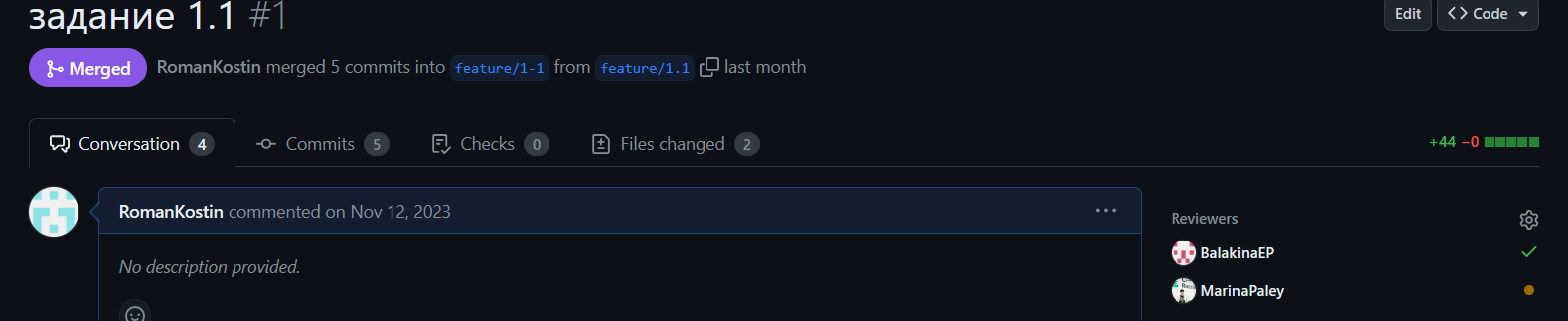


Рисунок  – Approve задачи 1.1

1. Решение задачи 1.2
   1. Формулировка задачи

Создать консольное приложение для решения задачи, представленной в таблице. Данные для решения вводит пользователь. Помните, что ввод необходимо проверять на правильность (только числа). Вывести результат вычислений на экран. При необходимости дополнить свой отчёт поясняющими формулами, помогающими решить задачу. Дополнить свой отчёт блок-схемой алгоритма

Таблица  – Условие задачи 1.2

|  |  |
| --- | --- |
| Вариант | Задача |
| 7 | Даны две стороны треугольника и угол между ними. Определить третью сторону, площадь треугольника и радиус описанной окружности. |

* 1. Блок-схема алгоритма

Ниже представлена блок-схема алгоритма (Рисунок 7, Рисунок 8, Рисунок 9, Рисунок 10, Рисунок 11, Рисунок 12)

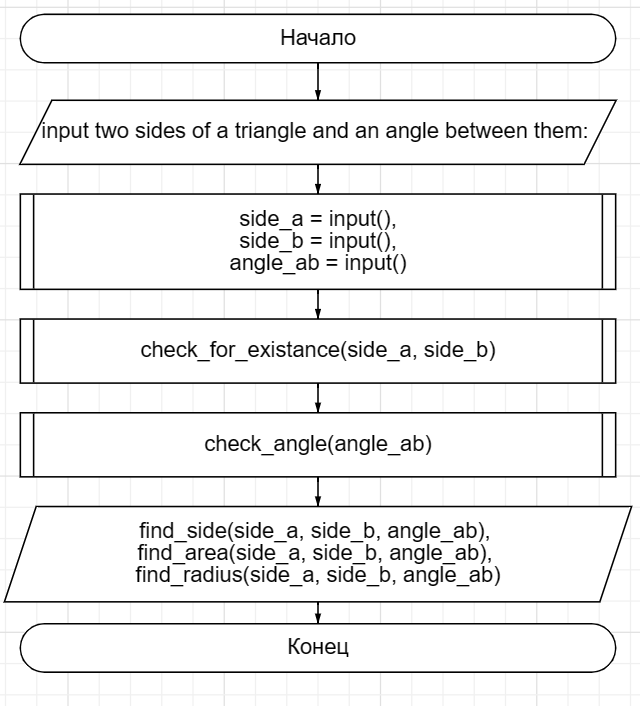


Рисунок  – блок-схема функции main()

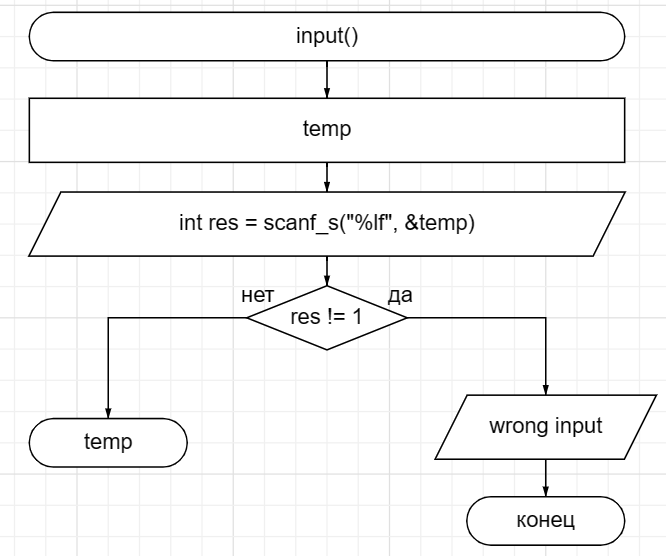


Рисунок  – блок-схема функции input()

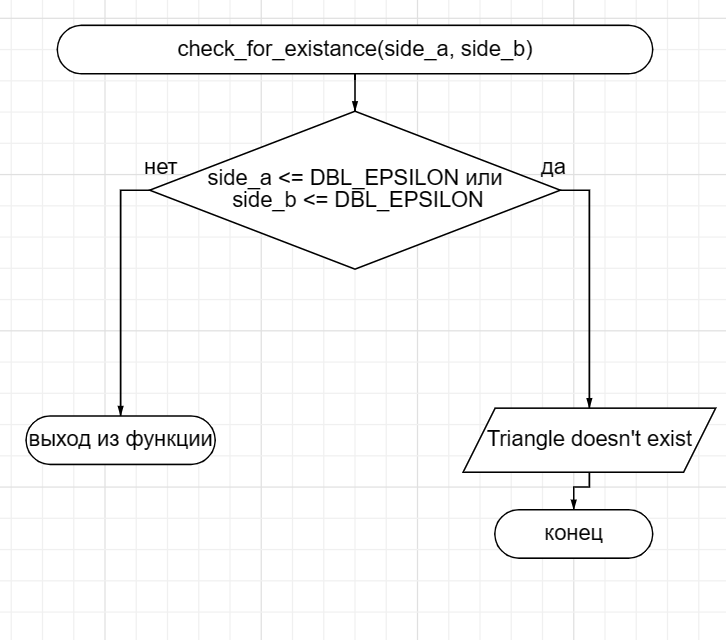


Рисунок 9 – блок-схема функции check\_for\_existance()

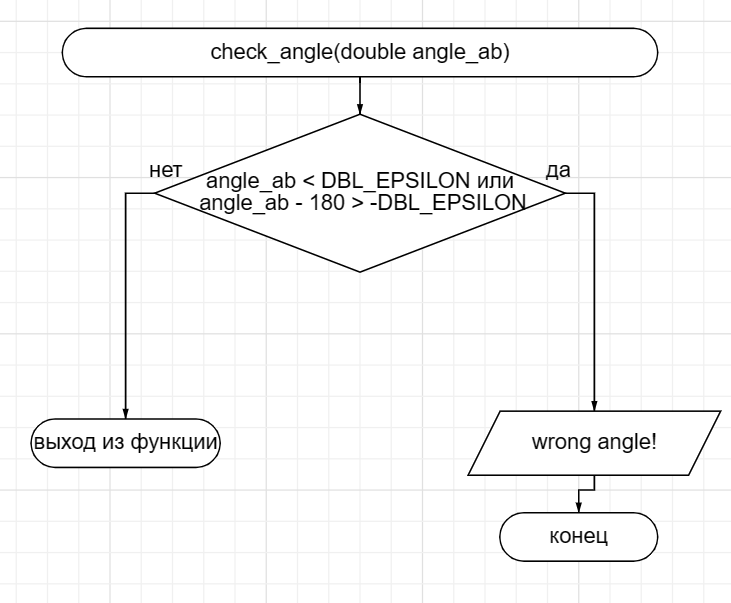


Рисунок  – блок-схема функции check\_angle()

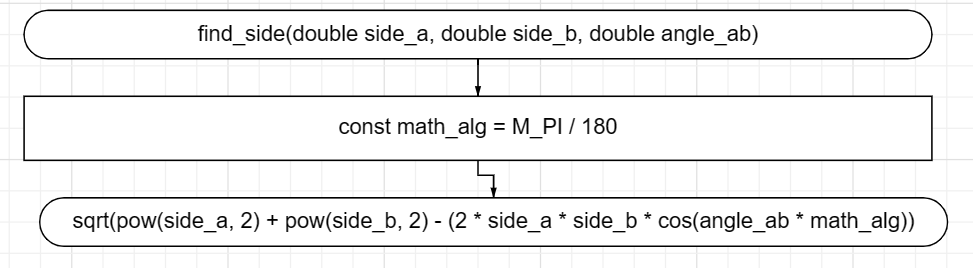


Рисунок  – блок-схема функции find\_side()

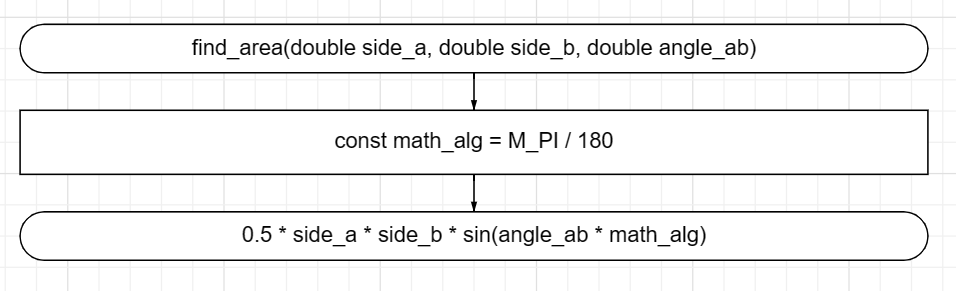


Рисунок  – блок-схема функции find\_area()

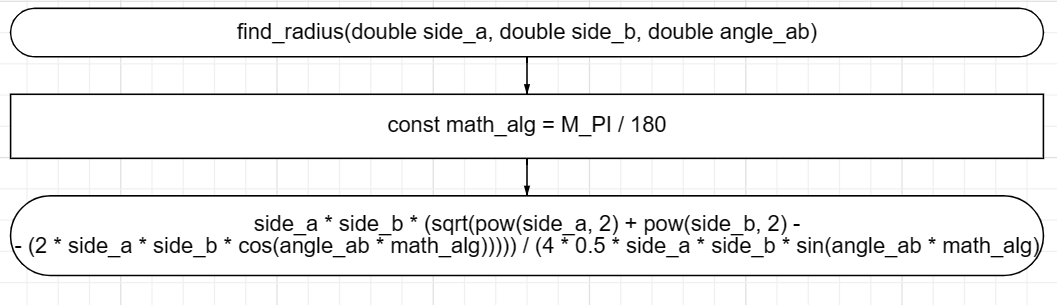


Рисунок  – блок-схема функции find\_radius()

* 1. Поясняющие формулы

S = – где:

S – площадь треугольника

a, b – стороны треугольника

angle – угол между сторонами a и b

= + - 2ab\*cos(angle) – где:

c – третья сторона треугольника

a, b – первая и вторая сторона треугольника

angle – угол между a и b

R = – где:

R – радиус описанной окружности

a, b, c – стороны треугольника

S – площадь треугольника

* 1. Код для задания 1.2

#include <stdio.h>

#include <stdlib.h>

#define \_USE\_MATH\_DEFINES

#include <math.h>

#include <float.h>

#include <errno.h>

/\*\*

\* @brief function used to get value from user

\* @return temp - result of user's input

\*/

double input();

/\*\*

\* @brief function checks if triangle exist

\* @param side\_a length of the first side

\* @param side\_b length of the second side

\*/

void check\_for\_existance(double side\_a, double side\_b);

/\*\*

\* @brief formula for finding the third side of a triangle

\* @param side\_a length of the first side

\* @param side\_b length of the second side

\* @param angle\_ab angle between two sides of a triangle (in degrees)

\* @return length of the third side

\*/

double find\_side(double side\_a, double side\_b, double angle\_ab);

/\*\*

\* @brief formula for finding the area of a triangle

\* @param side\_a length of the first side

\* @param side\_b length of the second side

\* @param angle\_ab angle between two sides of a triangle (in degrees)

\* @return area of the triangle

\*/

double find\_area(double side\_a, double side\_b, double angle\_ab);

/\*\*

\* @brief formula for finding the circumscribed circle radius of a triangle

\* @param side\_a length of the first side

\* @param side\_b length of the second side

\* @param angle\_ab angle between two sides of a triangle (in degrees)

\* @return circumscribed circle radius

\*/

double find\_radius(double side\_a, double side\_b, double angle\_ab);

/\*\*

\* @brief function checks if angle is in range of 0 and 180 degrees

\* @param angle\_ab - angle between two sides of a triangle

\*/

void check\_angle(double angle\_ab);

/\*\*

\* @brief point of entry into the programm

\* @return 0 if programm works correctly

\*/

int main()

{

printf\_s("%s", "input two sides of a triangle and an angle between them: ");

double side\_a = input(),

side\_b = input(),

angle\_ab = input();

check\_for\_existance(side\_a, side\_b);

check\_angle(angle\_ab);

printf\_s("%lf\n%lf\n%lf\n", find\_side(side\_a, side\_b, angle\_ab), find\_area(side\_a, side\_b, angle\_ab), find\_radius(side\_a, side\_b, angle\_ab));

return 0;

}

double find\_side(double side\_a, double side\_b, double angle\_ab)

{

double const math\_alg = M\_PI / 180;

return sqrt(pow(side\_a, 2) + pow(side\_b, 2) - (2 \* side\_a \* side\_b \* cos(angle\_ab \* math\_alg)));

}

double find\_area(double side\_a, double side\_b, double angle\_ab)

{

double const math\_alg = M\_PI / 180;

return 0.5 \* side\_a \* side\_b \* sin(angle\_ab \* math\_alg);

}

double find\_radius(double side\_a, double side\_b, double angle\_ab)

{

double const math\_alg = M\_PI / 180;

return (side\_a \* side\_b \* (sqrt(pow(side\_a, 2) + pow(side\_b, 2) - (2 \* side\_a \* side\_b \* cos(angle\_ab \* math\_alg))))) / (4 \* 0.5 \* side\_a \* side\_b \* sin(angle\_ab \* math\_alg));

}

double input()

{

double temp;

int res = scanf\_s("%lf", &temp);

if (res != 1)

{

errno = EIO;

perror("wrong input");

abort();

}

return temp;

}

void check\_for\_existance(double side\_a, double side\_b)

{

if (side\_a <= DBL\_EPSILON || side\_b <= DBL\_EPSILON)

{

printf\_s("Triangle doesn't exist");

abort();

}

}

void check\_angle(double angle\_ab)

{

if (angle\_ab < DBL\_EPSILON || angle\_ab - 180 > -DBL\_EPSILON)

{

errno = EIO;

perror("wrong angle!");

abort();

}

}

* 1. Решение различных тестовых примеров на C

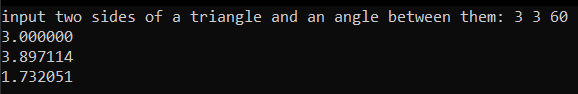


Рисунок  – вывод программы 1.2 на C при корректном вводе значений

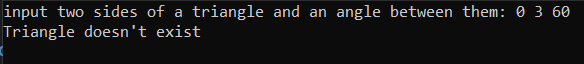


Рисунок  – вывод программы 1.2 на C если введена сторона неправильно

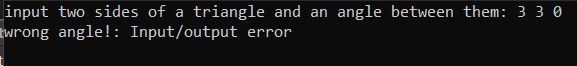


Рисунок  – вывод программы 1.2 на C если введен неправильно угол

* 1. Решение различных тестовых примеров в Excel

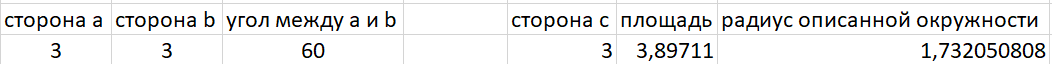


Рисунок  – Решение тестового примера 1.2 в Excel

* 1. Подтверждение Approve 1.2

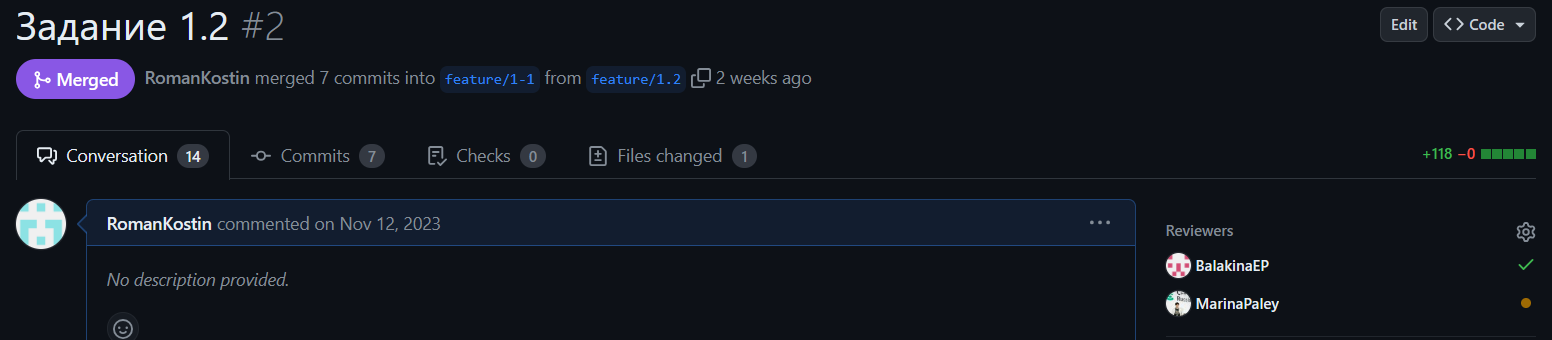


Рисунок  -approve задачи 1.2

1. решение задания 1.3
   1. Формулировка задачи

Создать консольное приложение для решения задачи, представленной в таблице. Данные для решения вводит пользователь. Помните, что ввод необходимо проверять на правильность (только числа). Вывести результат вычислений на экран. При необходимости дополнить свой отчёт поясняющими формулами, помогающими решить задачу. Дополнить свой отчёт блок-схемой алгоритма.

Таблица  – Условия задачи 1.3

|  |  |
| --- | --- |
| Вариант | Задача |
| 7 | Длина алюминиевого провода l метров, площадь его поперечного сечения S мм. Чему равно сопротивление провода? |

* 1. Блок-схема алгоритма

Блок-схема алгоритма представлена ниже (Рисунок 19, Рисунок 20, Рисунок 21, Рисунок 22)

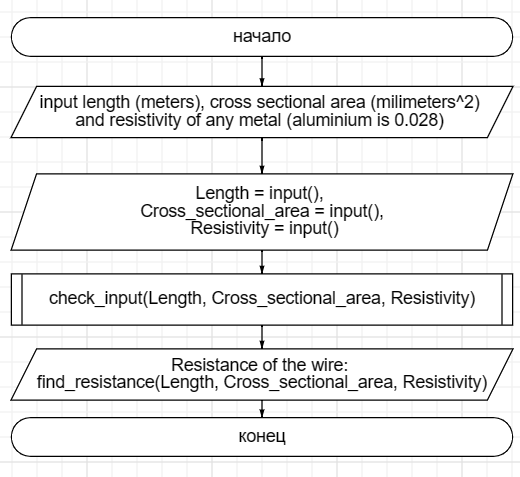


Рисунок  – блок-схема функции main()

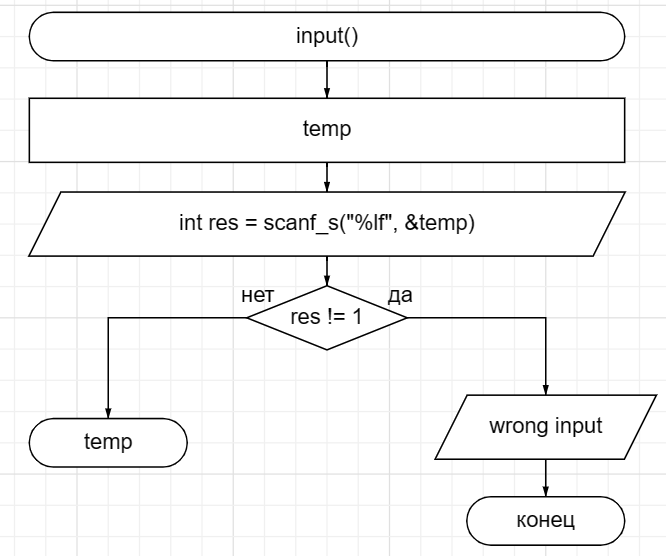


Рисунок  – блок-схема функции input()

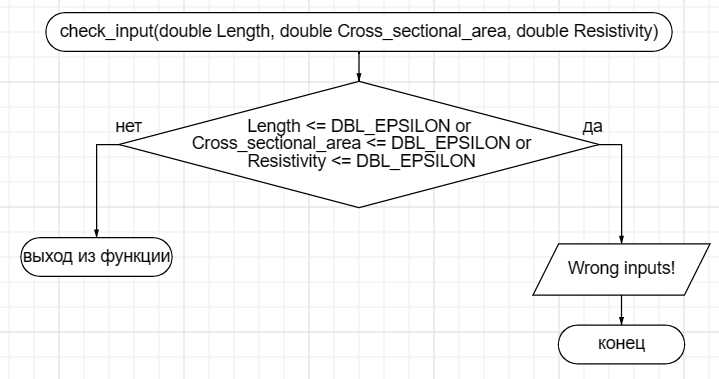


Рисунок  – блок-схема функции check\_input()

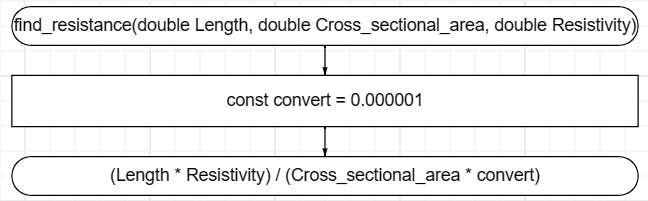


Рисунок  – блок-схема функции find\_resistance()

* 1. Поясняющие формулы

R = – где:

R – сопротивление проводника

p – удельное сопротивление проводника

l – длина проводника

S – площадь сечения провода

* 1. Код для задания

#include <stdio.h>

#include <float.h>

#include <errno.h>

/\*\*

\* @brief function used to get value from user

\* @return result - result of user's input

\*/

double input();

/\*\*

\* @brief function checks if inputs are right

\* @param Length - length of the wire

\* @param Cross\_sectional\_area - cross sectional area of the wire

\* @param Resistivity - resistivity of the metal which is used in the wire

\*/

void check\_input(double Length, double Cross\_sectional\_area, double Resistivity);

/\*\*

\*@brief function for finding resistivity of the wire (length \* resistivity of the metal) divided by cross sectional area, which is turned in meters

\*@param Length - length of the wire

\*@param Cross\_sectional\_area - cross sectional area of the wire

\*@param Resistivity - resistivity of the metal which is used in the wire

\*@return resistivity of the wire

\*/

double find\_resistance(double Length, double Cross\_sectional\_area, double Resistivity);

/\*\*

\* @brief entry point into the program

\* @return 0 if program works correctly

\*/

int main()

{

printf\_s("input length (meters), cross sectional area (milimeters^2) and resistivity of any metal (aluminium is 0.028) ");

double Length = input(),

Cross\_sectional\_area = input(),

Resistivity = input();

check\_input(Length, Cross\_sectional\_area, Resistivity);

printf\_s("Resistance of the wire %lf", find\_resistance(Length, Cross\_sectional\_area, Resistivity));

return 0;

}

double input()

{

double temp;

int res = scanf\_s("%lf", &temp);

if (res != 1)

{

errno = EIO;

perror("wrong input");

abort();

}

return temp;

}

void check\_input(double Length, double Cross\_sectional\_area, double Resistivity)

{

if (Length <= DBL\_EPSILON || Cross\_sectional\_area <= DBL\_EPSILON || Resistivity <= DBL\_EPSILON)

{

printf\_s("Wrong inputs!");

abort();

}

}

double find\_resistance(double Length, double Cross\_sectional\_area, double Resistivity)

{

double const convert = 0.000001;

return (Length \* Resistivity) / (Cross\_sectional\_area \* convert);

}

* 1. Решение различных тестовых примеров на C



Рисунок  – вывод программы 1.3 на C при корректном вводе



Рисунок  – вывод программы 1.3 на C при некорректном вводе длины провода

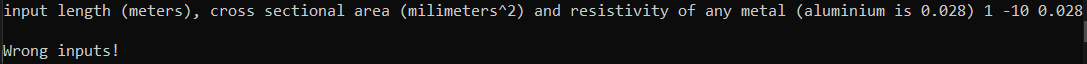


Рисунок  – вывод программы 1.3 на C при некорректном вводе площади



Рисунок  – вывод программы 1.3 на C при некорректном вводе удельного сопротивления проводника

* 1. Решение различных тестовых примеров в Excel



Рисунок  – решение задания 1.3 в Excel

* 1. Подтверждение Approve

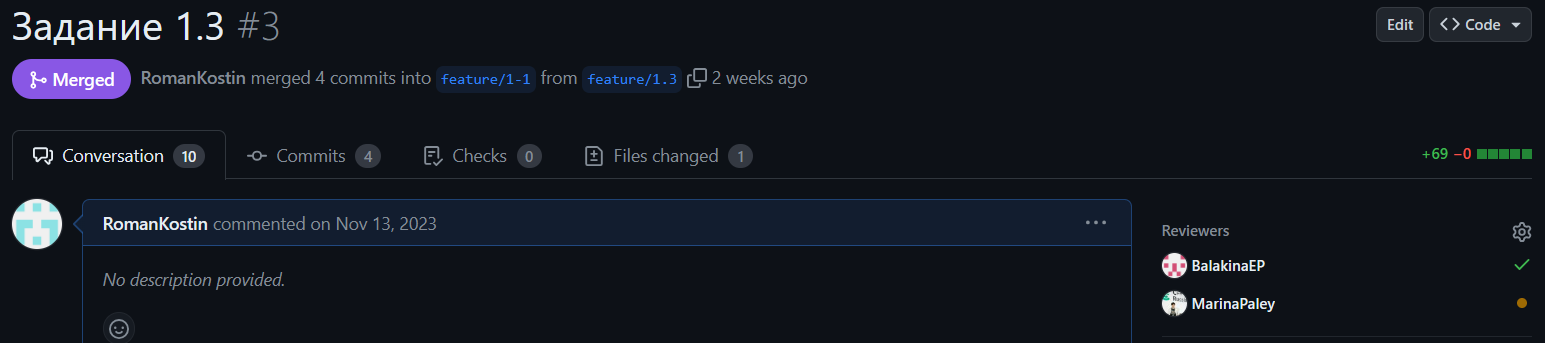


Рисунок  – Approve задания 1.3

Оглавление

[1 Решение задачи 1.1 2](#_Toc155713371)

[1.1 Формулировка задачи 2](#_Toc155713372)

[1.2 Блок–схема алгоритма 3](#_Toc155713373)

[1.3 Код задания 1.1 4](#_Toc155713374)

[1.4 Решение различных тестовых примеров на C 5](#_Toc155713375)

[1.5 Решение различных тестовых примеров в Excel 5](#_Toc155713376)

[1.6 Подтверждение approve 1.1 5](#_Toc155713377)

[2 Решение задачи 1.2 6](#_Toc155713378)

[2.1 Формулировка задачи 6](#_Toc155713379)

[2.2 Блок-схема алгоритма 7](#_Toc155713380)

[2.3 Поясняющие формулы 10](#_Toc155713381)

[2.4 Код для задания 1.2 10](#_Toc155713382)

[2.5 Решение различных тестовых примеров на C 12](#_Toc155713383)

[2.6 Решение различных тестовых примеров в Excel 13](#_Toc155713384)

[2.7 Подтверждение Approve 1.2 13](#_Toc155713385)

[3 решение задания 1.3 14](#_Toc155713386)

[3.1 Формулировка задачи 14](#_Toc155713387)

[3.2 Блок-схема алгоритма 15](#_Toc155713388)

[3.3 Поясняющие формулы 17](#_Toc155713389)

[3.4 Код для задания 17](#_Toc155713390)

[3.5 Решение различных тестовых примеров на C 19](#_Toc155713391)

[3.6 Решение различных тестовых примеров в Excel 19](#_Toc155713392)

[3.7 Подтверждение Approve 19](#_Toc155713393)