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 5

Выполнил: ст. гр. ТКИ-142

Гаспарян Артём Артурович

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

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

Москва 2024

**Оглавление**

[1 ЗАДАНИЕ 1.1 3](#_Toc183878089)

[1.1) Формулировка задания 3](#_Toc183878090)

[1.2) Блок-схема алгоритма 4](#_Toc183878091)

[1.3) Текст программы на языке С 4](#_Toc183878092)

[1.4) Результат выполнения программы 6](#_Toc183878093)

[1.5) Выполнение тестовых примеров 7](#_Toc183878094)

[1.6) Отметка о выполнение задания 7](#_Toc183878095)

[2 ЗАДАНИЕ 1.2 9](#_Toc183878096)

[2.1) Формулировка задания 9](#_Toc183878097)

[2.2) Блок-схема алгоритма 10](#_Toc183878098)

[2.3) Текст программы на языке С 12](#_Toc183878099)

[2.4) Результат выполнения программы 13](#_Toc183878100)

[2.5) Выполнение тестовых примеров 14](#_Toc183878101)

[2.6) Отметка о выполнение задания 15](#_Toc183878102)

[3 ЗАДАНИЕ 1.3 16](#_Toc183878103)

[3.1) Формулировка задания 16](#_Toc183878104)

[3.2) Блок-схема алгоритма 17](#_Toc183878105)

[3.3) Текст программы на языке С 19](#_Toc183878106)

[3.4) Результат выполнения программы 20](#_Toc183878107)

[3.5) Выполнение тестовых примеров 21](#_Toc183878108)

[3.6) Отметка о выполнении задания 22](#_Toc183878109)

1. ЗАДАНИЕ 1.1
   1. Формулировка задания

Создать консольное приложение, вычисляющее значения переменных по представленным в таблице формулам (Таблица 1). Расчёт примера осуществить по заданным константам. Вывести на экран значения исходных данных, а также результат вычислений. Дополнить свой отчёт блок-схемой алгоритма.

Таблица 1 – Исходные данные

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Формулы** | **Константы** |
| 5 |  | x = -2.9  y = 15.5  z = 1.5 |

* 1. Блок-схема алгоритма

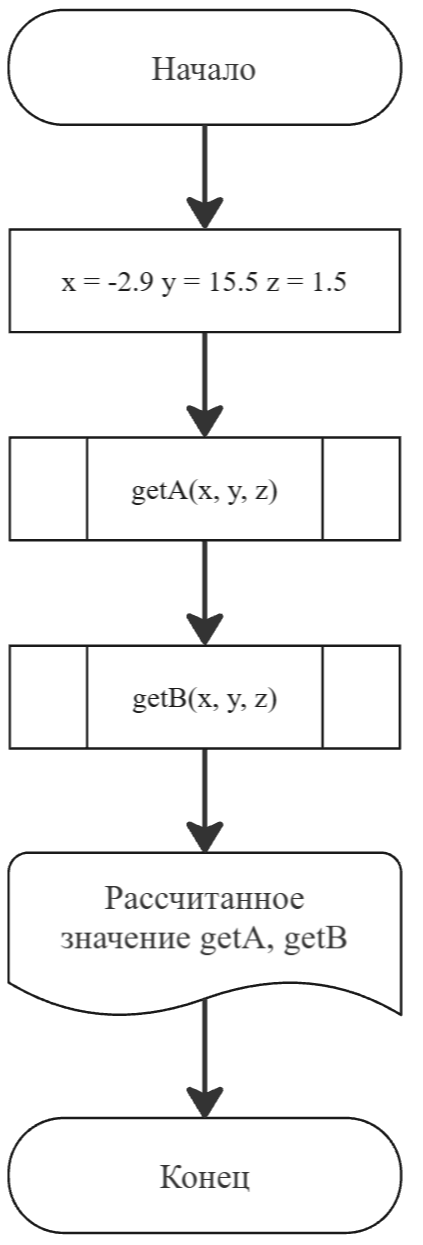
Блок-схема основного алгоритма представлена ниже (Рисунок 1). Блок-схемы функции расчета значений a и b представлены ниже (Рисунок 2). 

Рисунок 1 – Блок-схема основного алгоритма

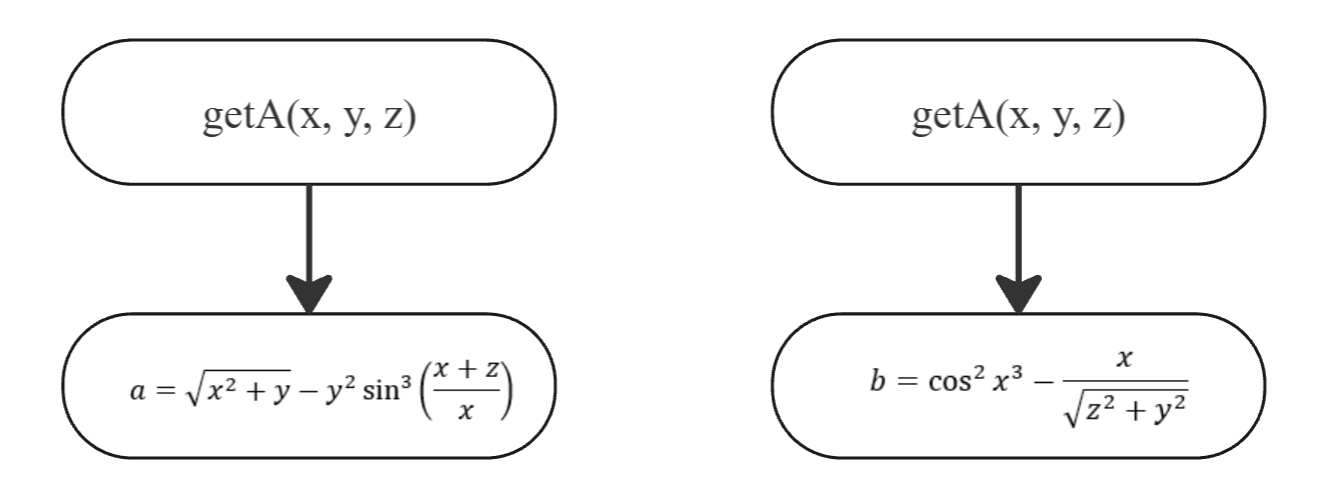


Рисунок 2 – Блок-схема функций расчета

* 1. Текст программы на языке С

#include <stdio.h>

#include <math.h>

/\* @brief рассчитывает выражение A по формуле

\* @param x значение константы x

\* @param y значение константы y

\* @param z значение константы z

\* @return рассчитанное значение

\*/

double getA(const double x, const double y, const double z);

/\* @brief рассчитывает выражение B по формуле

\* @param x значение константы x

\* @param y значение константы y

\* @param z значение константы z

\* @return рассчитанное значение.

\*/

double getB(const double x, const double y, const double z);

/\* @brief Точка входа в программу

\* @return 0 в случае успеха

\*/

int main()

{

const double x = -2.9;

const double y = 15.5;

const double z = 1.5;

printf("%lf\n%lf", getA(x, y, z), getB(x, y, z));

return 0;

}

double getA(const double x, const double y, const double z)

{

return sqrt(pow(x, 2) + y) - pow(y, 2)\*pow(sin((x + z) / x), 3);

}

double getB(const double x, const double y, const double z)

{

return pow(cos(pow(x, 3)), 2) - (x / (sqrt(pow(z, 2) + pow(y, 2))));

}

* 1. Результат выполнения программы

Результат выполнение программы представлен на рисунке ниже (Рисунок 3)

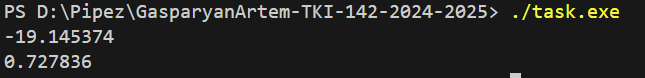


Рисунок 3 – Результат выполнения программы

* 1. Выполнение тестовых примеров

В программе MS Excel выполнены тестовые примеры. Результат их выполнения представлен ниже (Рисунок 4, Рисунок 5)

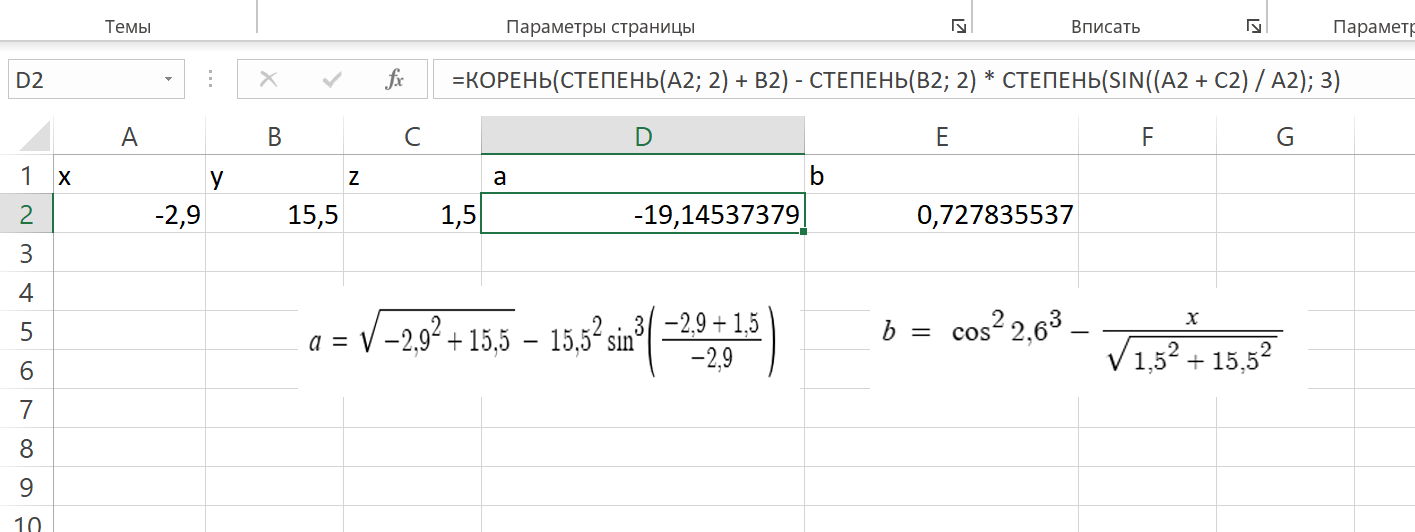


Рисунок 4 – Результат расчета переменной a

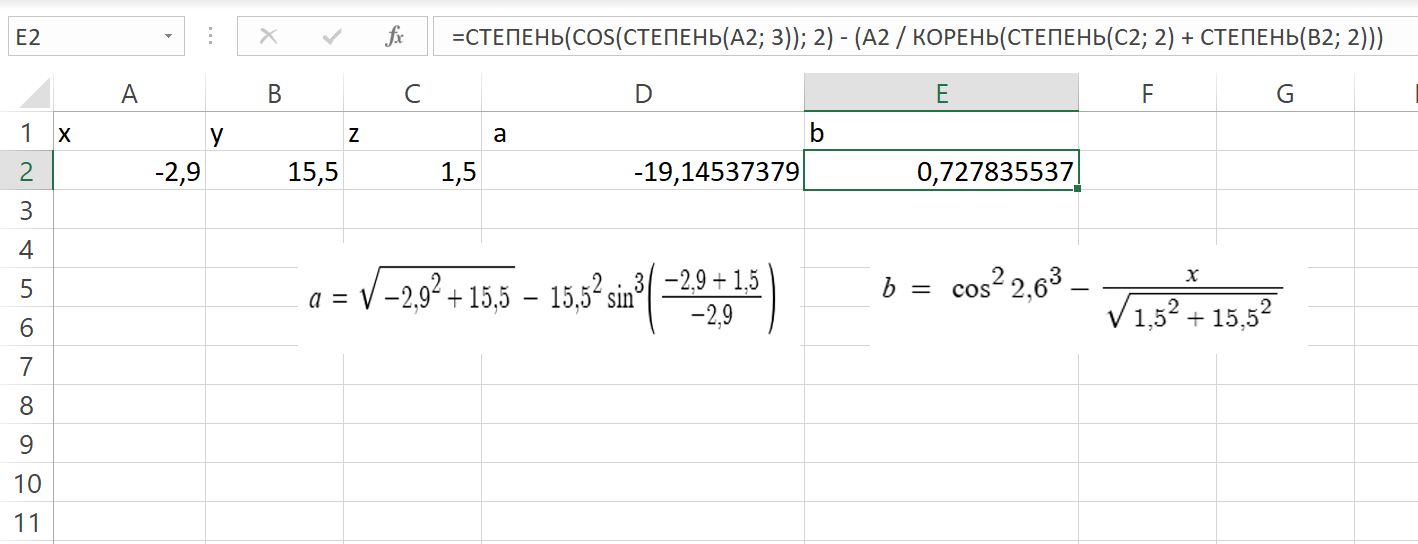


Рисунок 5 – Результат расчета переменной b

* 1. Отметка о выполнение задания

Отметка о выполнение задания представлена ниже (Рисунок 6)

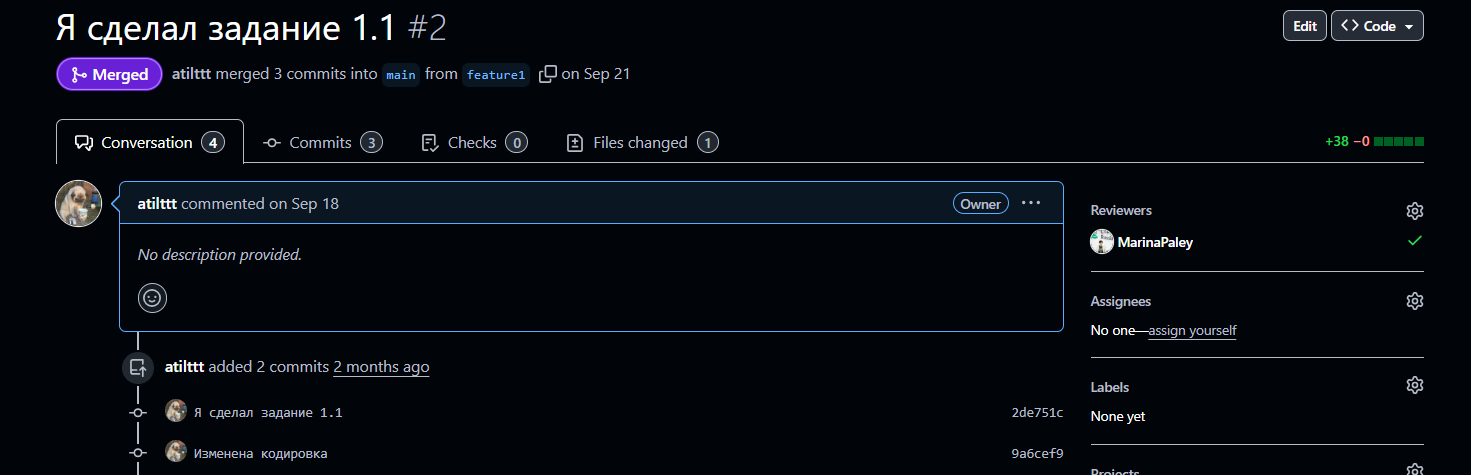


Рисунок 6 – Отметка о выполнении задания

1. ЗАДАНИЕ 1.2
   1. Формулировка задания

Создать консольное приложение для решения задачи, представленной в таблице (Таблица 2). 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 2 – Исходные данные

|  |  |
| --- | --- |
| **Вариант** | **Задача** |
| 5 | Перевести значение веса, выраженное в граммах, в унции (1 унция = 28.3 г) |

* 1. Блок-схема алгоритма

Блок-схема основного алгоритма представлена ниже (Рисунок 7). Блок-схема функций расчета представлена ниже (Рисунок 8, Рисунок 9).

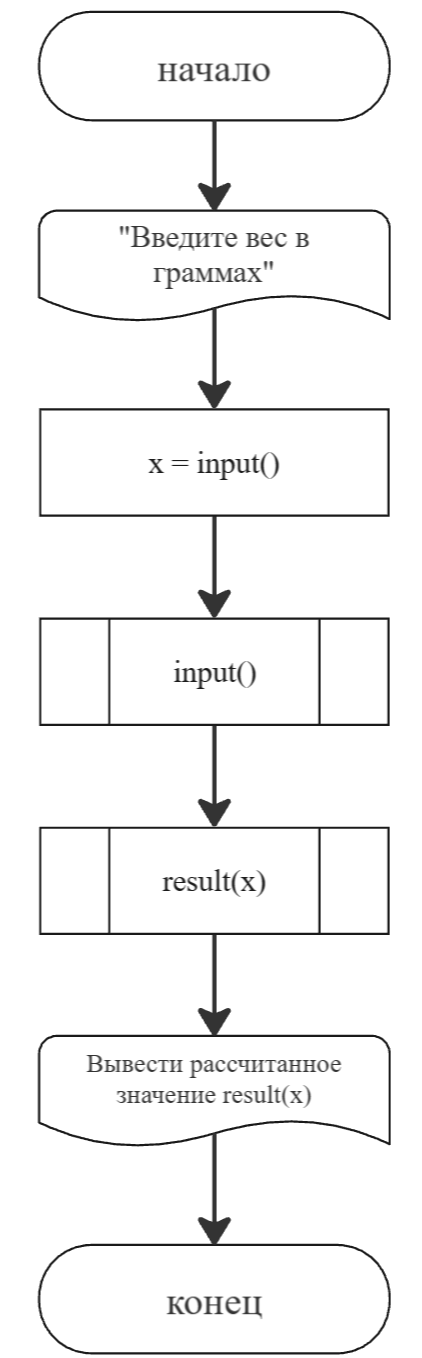


Рисунок 7 – Блок-схема основного алгоритма

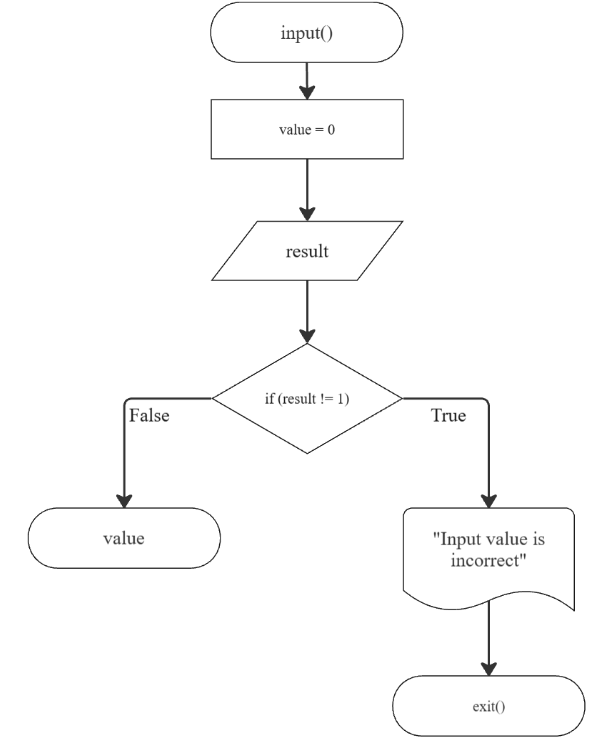


Рисунок 8 – Блок-схема функции input()

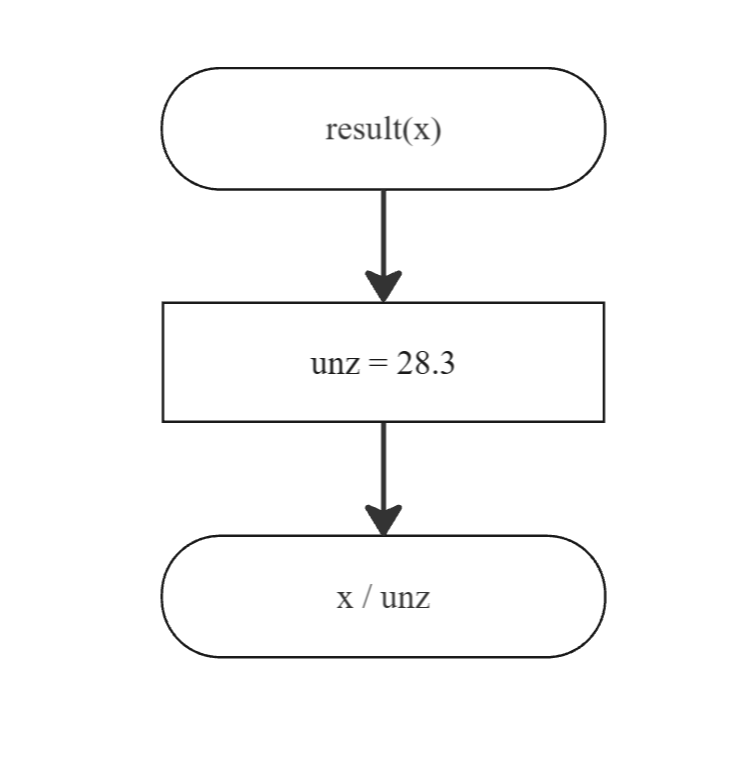


Рисунок 9 – Блок-схема функции result()

* 1. Текст программы на языке С

#include <stdio.h>

#include <errno.h>

#include <stdlib.h>

/\*\*

\* @brief функия, получающая значение из потока ввода

\* @return возвращает оишбку, в случае если значение некорректное.

\*/

int input(void);

/\*\*

\* @brief перевод значения веса в граммах в унции

\* @param inch значение 1 унция

\* @param x вес в граммах

\* @return возвращает рассчитаное значение

\*/

double result(double x);

/\*\*

\* @brief вход в программу

\* @return 0 в случае успеха

\*/

int main(void)

{

printf("Введите вес в граммах:");

double x = input();

printf("%lf", result(x));

return 0;

}

int input(void)

{

double value = 0.0;

int result = scanf("%lf", &value);

if (result != 1)

{

errno = EIO;

perror("Input value is discorrect\n");

exit(EXIT\_FAILURE);

}

return value;

}

double result(double x)

{

const double unz = 28.3;

return x / unz;

}

* 1. Результат выполнения программы

Результат выполнения программы представлен на рисунках ниже (Рисунок 10, Рисунок 11, Рисунок 12).

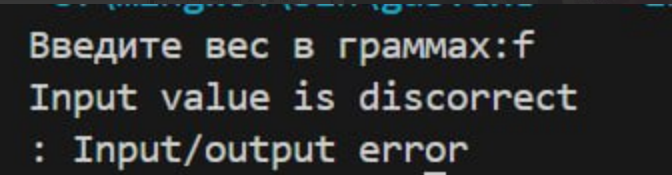


Рисунок 10 – проверка работы функции input()

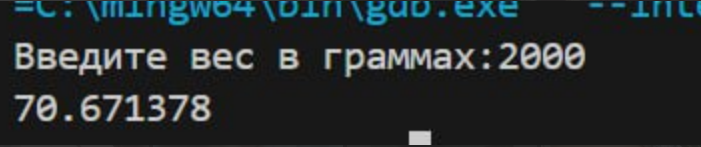


Рисунок 11 – первый результат выполнения программы

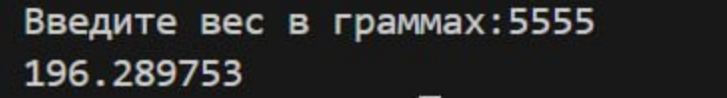


Рисунок 12 – второй результат выполнения программы

* 1. Выполнение тестовых примеров

В программе MS Excel были выполнены тестовые примеры. Результат их выполнения представлен ниже (Рисунок 13, Рисунок 14).

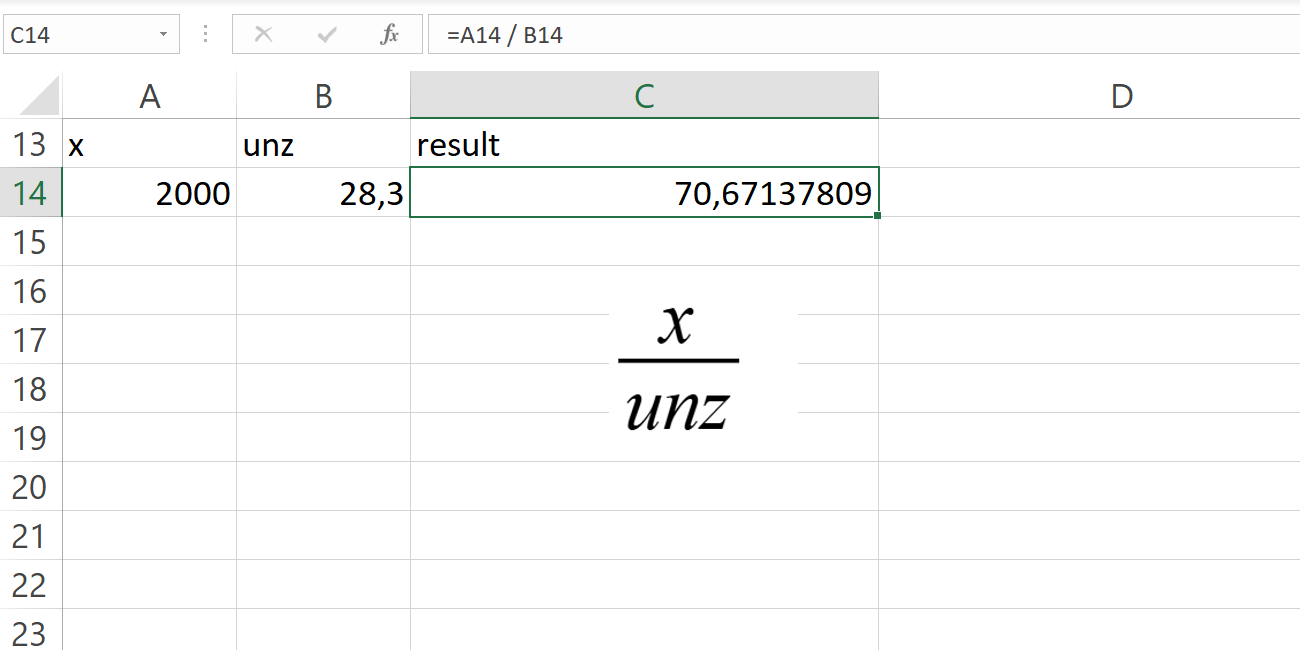


Рисунок 13 – выполнение первого тестового примера

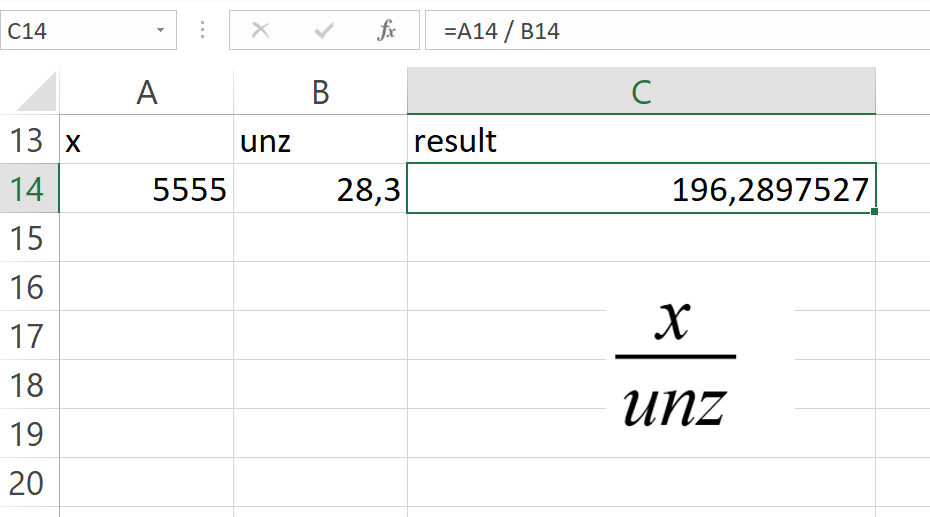


Рисунок 14 – Выполнение второго тестового примера

* 1. Отметка о выполнение задания

Отметка о выполнение задания в GitHub (Рисунок 15).

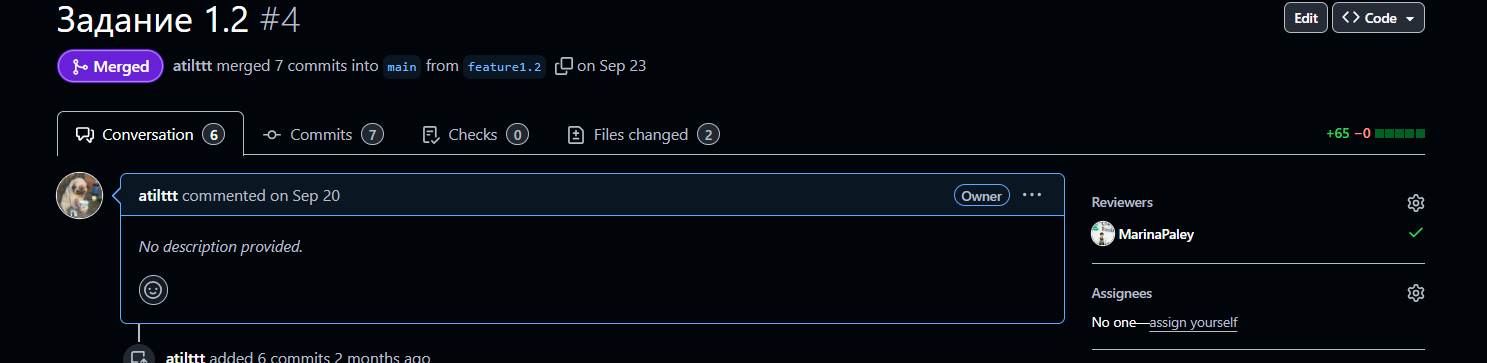


Рисунок 15 – отметка о выполнение задания

1. ЗАДАНИЕ 1.3
   1. Формулировка задания

Создать консольное приложение для решения задачи, представленной в таблице (Таблица 3). 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 3 – Исходные данные

|  |  |
| --- | --- |
| **Вариант** | **Задача** |
| 5 | Какую работу совершит электрический ток в электродвигателе вентилятора за 20 минут, если сила тока в цепи равна *I* А, а напряжение равно *U* В? |

* 1. Блок-схема алгоритма

Блок-схема основного алгоритма представлена ниже (). Блок-схемы функций ().

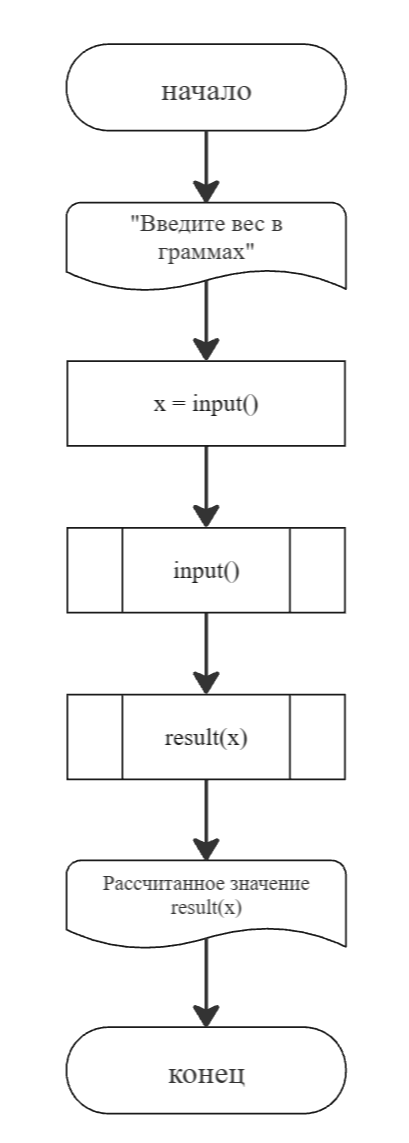


Рисунок 16 – блок-схема основного алгоритма

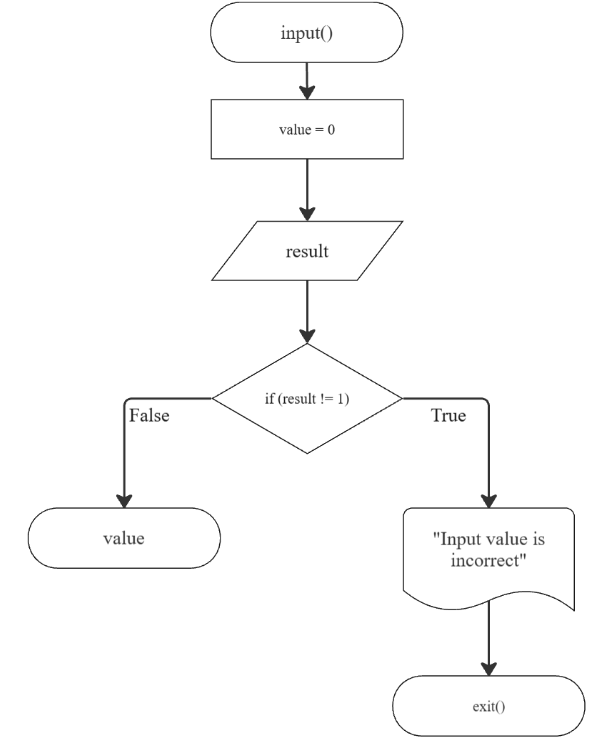


Рисунок 17 – блок-схема функции input()

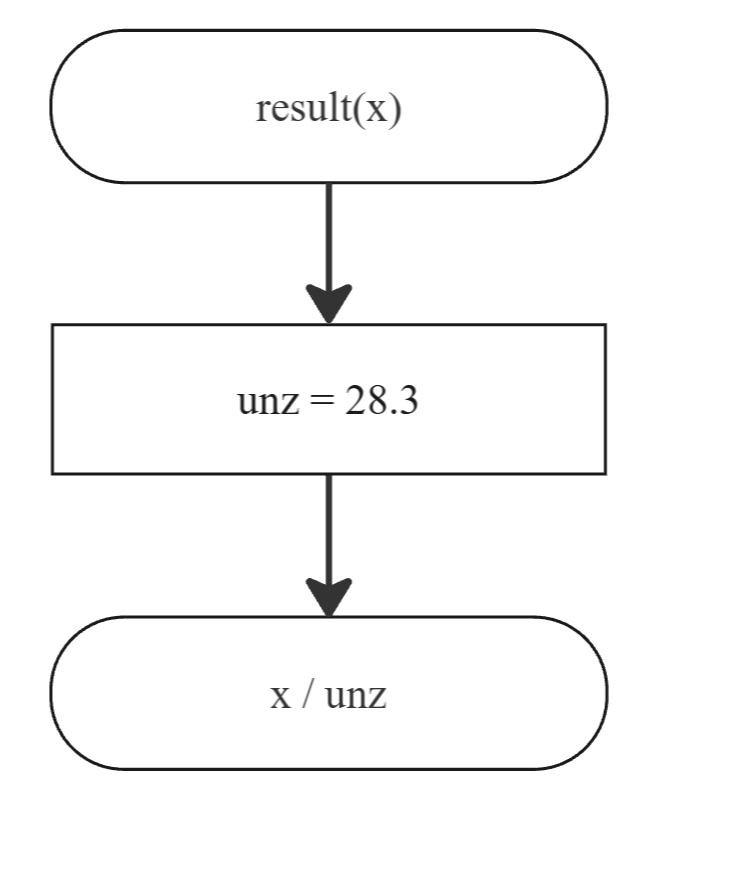


Рисунок 18 – блок-схема функции result()

* 1. Текст программы на языке С

#include <stdio.h>

#include <errno.h>

#include <stdlib.h>

/\*

\* @brief рассчитывает работу эл.тока по формуле

\* @param I - сила тока

\* @param U - напряжение

\* @param time - время переведенное в секунды

\* return рассчитанное значение

\*/

double getA(const double time, double I, double U);

/\*

\* @brief считывает вещественное число

\* @return возвращает вещественное число

\*/

double input();

/\*

\* @brief точка входу в прогамму

\* return 0 в случае успехa

\*/

int main()

{

const double time = 20.0;

puts("Введите значение силы тока");

double I = input();

puts("Введите значение напряжения");

double U = input();

printf("%lf", getA(time, I, U));

return 0;

}

double getA(const double time, double I, double U)

{

return U \* I \* (time \* 60);

}

double input()

{

double value = 0.0;

int result = scanf("%lf", &value);

if (result != 1)

{

errno = EIO;

perror("Invalid input!");

exit(EXIT\_FAILURE);

}

return value;

}

* 1. Результат выполнения программы

Результаты выполнения программы представлены ниже (Рисунок 19, Рисунок 20, Рисунок 21).

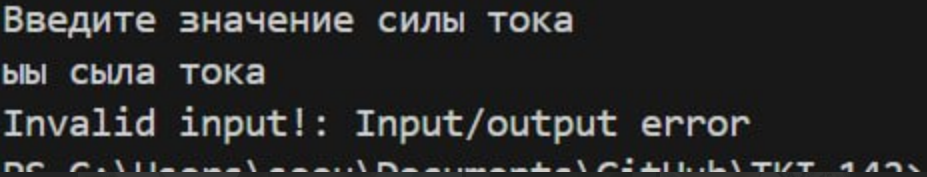


Рисунок 19 – проверка функции input()

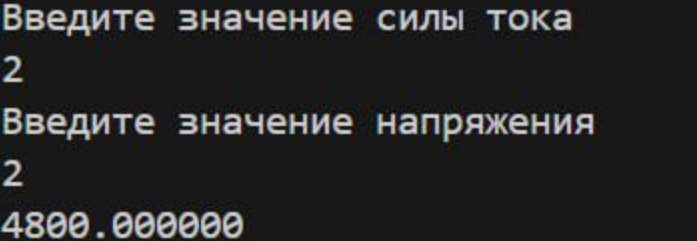


Рисунок 20 – первый результат выполнения программы

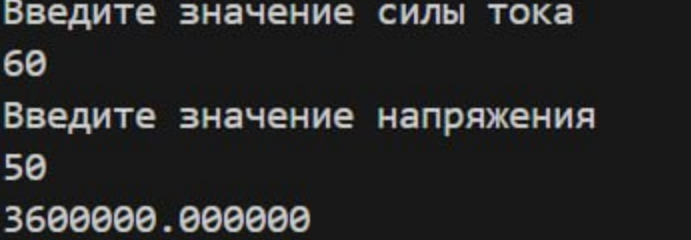


Рисунок 21 – второй результат выполнения программы

* 1. Выполнение тестовых примеров

В программе MS Excel были выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 22, Рисунок 23).

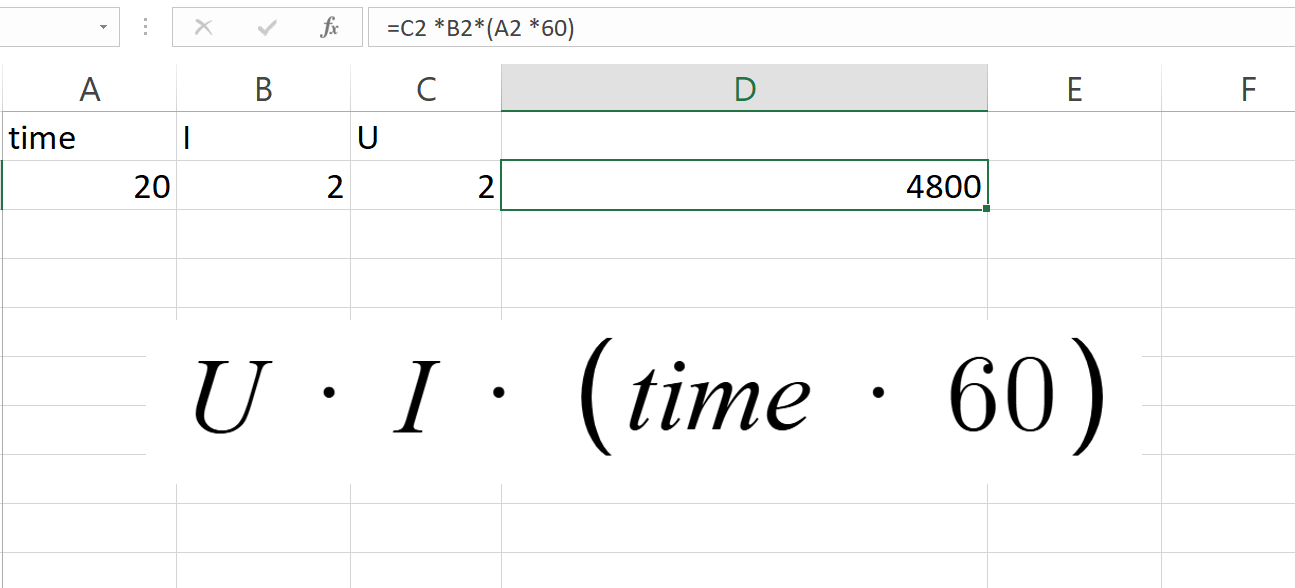


Рисунок 22 – результат выполнения первого тестового примера

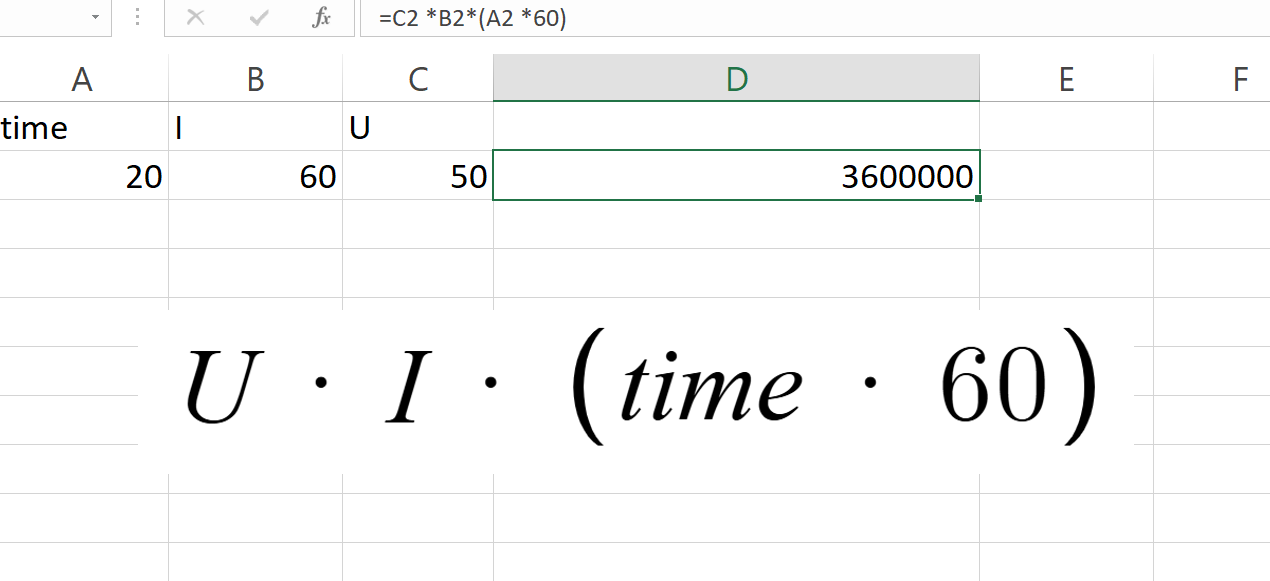


Рисунок 23 – результат выполнения второго тестового примера

* 1. Отметка о выполнении задания

Отметка о выполнение в GitHub (Рисунок 24).

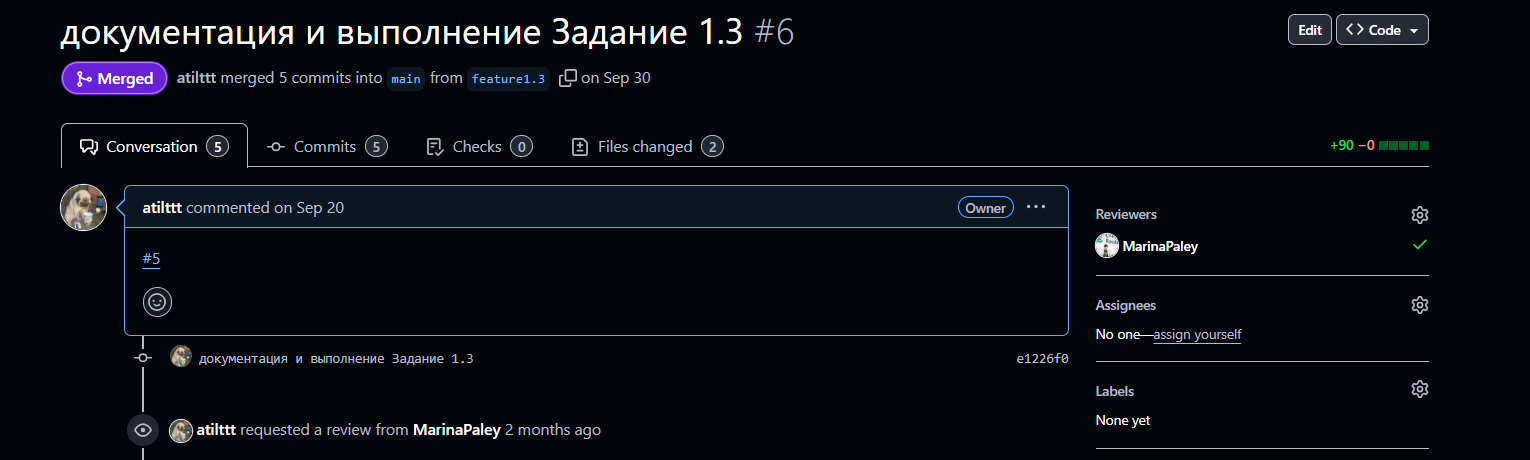


Рисунок 24 – отметка о выполнении