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-141

Буланый Сергей Евгеньевич

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

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

Москва 2023

Оглавление

[1 Решение задачи 1-1 3](#_Toc151483877)

[1.1 Формулировка задачи 1-1 3](#_Toc151483878)

[1.2 Блок-схема алгоритма задание 1–1 4](#_Toc151483879)

[1.3 Текст программы на языке С задание 1–1 5](#_Toc151483880)

[1.4 Результаты выполнения программы 1–1 6](#_Toc151483881)

[1.5 Выполнение тестовых примеров задание 1–1 7](#_Toc151483882)

[1.6 Отметка о выполнении задания в веб-хостинге системы контроля версий задание 1–1 8](#_Toc151483882)

[2 Решение задачи 1-2 9](#_Toc151483883)

[2.1 Формулировка задачи 1-2 9](#_Toc151483884)

[2.2 Блок-схема алгоритма задания 1-2 10](#_Toc151483885)

[2.3 Текст программы на языке С задание 1–2 12](#_Toc151483886)

[2.4 Результаты выполнения программы 1–2 13](#_Toc151483887)

[2.5 Выполнение тестовых примеров задание 1–2 14](#_Toc151483888)

[2.6 Отметка о выполнении задания в веб-хостинге системы контроля версий задание 1–2 16](#_Toc151483889)

[3 Решение задачи 1-3 17](#_Toc151483890)

[3.1 Формулировка задачи 1-3 1](#_Toc151483891)7

[3.2 Блок-схема алгоритма задания 1-3 18](#_Toc151483892)

[3.3 Текст программы на языке С задание 1–3 20](#_Toc151483893)

[3.4 Результаты выполнения программы 1–3 22](#_Toc151483894)

[3.5 Выполнение тестовых примеров задание 1–3 23](#_Toc151483895)

[3.6 Отметка о выполнении задания в веб-хостинге системы контроля версий задание 1–3 25](#_Toc151483896)

1. Решение задачи 1–1

**1.1 Формулировка** **задачи 1–1**

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 |

1.2 Блок-схема алгоритма задание 1–1

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

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рисунок 1 ­ Блок-схема основного алгоритма

Изображение выглядит как текст, Шрифт, снимок экрана, диаграмма

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Рисунок 2 – Блок-схема используемых функций

1.3 Текст программы на языке С задание 1–1

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

/\*

\* @brief Возвращает результат функции a

\* @param Параметр "x" для функуии a

\* @param Параметр "y" для функуии a

\* @param Параметр "z" для функуии a

\*\*/

double aFunc(double x, double y, double z);

/\*

\* @brief Возвращает результат функции a

\* @param Параметр "x" для функуии a

\* @param Параметр "y" для функуии a

\* @param Параметр "z" для функуии a

\*\*/

double bFunc(double x, double y, double z);

/\*

\* @brief Точка входа в программу

\* @return Возвращает 0, если программа работает корректно, иначе не 0

\*\*/

int main()

{

const double x = -2.9;

const double y = 15.5;

const double z = 1.5;

double a = aFunc(x, y, z);

double b = bFunc(x, y, z);

printf("a=%f\n", a);

printf("b=%f\n", b);

return 0;

}

double aFunc(double x, double y, double z)

{

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

}

double bFunc(double x, double y, double z)

{

double denominator = sqrt((pow(z, 2)) + (pow(y, 2)));

return (pow(cos(pow(x, 3)), 2)) - (x / denominator);

}

* 1. Результаты выполнения программы 1–1

Результаты выполнения программы в C представлена ниже (Рисунок 4).

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 4 – Результаты выполнения программы

* 1. Выполнение тестовых примеров задание 1–1

В программе Photomath выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 5, Рисунок 6).

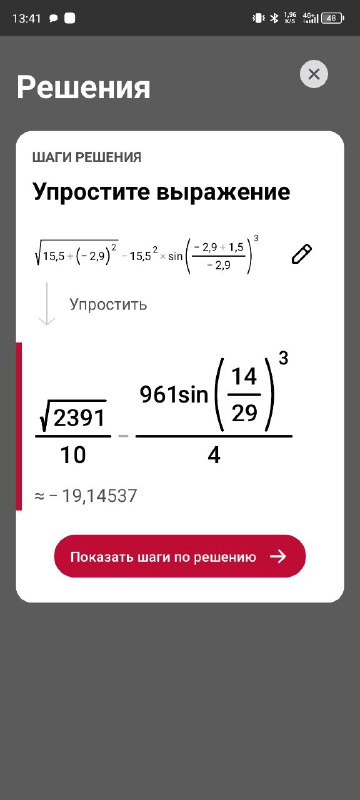


Рисунок 5 – Результат расчета переменной a

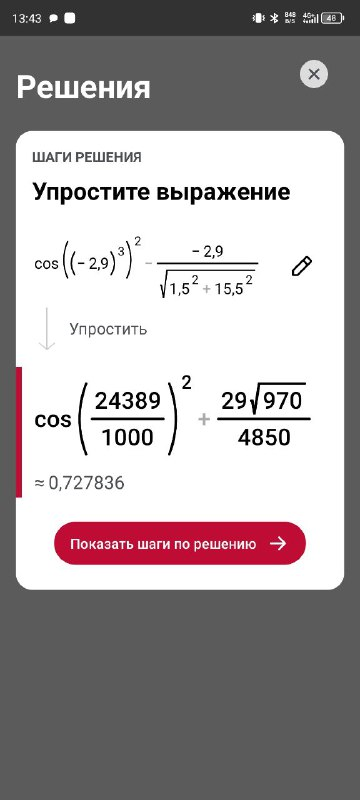


Рисунок 6 – Результат расчета переменной b

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

Ниже представлено доказательство того, что задание 1–1 было принято. (Рисунок 7)

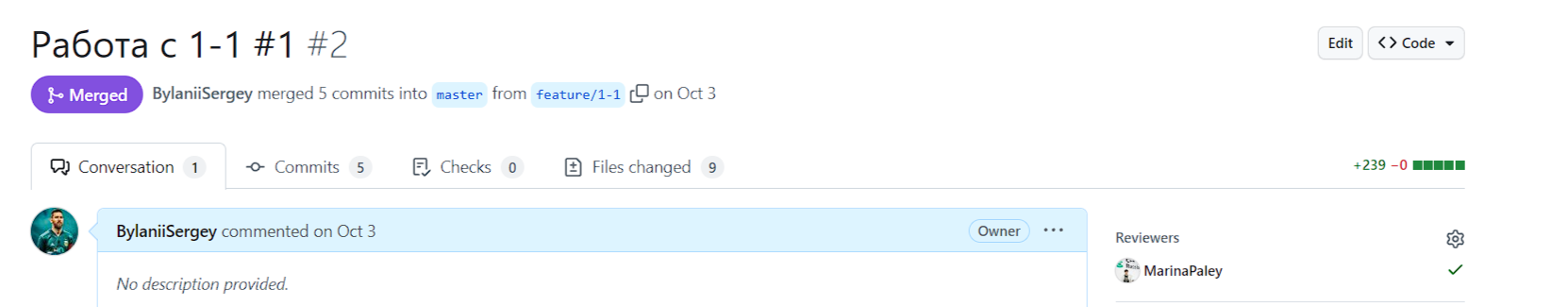


Рисунок 7 – Approve задачи 1–1

1. Решение задачи 1–2

**2.1 Формулировка** **задачи 1–2**

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 1 – Исходные данные

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

2.2 Блок-схема алгоритма задание 1–2

Блок-схема основного алгоритма представлена ниже (Рисунок 1). Блок-схемы функций расчета значений ounces и проверка представлены ниже (Рисунок 2 и Рисунок 3).

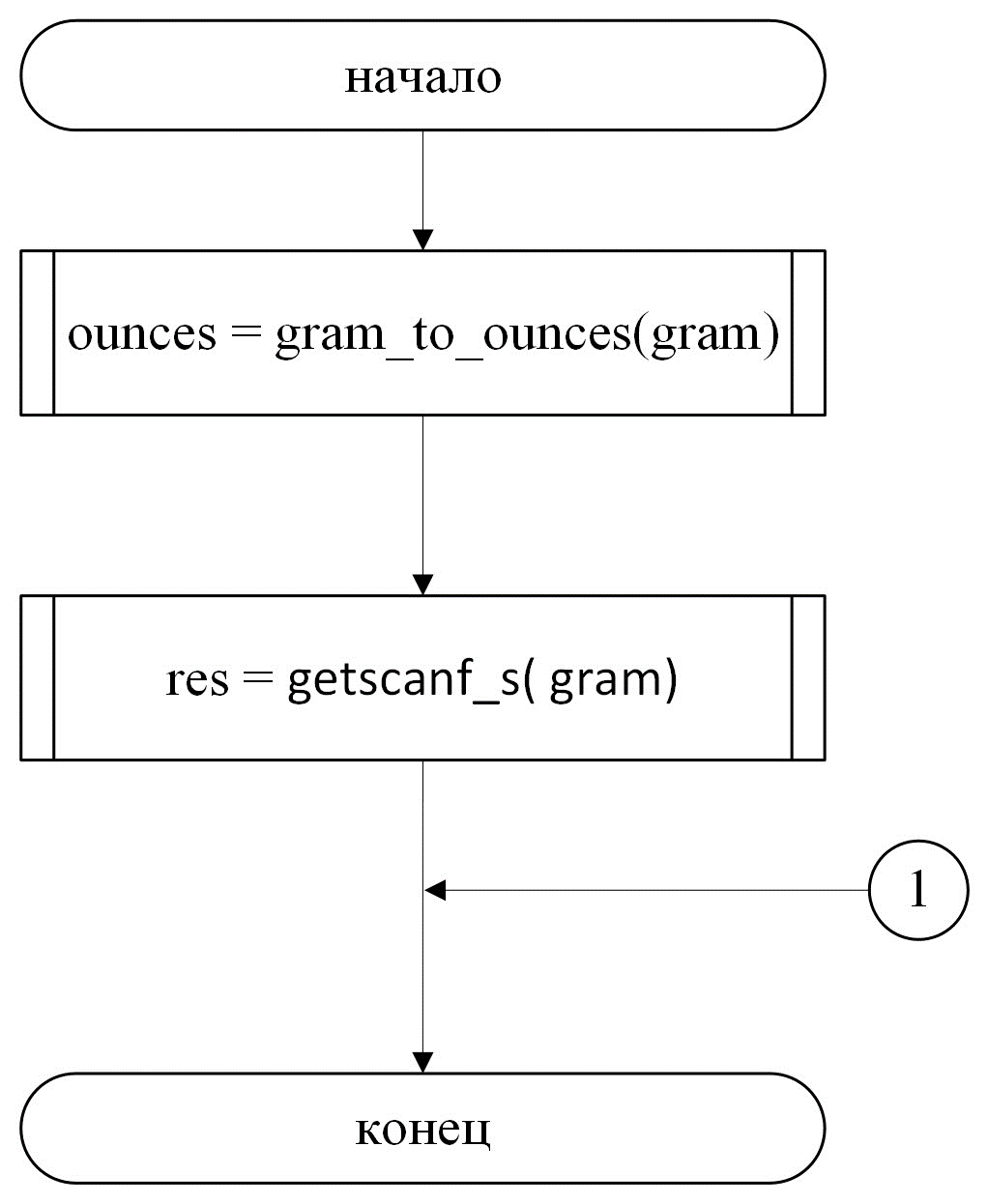


Рисунок 1 ­ Блок-схема основного алгоритма

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 2 – Блок-схема используемой функции

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 3 – Блок-схема используемой функции

2.3 Текст программы на языке С задание 1–2

#include <stdio.h>

#include <math.h>

/\*\*

\* @brief Перевод грамм в унции

\* @return Возвращает унции

\*/

double gram\_to\_ounces(double gram);

/\*\*

\* @brief Получает вещественное число

\* @return Возвращает вещественное число

\*/

double getscanf\_s(double gram);

/\*\*

\* @brief Точка входа в программу

\* @return Возвращает 0, если программа работает корректно, иначе не 0

\*/

int main()

{

double gram;

scanf\_s("%lf", &gram);

getscanf\_s(gram);

double ounces = gram\_to\_ounces(gram);

printf("%lf", ounces);

return 0;

}

double getscanf\_s(double gram)

{

if (gram != 1)

{

printf("%lf", "Введено неверное значение");

abort();

}

}

double gram\_to\_ounces(double gram)

{

return gram \* 28.3;

}

* 1. Результаты выполнения программы 1–2

Результаты выполнения программы в C представлена ниже (Рисунок 4, Рисунок 5, Рисунок 6).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4 – Результаты выполнения программы если грамм =1

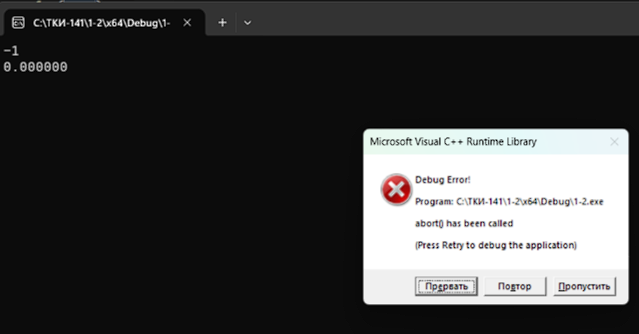


Рисунок 5 – Результаты выполнения программы если грамм =-1

Изображение выглядит как текст, программное обеспечение, веб-страница, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 6 – Результаты выполнения программы если грамм =b

* 1. Выполнение тестовых примеров задание 1–2

В программе Pythonвыполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 7, Рисунок 8, Рисунок 9).



Рисунок 7 – Результат расчета если мы берём 1 грамм

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 8 – Результат расчета если мы берём -1 грамм

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 9 – Результат расчета если мы берём b грамм

* 1. Отметка о выполнении задания в веб-хостинге системы контроля версий задание 1–2

Ниже представлено доказательство того, что задание 1–2 было принято. (Рисунок  10)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 10 – Approve задачи 1–2

1. Решение задачи 1–3

**3.1 Формулировка** **задачи 1–3**

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 1 – Исходные данные

|  |  |
| --- | --- |
| Вариант | Задачи |
| 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? |

3.2 Блок-схема алгоритма задание 1–3

Блок-схема основного алгоритма представлена ниже (Рисунок 1). Блок-схемы функций расчета значений A и проверка представлены ниже (Рисунок 2, Рисунок 3 и Рисунок 4).

Изображение выглядит как текст, снимок экрана, диаграмма, число

Автоматически созданное описание

Рисунок 1 ­ Блок-схема основного алгоритма

Изображение выглядит как текст, Шрифт, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 2 – Блок-схема используемой функции

Изображение выглядит как диаграмма, текст, линия, Технический чертеж

Автоматически созданное описание

Рисунок 3 – Блок-схема используемой функции

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 4 – Блок-схема используемой функции

3.3 Текст программы на языке С задание 1–3

# include <stdio.h>

# include <locale.h>

# include <stdlib.h>

# include <errno.h>

/\*\*

\* @brief Функция проверяет и принимает знпчения

\* @return Значение

\*/

double get\_value\_I();

/\*\*

\* @brief Функция проверяет и принимает знпчения

\* @return Значение

\*/

double get\_value\_U();

/\*\*

\* @brief Функция для нахождения работы в электродвигателе вентилятора

\* @param I - сила тока в электродвигателе вентилятора

\* @param t - время в электродвигателе вентилятора

\* @param U - напряжение в электродвигателе вентилятора

\* @return Работа в электродвигателе вентилятора

\*/

double def\_A(double I, double U, double t);

/\*\*

\* @brief Точка входа в программу

\* @return Возврящает 0, если программа работает верно, иначе 1

\*/

int main()

{

int res;

setlocale(LC\_ALL, "RU");

double t = 1200.0;

printf("Введите силу тока в электродвигателе вентилятора (I): ");

double I;

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

printf("Введите напряжение в электродвигателе вентилятора (U): ");

double U;

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

double A = def\_A(I, U, t);

printf("Работа в электродвигателе вентилятора: %lf", A);

return 0;

}

double get\_value\_I()

{

double I;

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

if (res != 1)

{

errno = EIO;

perror("Wrong value");

abort();

}

return I;

}

double get\_value\_U()

{

double U;

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

if (res != 1)

{

errno = EIO;

perror("Wrong value");

abort();

}

return U;

}

double def\_A(double I, double U, double t)

{

return (I \* U \* t);

}

3.4 Результаты выполнения программы 1–3

Результаты выполнения программы в C представлена ниже (Рисунок 5, Рисунок 6, Рисунок 7).

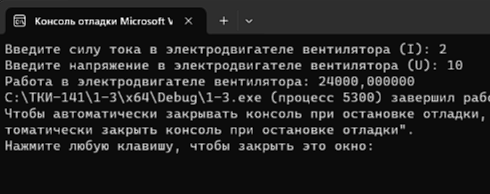


Рисунок 5 – Результаты выполнения программы если I =2 и U=10

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 6 – Результаты выполнения программы если I =-1 и U=-4

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 7 – Результаты выполнения программы если I =9 и U=d

3.5 Выполнение тестовых примеров задание 1–3

В программе Pythonвыполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 8, Рисунок 9, Рисунок 10).

Изображение выглядит как текст, программное обеспечение, снимок экрана

Автоматически созданное описание

Рисунок 8 – Результат расчета если мы берём I =2 и U=10

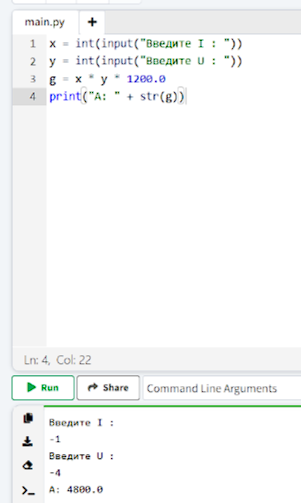


Рисунок 9 – Результат расчета если мы берём I =-1 и U=-4

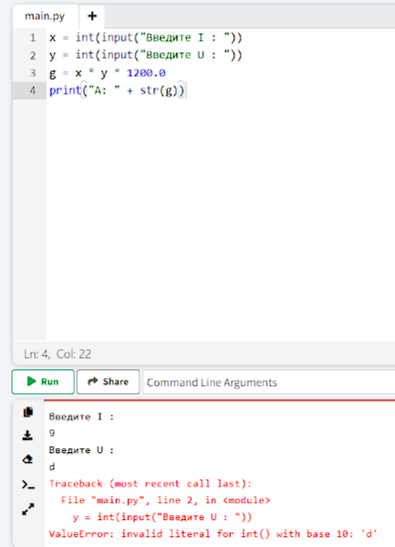


Рисунок 10 – Результат расчета если мы берём I =9 и U=d

3.6 Отметка о выполнении задания в веб-хостинге системы контроля версий задание 1–3

Ниже представлено доказательство того, что задание 1–3 было принято. (Рисунок  11)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 11 – Approve задачи 1–3