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 1

Выполнил: ст. гр. ТКИ - 142

Барсегян Артём Арменович

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

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

Москва 2023

**Оглавление**

[1 Задание 1-1 3](#_Toc156591026)

[1.1 Формулировка задачи 3](#_Toc156591027)

[1.2 Блок-схема алгоритма 4](#_Toc156591028)

[1.3 Решение задачи на языке программирования C 6](#_Toc156591029)

[1.4 Решение тестовых примеров 7](#_Toc156591030)

[1.5 Решение тестовых примеров в Excel. 7](#_Toc156591031)

[1.6 Зачёт задания в GitHub 7](#_Toc156591032)

[2 Задание 1-2 8](#_Toc156591033)

[2.1 Формулировка задачи 8](#_Toc156591034)

[2.2 Блок-схема алгоритма 9](#_Toc156591035)

[2.3 Решение задачи на языке программирования C 12](#_Toc156591036)

[2.4 Решение тестовых примеров 14](#_Toc156591037)

[2.5 Решение тестовых примеров в Excel 15](#_Toc156591038)

[2.6 Зачёт задания в GitHub 15](#_Toc156591039)

[3 Задание 1-3 16](#_Toc156591040)

[3.1 Формулировка задачи 16](#_Toc156591041)

[3.2 Блок-схема алгоритма 17](#_Toc156591042)

[3.3 Решение задачи на языке программирования C. 20](#_Toc156591043)

[3.4 Решение тестовых примеров. 21](#_Toc156591044)

[3.5 Решение тестовых примеров в Excel 21](#_Toc156591045)

[3.6 Зачёт задания в GitHub. 22](#_Toc156591046)

# Задание 1-1

* 1. Формулировка задачи

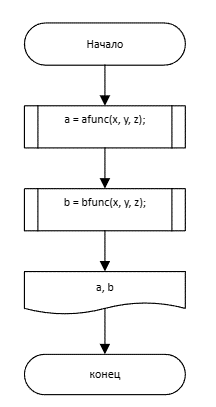
Создать консольное приложение, вычисляющее значения переменных по представленным в таблице формулам. Расчёт примера осуществить по заданным константам. Вывести на экран значения исходных данных, а также результат вычислений. Дополнить свой отчёт блок-схемой алгоритма.

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Формула** | **Константы** |
| 1 |  | x=1.426  y=-1.22  z=3.5 |

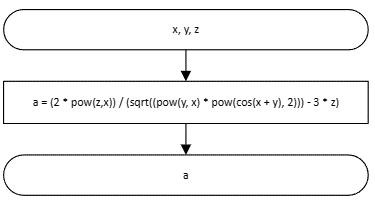
Таблица 1 – Условие задачи

* 1. Блок-схема алгоритма

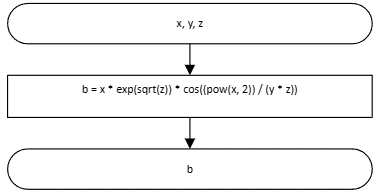
Блок-схемы алгоритмов функций представлены на рисунках (Рисунок 1, Рисунок 2, Рисунок 3).



**Рисунок 1 – Блок-схема функции main()**



**Рисунок 2 – Блок-схема функции afunc(x, y, z)**



**Рисунок 3 – Блок-схема функции bfunc(x, y, z)**

* 1. Решение задачи на языке программирования C

#include <stdio.h>

#include <math.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 = 0.78;

const double y = 1.24;

const double z = 0.5;

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

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

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

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

return 0;

}

double afunc(double x, double y, double z)

{

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

}

double bfunc(double x, double y, double z)

{

return x \* exp(sqrt(z)) \* cos((pow(x, 2)) / (y \* z));

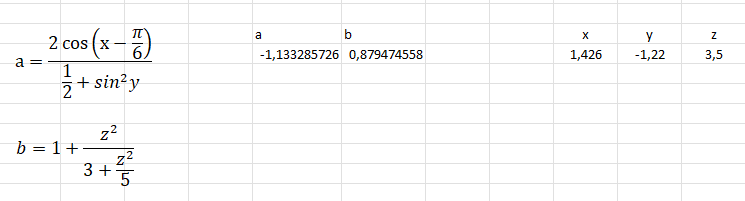
}

* 1. Решение тестовых примеров



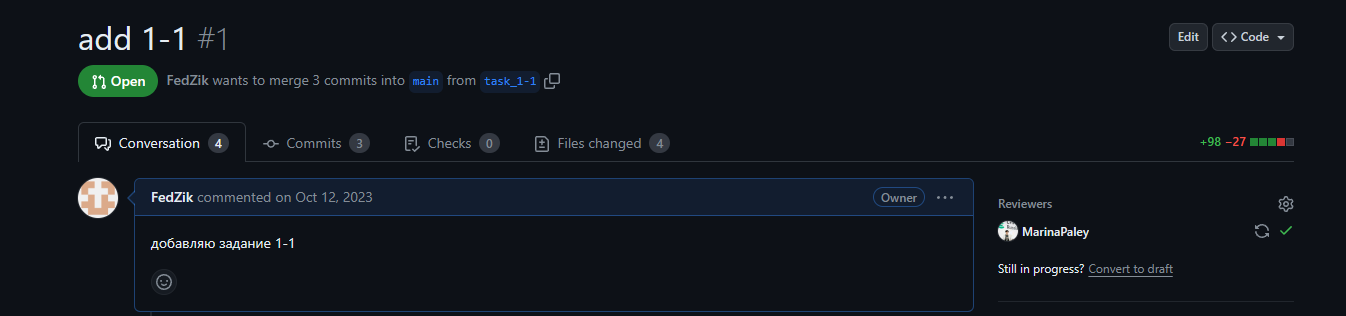
**Рисунок 4 – Решение тестового примера**

* 1. Решение тестовых примеров в Excel.



**Рисунок 5 – Расчёт значений**

* 1. Зачёт задания в GitHub



**Рисунок 6 – Зачёт задания в GitHub**

# Задание 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 2 – Условие задачи

|  |  |
| --- | --- |
| Вариант | Задачи |
| 1 | Вычислить площадь и периметр прямоугольника, если задана длина одной стороны (*a*) и коэффициент *n* (%), позволяющий вычислить длину второй стороны (*b=n\*a*). |

* 1. Блок-схема алгоритма

Блок-схемы алгоритмов функций представлены на рисунках (Рисунок 7, Рисунок 8, Рисунок 9, Рисунок 10, Рисунок 11).

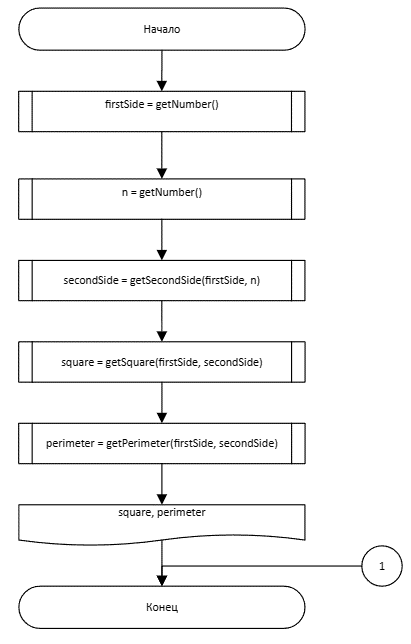


Рисунок 7 – Блок-схема функции main()

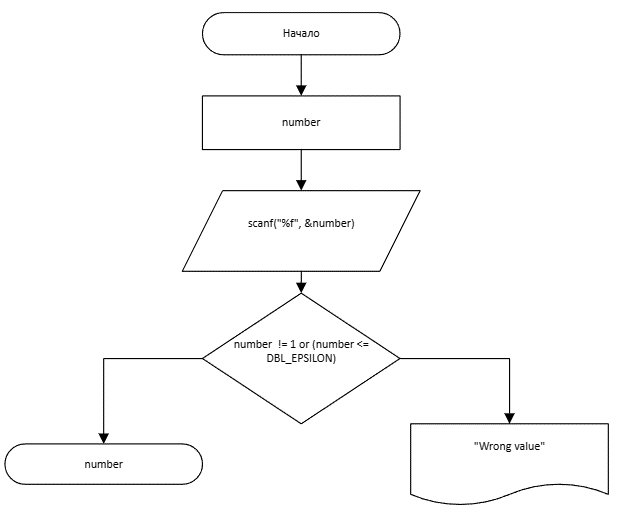


Рисунок 8 – Блок-схема функции getNumber()

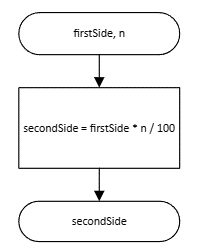


Рисунок 9 – Блок-схема функции getSecondSide(firstSide, n)

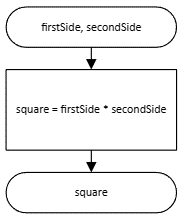


Рисунок 10 – Блок-схема функции getSquare(firstSide, secondSide)

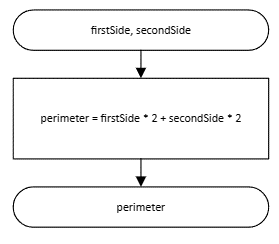


Рисунок 11 – Блок-схема функции getPerimeter(firstSide, secondSide)

* 1. Решение задачи на языке программирования C

#include <stdio.h>

#include <stdlib.h>

#include <float.h>

/\*\*

\* @brief расчет второй стороны

\* @param firstSide - первая сторона

\* @param n - коэффициент, на который надо умножить 1 сторону для получения второй

\* @return возвращает численное значение второй стороны

\*/

float getSecondSide(float firstSide, float n);

/\*\*

\* @brief расчет площади

\* @param firstSide первая сторона

\* @param secondSide вторая сторона

\* @return возвращает численное значение площади

\*/

float getSquare(float firstSide, float secondSide);

/\*\*

\* @brief расчет периметра

\* @param firstSide первая сторона

\* @param secondSide вторая сторона

\* @return возвращает численное значение периметра

\*/

float getPerimeter(float firstSide, float secondSide);

/\*\*

\* @brief проверка введеного значения

\* @return возвращает значение если ввод правильный, иначе выводит сообщение об ошибке

\*/

float getNumber();

/\*\*

\* @brief точка входа в программу

\* @return возвращает 0, если программа работает корректно, иначе 1

\*/

int main()

{

float firstSide = getNumber();

float n = getNumber();

float secondSide = getSecondSide(firstSide, n);

float square = getSquare(firstSide, secondSide);

float perimeter = getPerimeter(firstSide, secondSide);

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

printf("\n preimeter=%f", perimeter);

return 0;

}

float getSecondSide(float firstSide, float n)

{

return firstSide \* n / 100;

}

float getSquare(float firstSide, float secondSide)

{

return firstSide \* secondSide;

}

float getPerimeter(float firstSide, float secondSide)

{

return firstSide \* 2 + secondSide \* 2;

}

float getNumber()

{

float number;

if (scanf("%f", &number) != 1 || (number <= DBL\_EPSILON))

{

printf("%s" "Wrong value");

abort();

}

return number;

}

* 1. Решение тестовых примеров

Вывод программы в разных случаях представлен на рисунках (Рисунок 13, Рисунок 14, Рисунок 15, Рисунок 16)

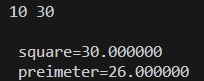


Рисунок 12 – Решение тестового примера



Рисунок 13 ***–***Вывод программы, если переменная firstSide отрицательная

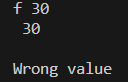


Рисунок 14 ***–***Вывод программы, если переменная firstSide – буква



Рисунок 15 – Вывод программы, если переменная n отрицательная

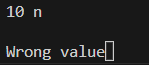
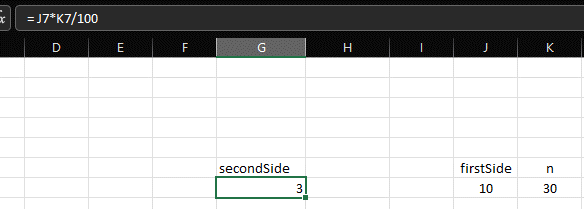


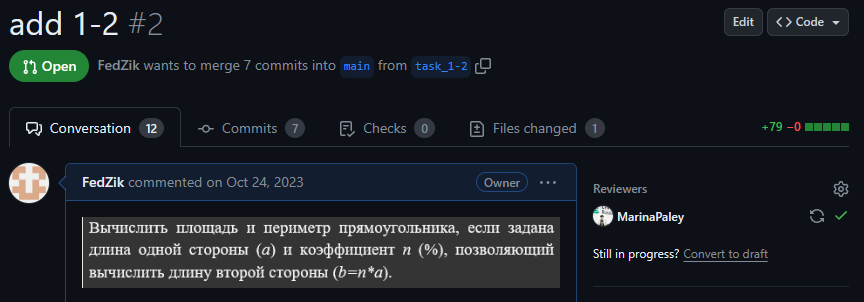
Рисунок 16 – Вывод программы, если переменная n – буква

* 1. Решение тестовых примеров в Excel



**Рисунок 17 – Расчёт значения secondSide**

* 1. Зачёт задания в GitHub



**Рисунок 17 –**Зачёт задания в GitHub

# Задание 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 3 – Условие задачи

|  |  |
| --- | --- |
| Вариант | Задачи |
| 1 | Чему равна кинетическая энергия пули массой *m* г, летящей со скоростью *v* м/с? |

* 1. Блок-схема алгоритма

Блок-схемы алгоритмов представлены на рисунках (Рисунок 18, Рисунок 19, Рисунок 20)

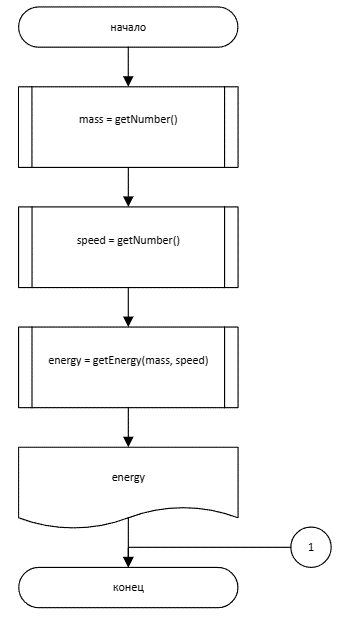


Рисунок 18 – Блок-схема к функции main()

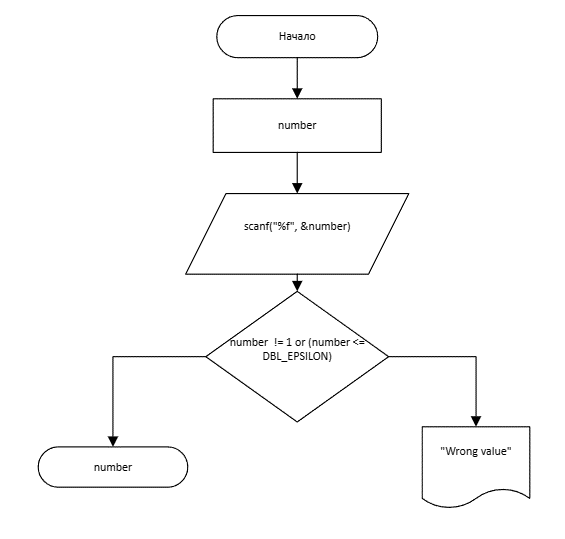


Рисунок 19 – Блок-схема к функции getNumber()

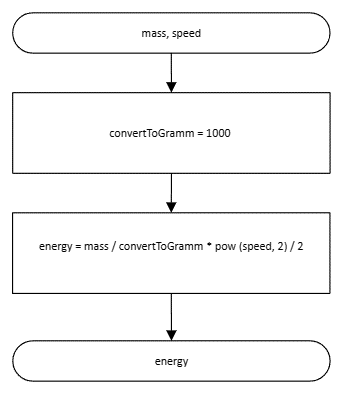


Рисунок 20 – Блок-схема к функции getEnergy(mass, speed)

* 1. Решение задачи на языке программирования C.

#include <stdio.h>

#include <float.h>

/\*\*

\* @brief расчет кинетической энергии

\* @param mass - масса

\* @param speed - скорость

\* @return возвращает численное значение кинетической энергии

\*/

double getEnergy(double mass, double speed);

/\*\*

\* @brief проверка введеного значения

\* @param convertToGramm значение для перевода массы в систему СИ

\* @return возвращает значение если ввод правильный, иначе выводит сообщение об ошибке

\*/

float getNumber();

/\*\*

\* @brief точка входа в программу

\* @return возвращает 0, если программа работает корректно, иначе 1

\*/

int main()

{

double mass = getNumber();

double speed = getNumber();

double energy = getEnergy(mass, speed);

printf("\n energy=%lf", energy);

return 0;

}

double getEnergy(double mass, double speed)

{

const double convertToGramm = 1000;

return mass / convertToGramm \* pow (speed, 2) / 2;

}

float getNumber()

{

float number;

if (scanf("%f", &number) != 1 || (number <= DBL\_EPSILON))

{

printf("%s" "Wrong value");

abort();

}

return number;

}

* 1. Решение тестовых примеров.



Рисунок 21 – Решение тестового примера



Рисунок 22 – Вывод программы, если переменная mass отрицательная

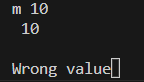


Рисунок 23 – Вывод программы, если переменная mass – буква



Рисунок 24 – Вывод программы, если переменная speed отрицательная

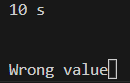


Рисунок 25 – Вывод программы, если переменная speed – буква

* 1. Решение тестовых примеров в Excel

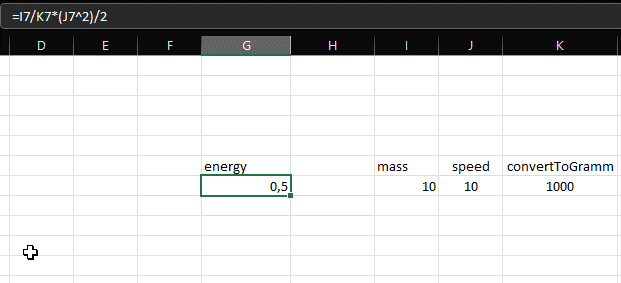


Рисунок 26 – Решение тестового примера в Excel

* 1. Зачёт задания в GitHub.

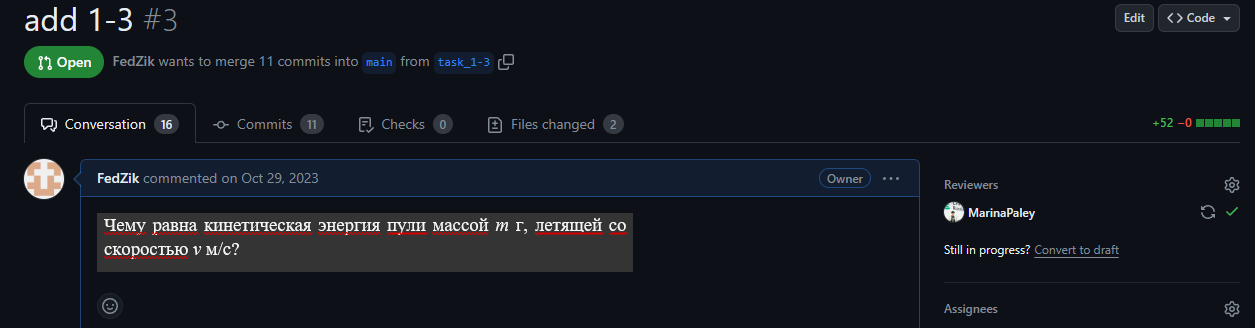


Рисунок 27 – Зачёт задания в GitHub