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 8

Выполнил: ст. гр. ТКИ - 142

Лазарев Андрей Александрович

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

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

Москва 2023

**Оглавление**

[1 Задание 1-1 3](#_Toc152708342)

[1.1 Формулировка задачи 3](#_Toc152708343)

[1.2 Блок-схема алгоритма 3](#_Toc152708344)

[1.3 Решение задачи на языке программирования C. 5](#_Toc152708345)

[1.4 Решение тестовых примеров. 6](#_Toc152708346)

[1.5 Решение тестовых примеров в Excel. 6](#_Toc152708347)

[1.6 Зачёт задания в GitHub 6](#_Toc152708348)

[2 Задание 1-2 7](#_Toc152708349)

[2.1 Формулировка задачи. 7](#_Toc152708350)

[2.2 Блок-схема алгоритма. 7](#_Toc152708351)

[2.3 Решение задачи на языке программирования C. 11](#_Toc152708352)

[2.4 Решение тестовых примеров. 13](#_Toc152708353)

[2.5 Решение тестовых примеров в Excel 13](#_Toc152708354)

[2.6 Зачёт задания в GitHub. 13](#_Toc152708355)

[3 Задание 1-3 13](#_Toc152708356)

[3.1 Формулировка задачи. 13](#_Toc152708357)

[3.2 Блок-схема алгоритма. 14](#_Toc152708358)

[3.3 Решение задачи на языке программирования C. 16](#_Toc152708359)

[3.4 Решение тестовых примеров. 17](#_Toc152708360)

[3.5 Решение тестовых примеров в Excel 18](#_Toc152708361)

[3.6 Зачёт задания в GitHub. 18](#_Toc152708362)

# Задание 1-1

* 1. Формулировка задачи

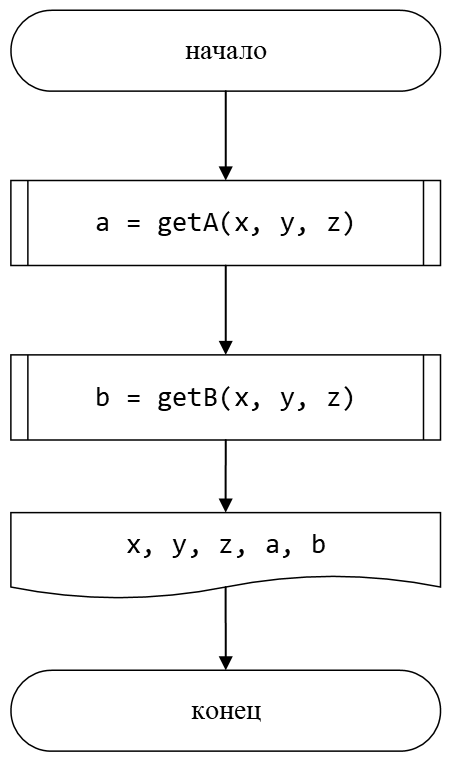
Создать консольное приложение, вычисляющее значения переменных по представленным в таблице формулам. Расчёт примера осуществить по заданным константам. Вывести на экран значения исходных данных, а также результат вычислений. Дополнить свой отчёт блок-схемой алгоритма.

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Формула** | **Константы** |
| 8 |  | x=0.2  y=0.004  z=1.1 |

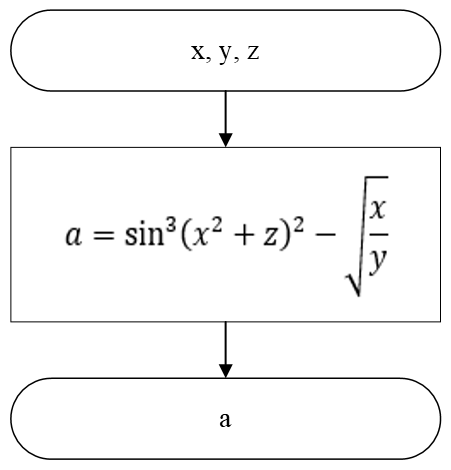
Таблица  – Условие задачи

* 1. Блок-схема алгоритма

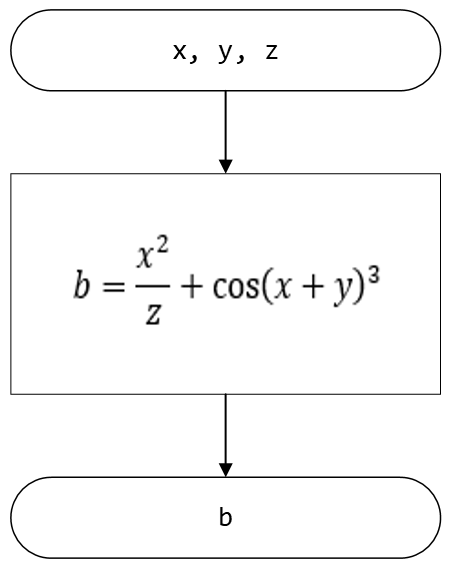
Блок-схемы алгоритмов функций представлены на рисунках (Рисунок 1, Рисунок 2, Рисунок 3).



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



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



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

* 1. Решение задачи на языке программирования C

#include <stdio.h>

#include <math.h>

/\*\*

\* @brief Функция расчёта по заданной формуле.

\* @param x Аргумент функции.

\* @param y Аргумент функции.

\* @param z Аргумент функции.

\* @return Возвращает значение функции.

\*/

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

/\*\*

\* @brief Функция расчёта по заданной формуле.

\* @param x Аргумент функции.

\* @param y Аргумент функции.

\* @param z Аргумент функции.

\* @return Возвращает значение функции.

\*/

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

/\*\*

\* @brief Точка входа в программу.

\* @return Возвращает 0 в случае успеха.

\*/

int main()

{

const double x = 0.2;

const double y = 0.004;

const double z = 1.1;

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

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

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

printf("a = %lf b = %lf", a, b);

return 0;

}

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

{

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

}

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

{

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

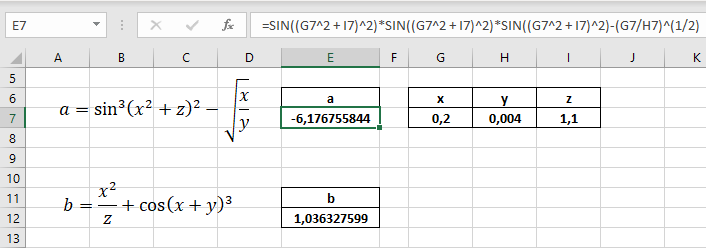
}

* 1. Решение тестовых примеров

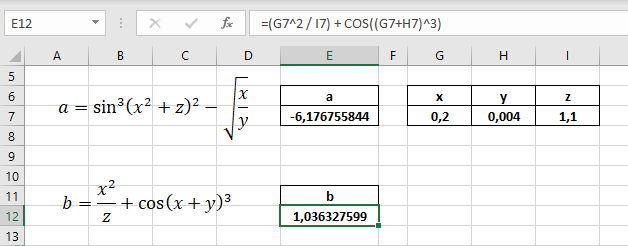


**Рисунок 4 – Решение тестового примера**

* 1. Решение тестовых примеров в Excel.

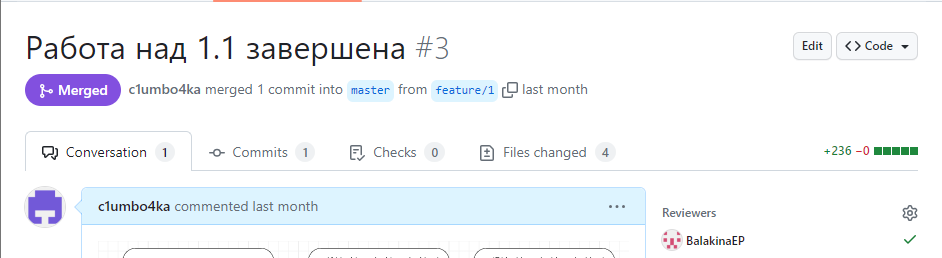


**Рисунок 5 – Расчёт значения a**



**Рисунок 6 – Расчёт значения b**

* 1. Зачёт задания в GitHub



**Рисунок 7 – 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  – Условик задачи

|  |  |
| --- | --- |
| Вариант | Задачи |
| 8 | В шар радиуса R вписан конус с углом a при вершине в осевом сечении конуса. Определить объем и полную поверхность конуса. |

* 1. Блок-схема алгоритма

Блок-схемы алгоритмов функций представлены на рисунках (Рисунок 8, Рисунок 9, Рисунок 10, Рисунок 11, Рисунок 12, Рисунок 13).

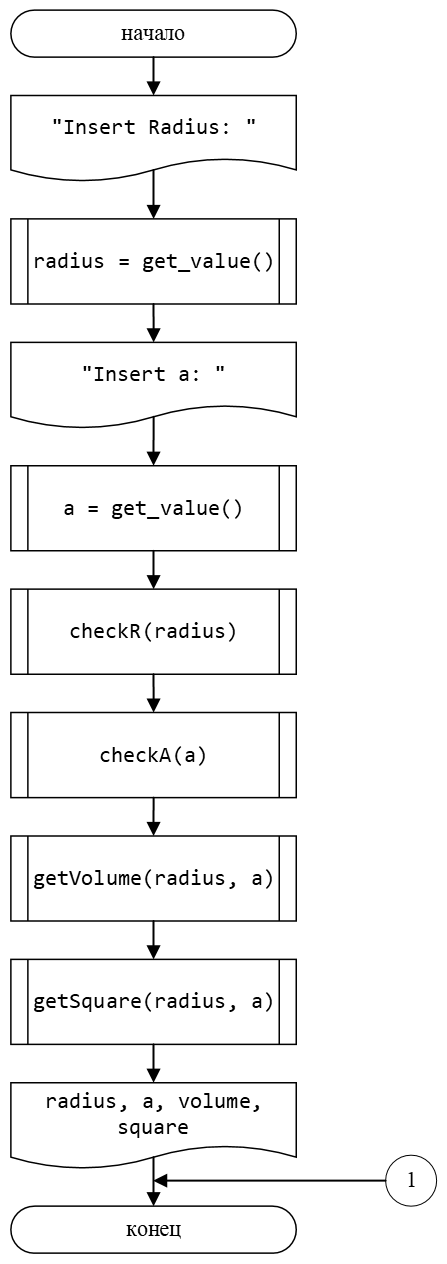


Рисунок 8 – Блок-схема функции main()

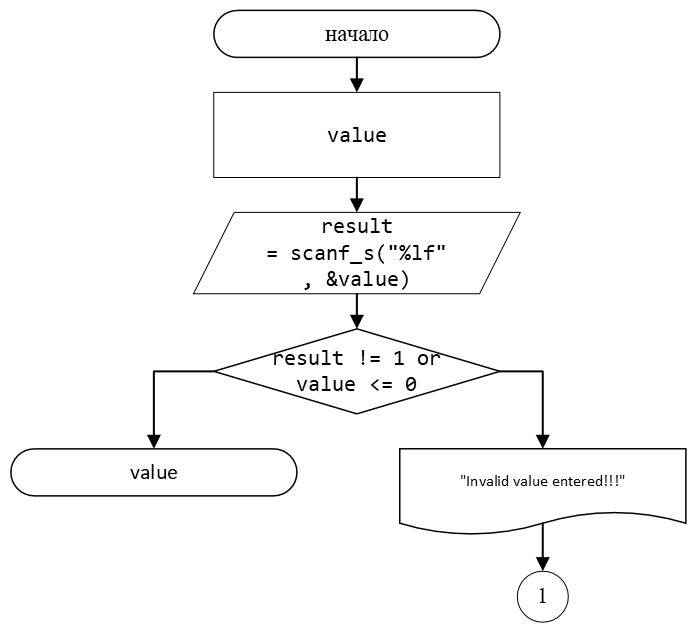


Рисунок 9 – Блок-схема функции get\_value()

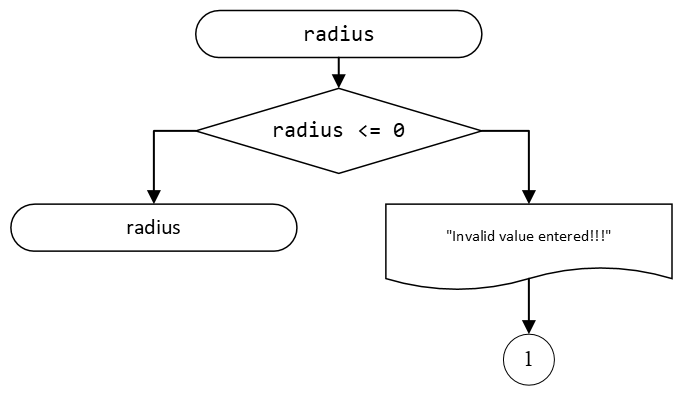


Рисунок 10 – Блок-схема функции checkR(radius)

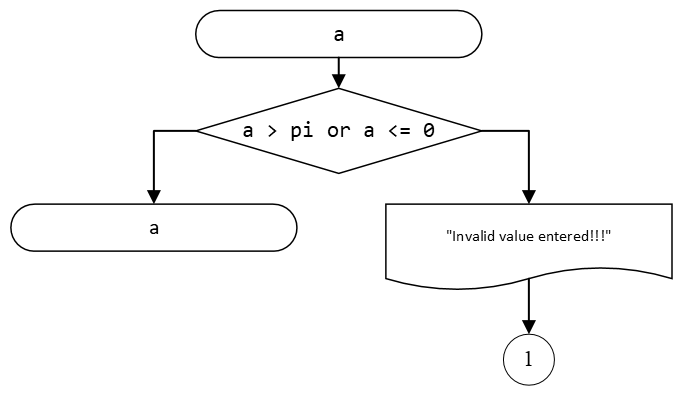


Рисунок 11 – Блок-схема функции checkA(a)

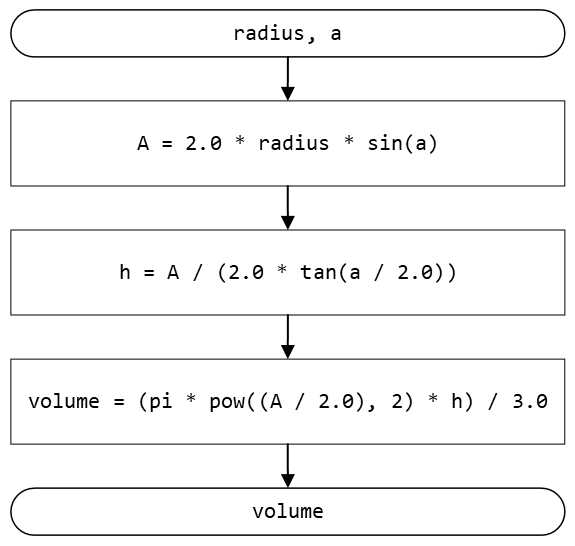


Рисунок 12 – Блок-схема функции getVolume(radius, a)

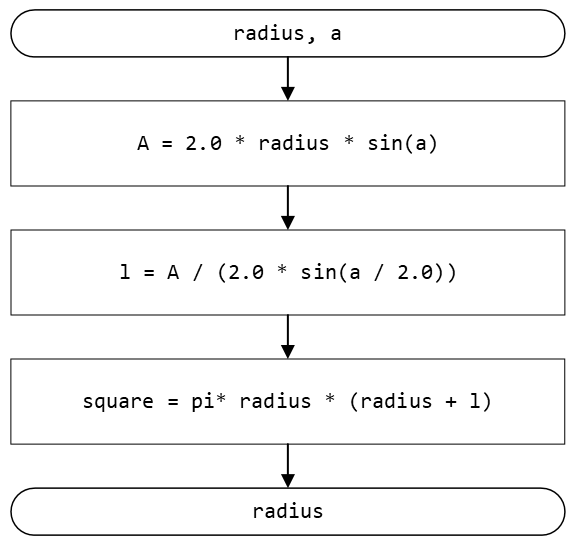


Рисунок 13 – Блок-схема функции getSquare(radius, a)

* 1. Решение задачи на языке программирования C

#define \_USE\_MATH\_DEFINES

#include <stdlib.h>

#include <math.h>

#include <stdio.h>

#include <float.h>

#include <errno.h>

/\*\*

\* @brief Функция проверки переменной Radius.

\* @param Radius Аргумент функции.

\* @return Возвращает истину в случае успеха.

\*/

double checkR(double radius);

/\*\*

\* @brief Функция проверки переменной a.

\* @param a Аргумент функции.

\* @return Возвращает истину в случае успеха.

\*/

double checkA(double a);

/\*\*

\* @brief Функция расчета объёма.

\* @param Radius Аргумент функции.

\* @param a Аргумент функции.

\* @return Возвращает значение функции.

\*/

double getVolume(double radius, double a);

/\*\*

\* @brief Функция расчета площади.

\* @param Radius Аргумент функции.

\* @param a Аргумент функции.

\* @return Возвращает значение функции.

\*/

double getSquare(double radius, double a);

/\*\*

\* @brief Функция проверки ввода на правильность.

\* @return Возвращает значение в случае успеха.

\*/

double get\_value();

/\*\*

\* @brief Точка входа в программу.

\* @return Возвращает 0 в случае успеха.

\*/

int main()

{

printf("%s", "Insert Radius: ");

double radius = get\_value();

printf("%s", "Insert a: ");

double a = get\_value();

checkR(radius);

checkA(a);

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

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

printf("Square = %lf", getSquare(radius, a));

return 0;

}

double checkR(double radius)

{

if (radius <= DBL\_EPSILON)

{

printf("%s", "Error!!!");

abort();

}

}

double checkA(double a)

{

if (a - M\_PI > -DBL\_EPSILON || a <= DBL\_EPSILON)

{

printf("%s", "Error!!!");

abort();

}

}

double getVolume(double radius, double a)

{

double side = 2.0 \* radius \* sin(a);

double h = side / (2.0 \* tan(a / 2.0));

return (M\_PI \* pow((side / 2.0), 2) \* h) / 3.0;

}

double getSquare(double radius, double a)

{

double side = 2.0 \* radius \* sin(a);

double l = side / (2.0 \* sin(a / 2.0));

return M\_PI \* radius \* (radius + l);

}

double get\_value()

{

double value;

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

if (result != 1 || value <= 0)

{

errno = EIO;

perror("Error: ");

abort();

}

return value;

}

* 1. Решение тестовых примеров

Вывод программы в разных случаях представлен на рисунках (Рисунок 14, Рисунок 15, Рисунок 16, Рисунок 17, Рисунок 18)

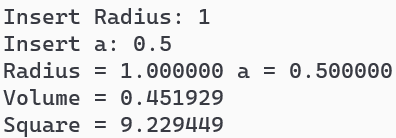


Рисунок 14 – Решение тестового примера

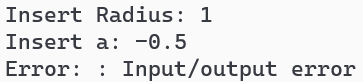


Рисунок 15 ***–***Вывод программы, если переменная a отрицательная

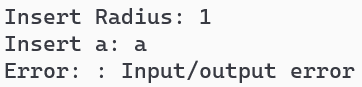


Рисунок 16 ***–***Вывод программы, если переменная a – буква

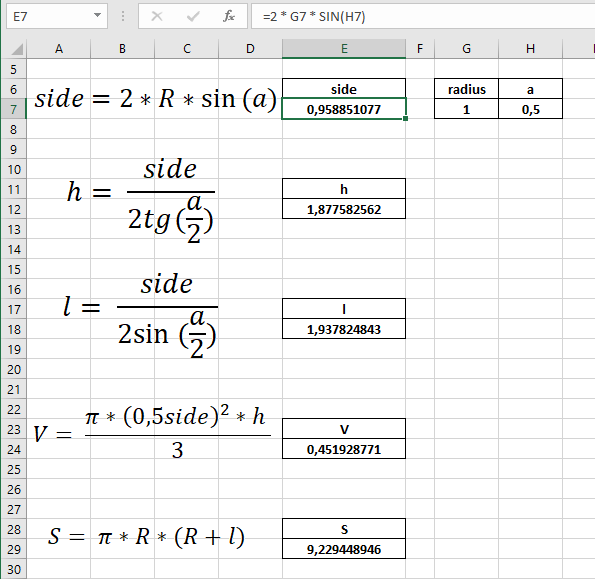


Рисунок 17 – Вывод программы, если переменная radius отрицательная

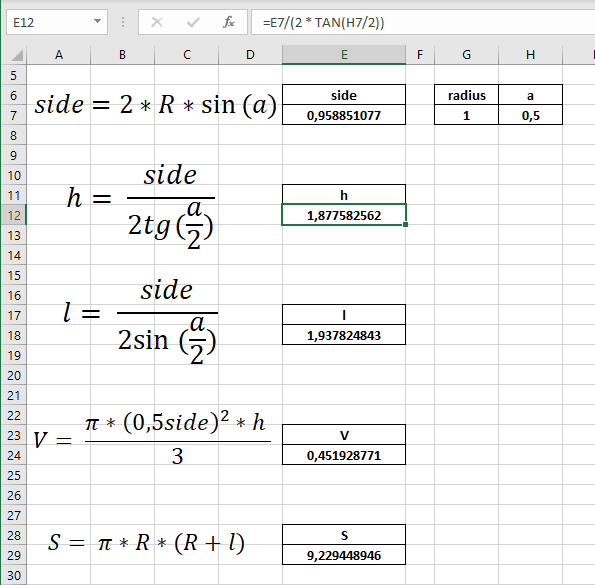


Рисунок 18 – Вывод программы, если переменная radius – буква

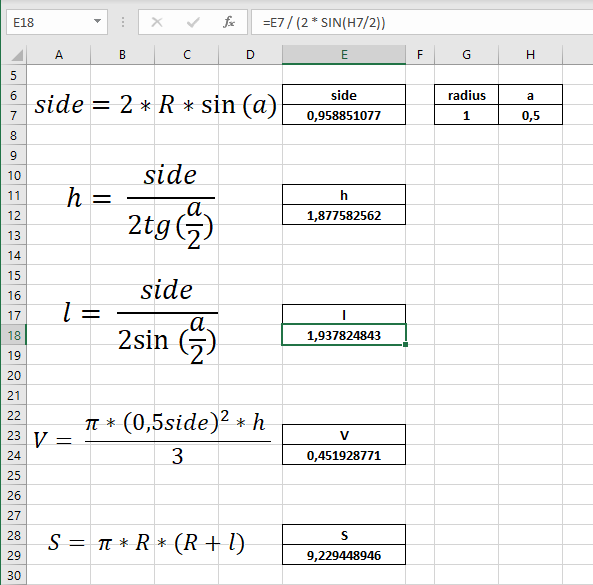
* 1. Решение тестовых примеров в Excel



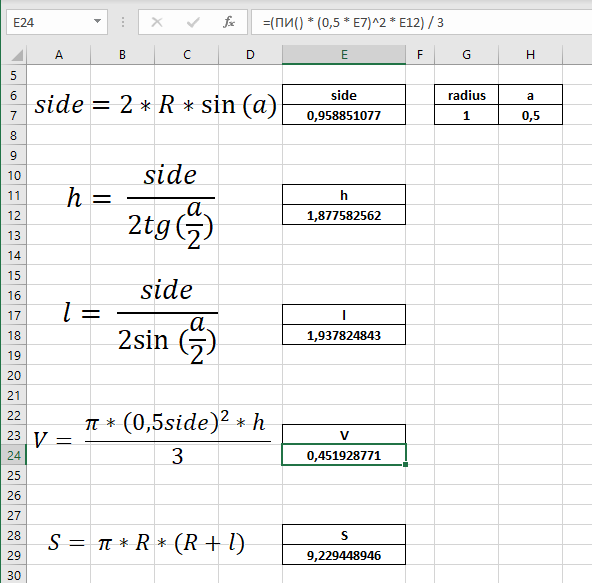
**Рисунок 19 – Расчёт значения side**



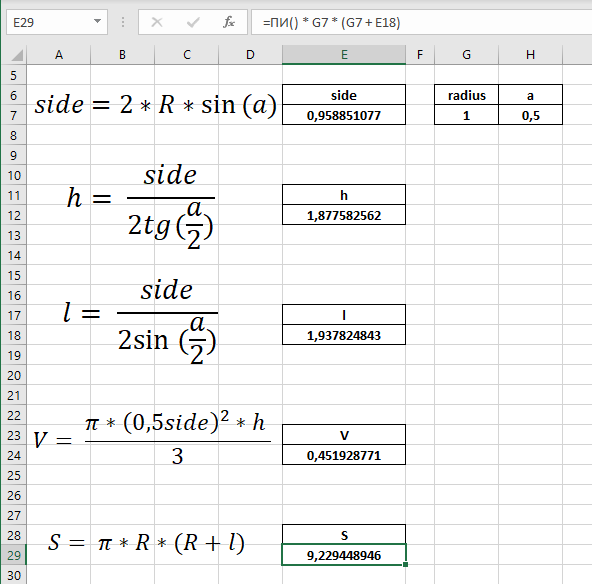
**Рисунок 20 – Расчёт значения h**



**Рисунок 21 – Расчёт значения l**

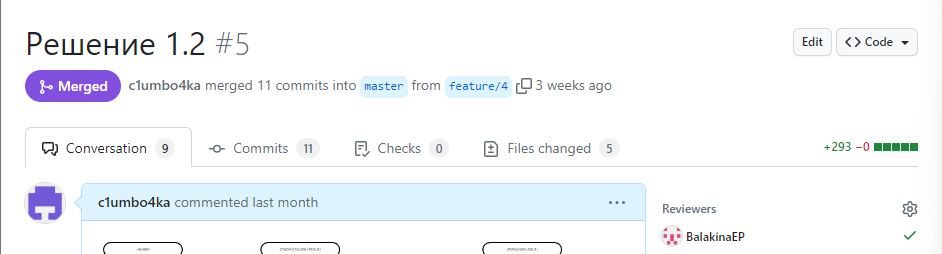


**Рисунок 22 – Расчёт значения V**



**Рисунок 23 – Расчёт значения V**

* 1. 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  – Условие задачи

|  |  |
| --- | --- |
| Вариант | Задачи |
| 8 | Определите путь, пройденный автомобилем, если при силе тяги *F* кН совершённая работа равна *A* МДж. |

* 1. Блок-схема алгоритма

Блок-схемы алгоритмов представлены на рисунках (Рисунок 24, Рисунок 25, Рисунок 26, Рисунок 27)

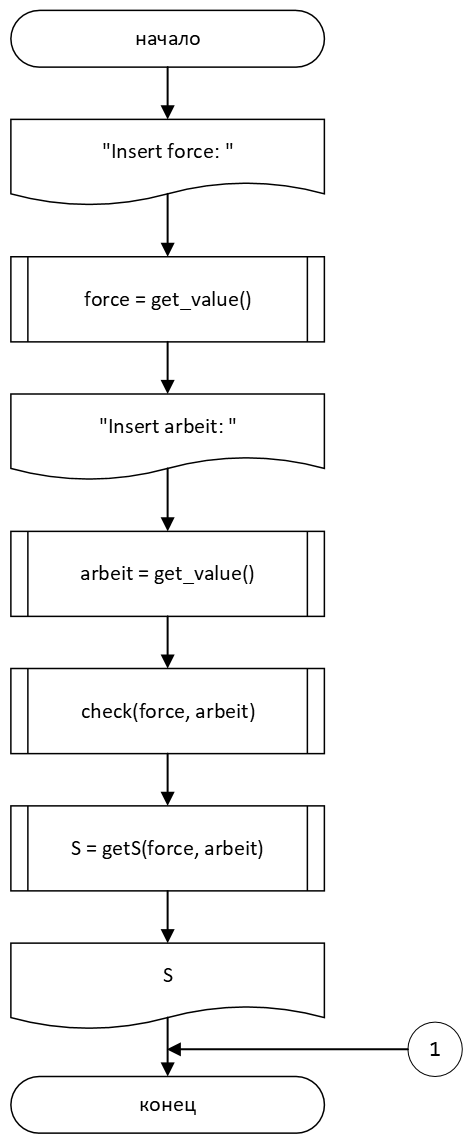


Рисунок 24 – Блок-схема к функции main()

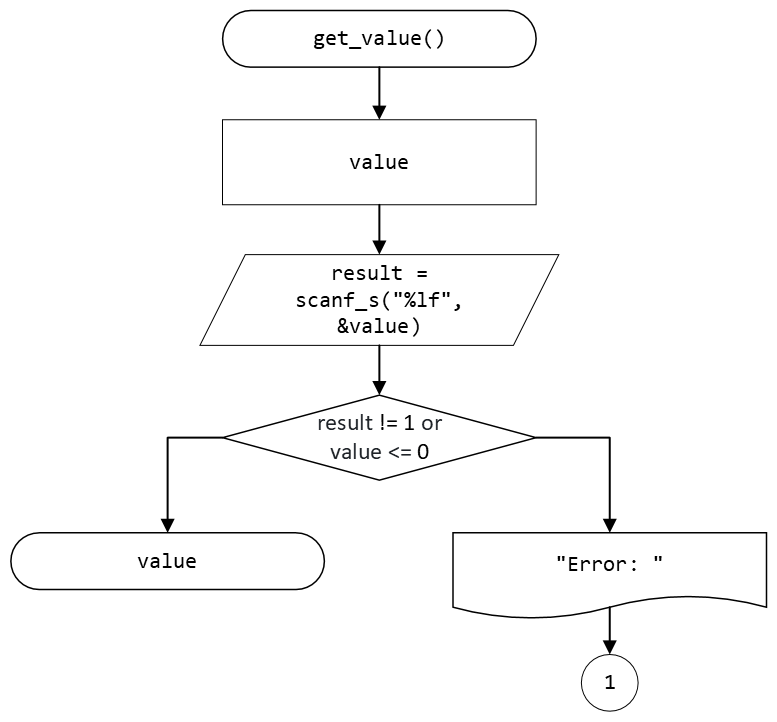


Рисунок 25 – Блок-схема к функции get\_value()

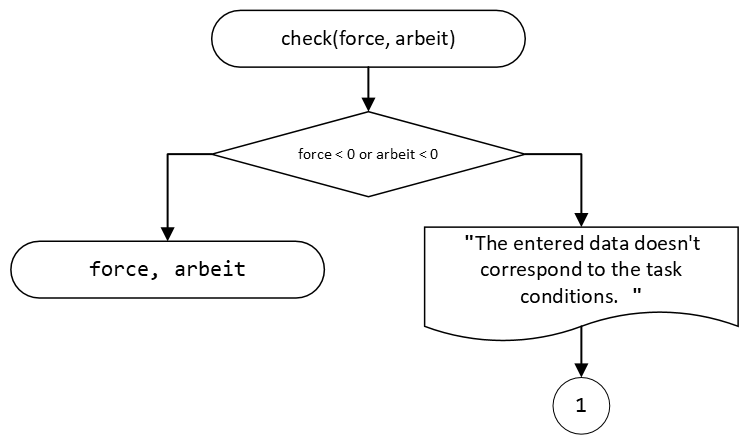


Рисунок 26 – Блок-схема к функции check(force, arbeit)

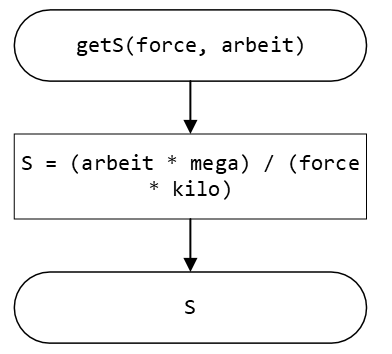


Рисунок 27 – Блок-схема к функции getS(force, arbeit)

* 1. Решение задачи на языке программирования C.

#include <stdio.h>

#include <errno.h>

#include <math.h>

#include <stdlib.h>

#include <float.h>

/\*\*

\* @brief Функция проверки ввода на правильность.

\* @return Возвращает значение в случае успеха.

\*/

double get\_value();

/\*\*

\* @brief Фукция проверки переменных force и arbeit.

\* @param force Аргумент функции.

\* @param arbeit Аргумент функции.

\* @return Возвращает ошибку в случае успеха.

\*/

double check(double force, double arbeit);

/\*\*

\* @brief Функция расчета пути.

\* @param force Аргумент функции.

\* @param arbeit Аргумент функции.

\* @return Возвращает значение функции.

\*/

double getS(double force, double aarbeit);

/\*\*

\* @brief Точка входа в программу.

\* @return Возвращает 0 в случае успеха.

\*/

int main()

{

printf("%s", "Insert force: ");

double force = get\_value();

printf("%s", "Insert arbeit: ");

double arbeit = get\_value();

check(force, arbeit);

printf("The path traveled by the car: %lf meters", getS(force, arbeit));

return 0;

}

double check(double force, double arbeit)

{

if ((force < DBL\_EPSILON) || (arbeit < DBL\_EPSILON))

{

printf("%s", "The entered data doesn't correspond to the task conditions.");

abort();

}

}

double getS(double force, double arbeit)

{

const double mega = pow(10, 6);

const double kilo = pow(10, 3);

return (arbeit \* mega) / (force \* kilo);

}

double get\_value()

{

double value;

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

if (result != 1 || value <= 0)

{

errno = EIO;

perror("Error");

abort();

}

return value;

}

* 1. Решение тестовых примеров.

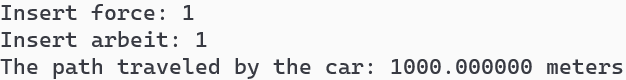


Рисунок 28 – Решение тестового примера

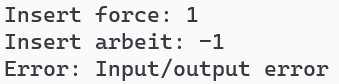


Рисунок 29 – Вывод программы, если переменная arbeit отрицательная

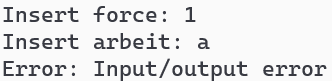


Рисунок 30 – Вывод программы, если переменная arbeit – буква



Рисунок 31 – Вывод программы, если переменная force отрицательная



Рисунок 32 – Вывод программы, если переменная force – буква

* 1. Решение тестовых примеров в Excel

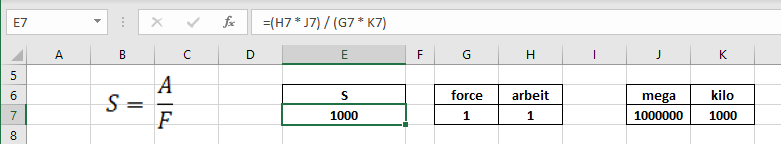


Рисунок 33 – Решение тестового примера в Excel

* 1. Зачёт задания в GitHub.

