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 11

Выполнил: ст. гр. ТКИ-142

Луканкин Вячеслав Сергеевич

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

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

Москва 2023

Оглавление

**Задание 1.11**

1.1 ФОРМУЛИРОВКА ЗАДАНИЯ 1

1.2 БЛОК-СХЕМА АЛГОРИТМА2-3

1.3 ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ C4-5

1.4 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ6

1.5 ВЫПОЛНЕНИЕ ТЕСТОВЫХ ПРИМЕРОВ7

1.6 ОТМЕТКА О ВЫПОЛНЕНИИ ЗАДАНИЯ В ВЕБ-ХОСТИНГЕ СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ8

**Задание 1.2** 9

1.1 ФОРМУЛИРОВКА ЗАДАНИЯ 9

1.2 БЛОК-СХЕМА АЛГОРИТМА10-12

1.3 ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ C12-14

1.4 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ15

1.5 ВЫПОЛНЕНИЕ ТЕСТОВЫХ ПРИМЕРОВ16

1.6 ОТМЕТКА О ВЫПОЛНЕНИИ ЗАДАНИЯ В ВЕБ-ХОСТИНГЕ СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ17

**Задание 1.3 18**

1.1 ФОРМУЛИРОВКА ЗАДАНИЯ 18

1.2 БЛОК-СХЕМА АЛГОРИТМА19-20

1.3 ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ C21-22

1.4 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ23

1.5 ВЫПОЛНЕНИЕ ТЕСТОВЫХ ПРИМЕРОВ24

1.6 ОТМЕТКА О ВЫПОЛНЕНИИ ЗАДАНИЯ В ВЕБ-ХОСТИНГЕ СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ25

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 |
| 11 |  | x=2.2  y=9.2  z=10.2 |

* 1. Блок-схема алгоритма

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

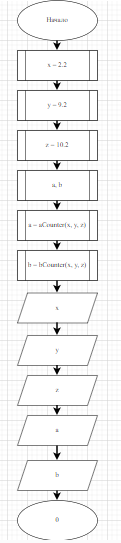


Рисунок 1 ­ Блок-схема основного алгоритма

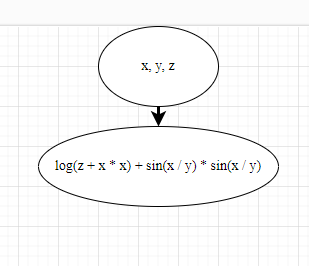


Рисунок 2 – Блок-схема первой используемых функций

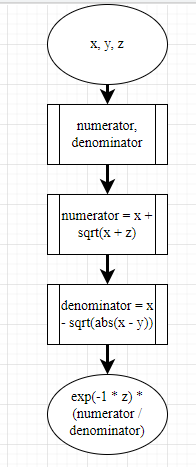


Рисунок 3 – Блок-схема второй вспомогательной функции

* 1. Текст программы на языке C

#include <stdio.h>

#include <math.h>

/\*\*

\* @brief Возвращает результат переменной а

\* @param x Заданная константа

\* @param y Заданная константа

\* @param z Заданная константа

\*/

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

/\*\*

\* @brief Возвращает результат переменной b

\* @param x Заданная константа

\* @param y Заданная константа

\* @param z Заданная константа

\*/

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

int main(){

float x = 2.2f, y = 9.2f, z = 10.2f;

double a, b;

a = aCounter(x, y, z);

b = bCounter(x, y, z);

printf("x equals: %.1f\n", x);

printf("y equals: %.1f\n", y);

printf("z equals: %.1f\n", z);

printf("A equals: %.2f\n", a);

printf("B equals: %.5f\n", b);

return 0;

}

double aCounter(double x, double y, double z){

return log(z + x \* x) + sin(x / y) \* sin(x / y);

}

double bCounter(double x, double y, double z){

double numerator, denominator;

numerator = x + sqrt(x + z);

denominator = x - sqrt(abs(x - y));

return exp(-1 \* z) \* (numerator / denominator);

}

* 1. Результаты выполнения программы

Результаты выполнения программы представлены ниже (Рисунок 3).

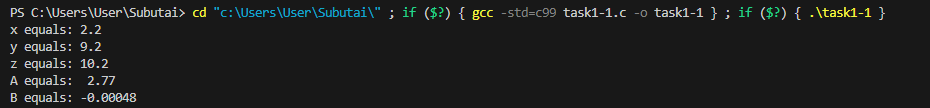


Рисунок 4 – Результаты выполнения программы

* 1. Выполнение тестовых примеров

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

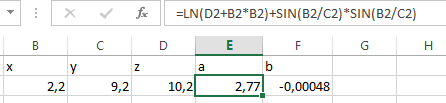


Рисунок 5 – Результат расчета переменной a

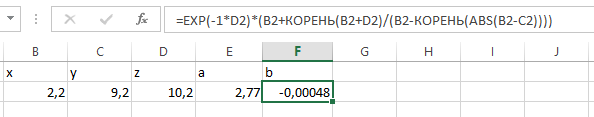
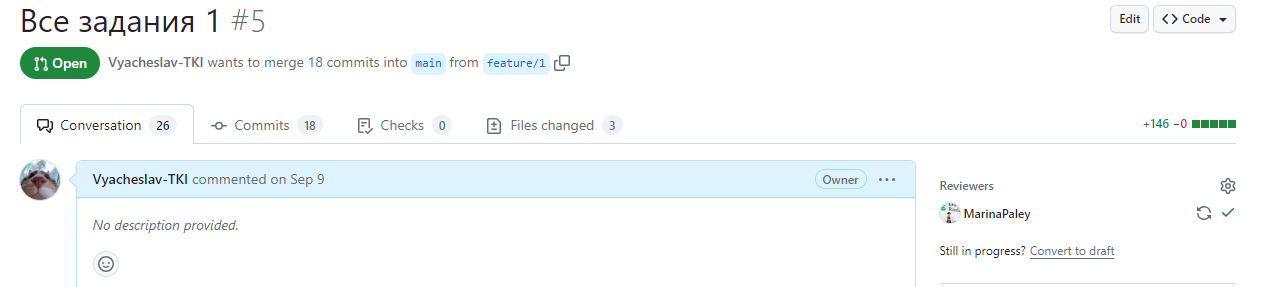


Рисунок 6 – Результат расчета переменной b

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



1. зАДАНИЕ 1.2
   1. Формулировка задания

Создать консольное приложение для решения задачи, представленной в таблице. (Таблица 1) 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



2.2 Блок-схема алгоритма

Блок-схема основного алгоритма представлена ниже (Рисунок 7). Блок-схемы функций расчета значений averange и geom\_mean представлены ниже (Рисунок 8), (Рисунок 9).

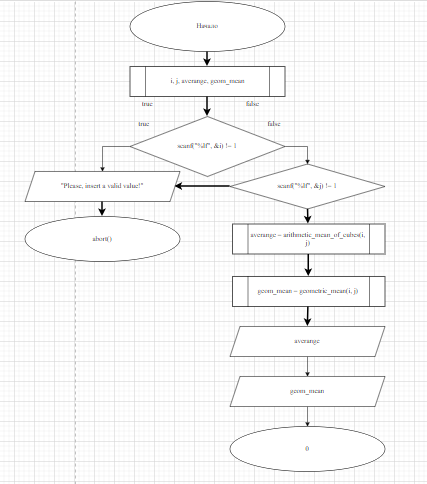


Рисунок 7 ­ Блок-схема основного алгоритма

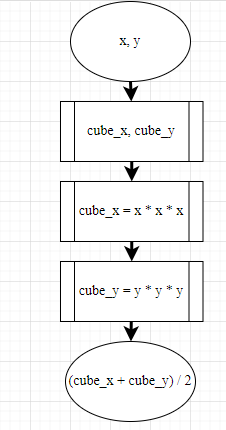


Рисунок 8 – Блок-схема расчёте среднего арифметического кубов чисел.

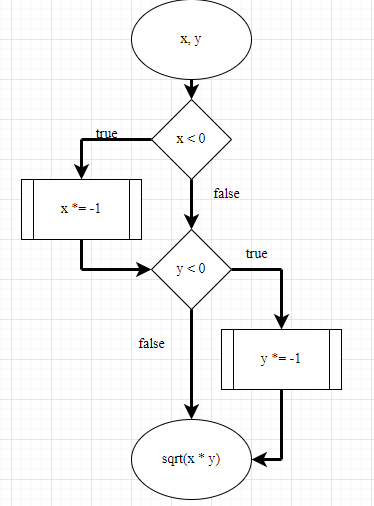


Рисунок 9 – Блок-схема вычисления среднего геометрического.

* 1. Текст программы на языке C

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

/\*\*

\* @brief Функция считает среднее арифметическое кубов заданных чисел

\* @param x Первое рациональное число

\* @param y Второе рациональное число

\*/

double arithmetic\_mean\_of\_cubes(double x, double y);

/\*\*

\* @brief Функция считает среднее геометрическое МОДУЛЕЙ чисел

\* @param x Первое рациональное число

\* @param y Второе рациональное число

\*/

double geometric\_mean(double x, double y);

int main(){

double i, j, average, geom\_mean;

puts("In the next strings insert just non-negative or just negative numbers\n");

if(scanf("%lf", &i) != 1){

puts("Please, insert a valid value!\n");

abort();}

if(scanf("%lf", &j) != 1){

puts("Please, insert a valid value!");

abort();}

average = arithmetic\_mean\_of\_cubes(i, j);

geom\_mean = geometric\_mean(i, j);

printf("The arithmetic mean of cubes of numbers is equal to: %.2f\n", average);

printf("The geometric mean of the numbers is: %.2f\n", geom\_mean);

return 0;

}

double arithmetic\_mean\_of\_cubes(double x, double y){

double cube\_x, cube\_y;

cube\_x = x \* x \* x;

cube\_y = y \* y \* y;

return (cube\_x + cube\_y) / 2;

}

double geometric\_mean(double x, double y){

if(x < 0){

x \*= -1;

}

if (y < 0){

y \*= -1;

}

return sqrt(x \* y);

}

* 1. Результаты выполнения программы

Результаты выполнения программы представлены ниже (Рисунок 10).

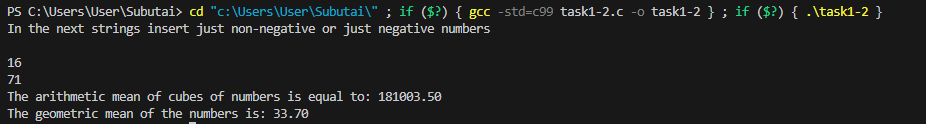


Рисунок 10 – Результаты выполнения программы

* 1. Выполнение тестовых примеров

В программе MS Excel выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 11, Рисунок 12).

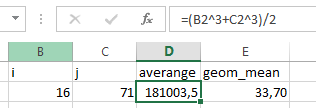


Рисунок 11 – Результат расчета переменной averange

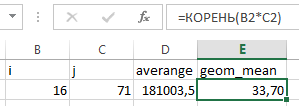
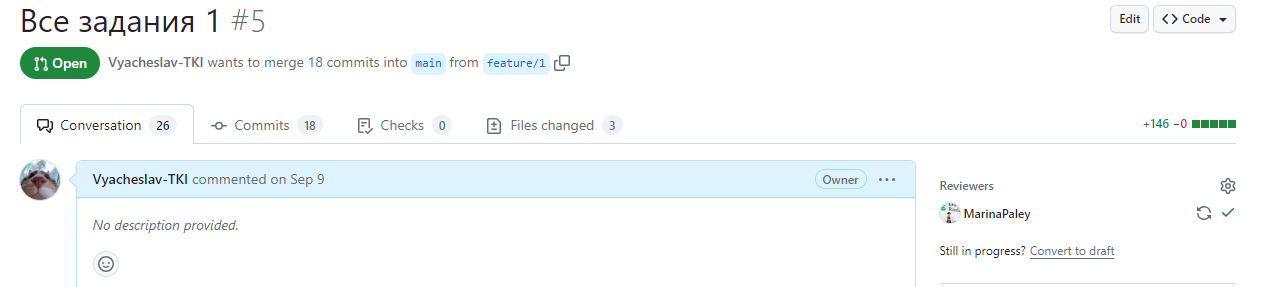


Рисунок 12 – Результат расчета переменной geom\_mean

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



1. зАДАНИЕ 1.3
   1. Формулировка задания

Создать консольное приложение для решения задачи, представленной в таблице. (Таблица 1). 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



* 1. Блок-схема алгоритма

Блок-схема основного алгоритма представлена ниже (Рисунок 13). Блок-схема функций расчета значения energy представлена ниже (Рисунок 14).

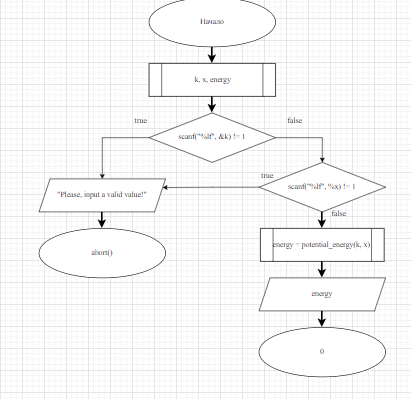


Рисунок 13 ­ Блок-схема основного алгоритма

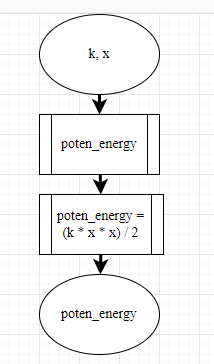


Рисунок 14 – Блок-схема используемой функции

* 1. Текст программы на языке C

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

/\*\*

\* @brief Функция рассчитывает потенциальную энергию

\* @param k - жесткость пружины,

\* @param x - растянутость пружины

\*/

double potential\_energy(double k, double x);

/\*\*

\* @brief Функция проверяет на положительность или нейтральность(число==0) числа

\* @param check\_param проверяемая переменная

\* @return 0 => Число отрицательное, 1 => Число положительное или равно нулю

\*/

int is\_num(double check\_param);

int main(){

double k, x, energy;

puts("Insert the k value first, then the x value, all values is non-negative number\n");

if(scanf("%lf", &k) != 1){

puts("Please input a valid value!\n");

abort();}

if(scanf("%lf", &x) != 1){

puts("Please input a valid value!\n");

abort();}

if(is\_num(k) + is\_num(x) < 2){

printf("%s", "Insert only pozitive numbers!\n");

abort();

}

energy = potential\_energy(k, x);

printf("Potential energy equals: %.2f", energy);

return 0;

}

double potential\_energy(double k, double x){

double poten\_energy;

poten\_energy = (k \* x \* x) / 2;

return poten\_energy;

}

int is\_num(double check\_param) {

if (check\_param >= 0.0f){

return 1;

}

return 0;

}

* 1. Результаты выполнения программы

Результаты выполнения программы представлены ниже (Рисунок 15).

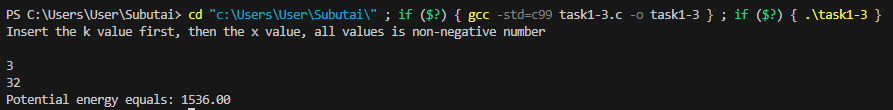


Рисунок 15 – Результат выполнения программы

* 1. Выполнение тестовых примеров

В программе MS Excel выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 16).

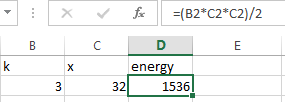


Рисунок 16 – Результат расчета переменной energy

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

