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 №2

По дисциплине «Языки программирования»

ВАРИАНТ 7

Выполнил: ст. гр. ТКИ - 141

Костин Роман Сергеевич

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

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

Москва 2024

Оглавление

[1 решение задачи 2.1 2](#_Toc156408412)

[1.1 Формулировка задачи 2](#_Toc156408413)

[1.2 Блок-схема алгоритма 3](#_Toc156408414)

[1.3 Код задания 2.2 4](#_Toc156408415)

[1.4 Решение различных текстовых задач на C 6](#_Toc156408416)

[1.5 Решение различных тестовых примеров в Excel 6](#_Toc156408417)

[1.6 Подтверждение Approve 7](#_Toc156408418)

[2 решение задания 2.2 8](#_Toc156408419)

[2.1 Формулировка задачи 8](#_Toc156408420)

[2.2 Блок-схема алгоритма 9](#_Toc156408421)

[2.3 Код для задания 2.2 10](#_Toc156408422)

[2.4 Решение различных тестовых примеров на C 12](#_Toc156408423)

[2.5 Решение различных тестовых примеров в Excel 12](#_Toc156408424)

[2.6 Подтверждение Approve 2.2 12](#_Toc156408425)

[3 решение задания 2.3 13](#_Toc156408426)

[3.1 Формулировка задачи 13](#_Toc156408427)

[3.2 Блок-схема алгоритма 14](#_Toc156408428)

[3.3 Код для задания 2.3 15](#_Toc156408429)

[3.4 Решение различных тестовых примеров на C 17](#_Toc156408430)

[3.5 Решение различных тестовых задач в Excel 17](#_Toc156408431)

[3.6 Подтверждение Approve 2.3 18](#_Toc156408432)

1. решение задачи 2.1
   1. Формулировка задачи

Создать консольное приложение для решения задачи, представленной в таблице, с использованием перечислимого типа. Выбор формулы вычисления зависит от пользователя. Данные для решения задачи так же вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

Таблица  – Условие задачи 2.1

|  |  |
| --- | --- |
| Вариант | Задача |
| 7 | Переменной А присвоить её значение, увеличенное в:   * N раз; * 2N раз; * 3N раз. |

* 1. Блок-схема алгоритма

Блок-схема алгоритма представлена ниже (Рисунок 1, Рисунок 2)

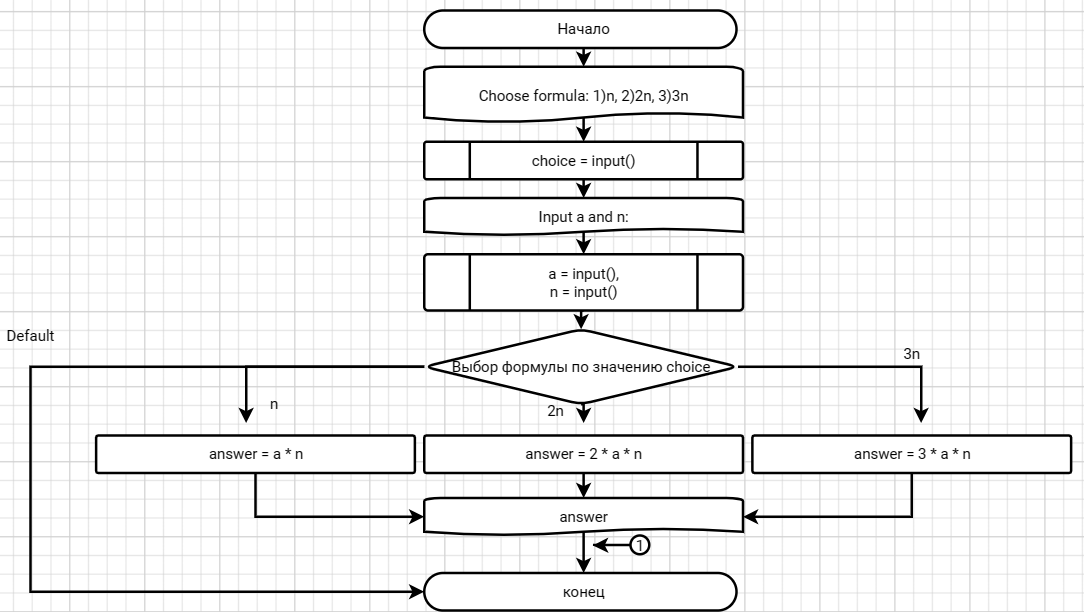


Рисунок  – Блок-схема функции main()

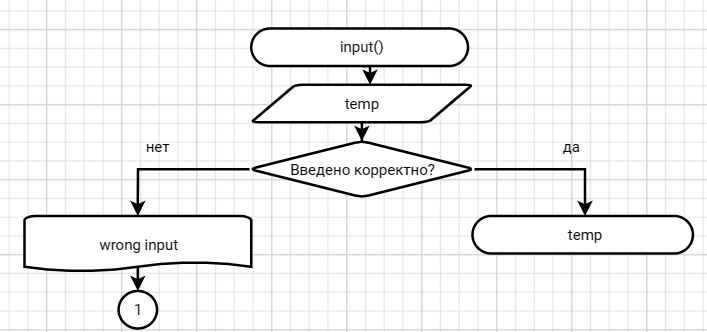


Рисунок  – Блок-схема функции input()

* 1. Код задания 2.2

#include <stdio.h>

#include <stdlib.h>

#include <errno.h>

/\*\*

\* @brief enum type used to choose formula

\*/

enum formula { N = 1, DBL\_N, TPL\_N };

/\*\*

\* @brief function used to get value from user

\* @return temp - result of user's input

\*/

double input();

/\*

\* @brief entry point into the programm

\* @return 0 if programm works correctly

\*/

int main()

{

printf\_s("%s", "choose formula 1) N; 2) 2N; 3) 3N ");

int choice = input();

printf\_s("input A = ");

double a = input();

printf\_s("input N = ");

double n = input();

enum formula equation = (enum formula)(choice);

switch (equation)

{

case N:

{

printf\_s("the answer is %lf", a \* n);

break;

}

case DBL\_N:

{

printf\_s("the answer is %lf", a \* n \* 2);

break;

}

case TPL\_N:

{

printf\_s("the answer is %lf", a \* n \* 3);

break;

}

default:

{

errno = EIO;

perror("Input error!");

return 1;

break;

}

}

return 0;

}

double input()

{

double temp;

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

if (res != 1)

{

errno = EIO;

perror("wrong input");

abort();

}

return temp;

}

* 1. Решение различных текстовых задач на C

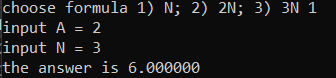


Рисунок  – Вывод программы 2.1 на C при корректном выборе первой формулы



Рисунок  – Вывод программы 2.1 на C при корректном выборе второй формулы

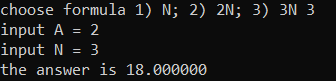


Рисунок  – Вывод программы на C при корректном выборе третьей формулы

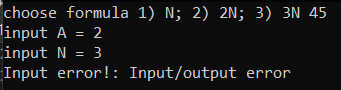


Рисунок  – Вывод программы на C при некорректном выборе формулы

* 1. Решение различных тестовых примеров в Excel



Рисунок  – Решение задачи 2.1 в Excel

* 1. Подтверждение Approve

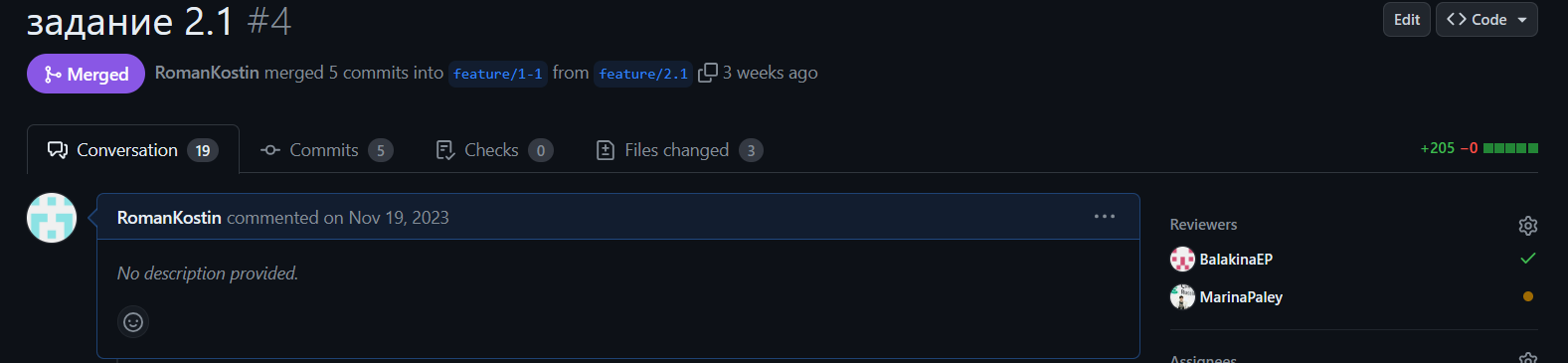


Рисунок  – Approve задачи 2.1

1. решение задания 2.2
   1. Формулировка задачи

Создать консольное приложение с организацией разветвлённого процесса для решения заданной в таблице функции. Данные для решения вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

Таблица  – Условие задачи 2.2

|  |  |  |
| --- | --- | --- |
| Вариант | Функция | Константы |
| 7 |  | = ввод пользователя |

* 1. Блок-схема алгоритма

Блок-схема алгоритма представлена ниже (Рисунок 9, Рисунок 10, Рисунок 11)

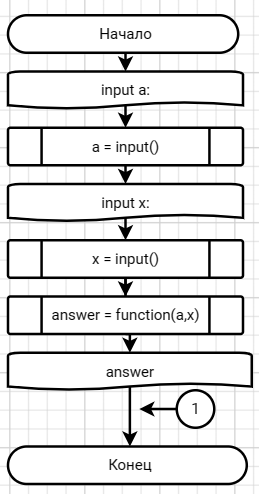


Рисунок  – Блок-схема функции main()

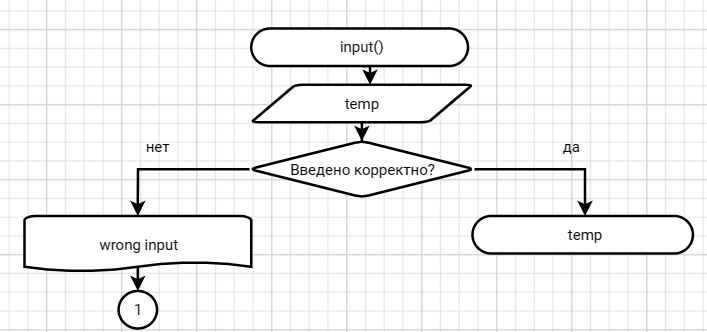


Рисунок  – Блок-схема функции input()

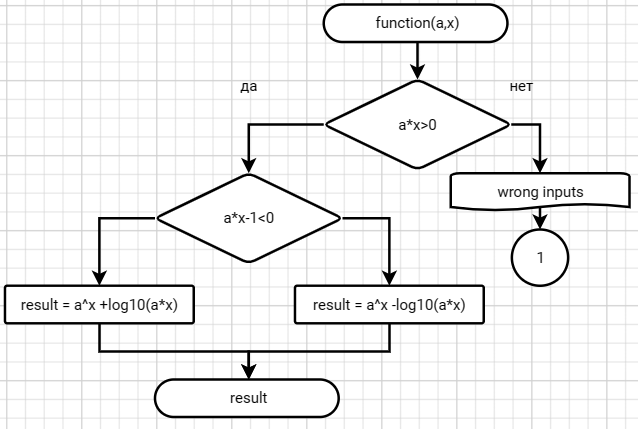


Рисунок  – Блок-схема функции function()

* 1. Код для задания 2.2

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <float.h>

#include <errno.h>

/\*\*

\*@brief function that finds answer to the equation. If a\*x is less than 1 it solves a^x - log10(a\*x) otherwise a^x + log10(a\*x);

\*@param a user's input

\*@param x user's input

\*@return answer to the equation

\*/

double function(double a, double x);

/\*\*

\* @brief function used to get value from user

\* @return result - result of user's input

\*/

double input();

/\*\*

\* @brief entrance point into the programm

\* @return 0 if programm works correctly

\*/

int main()

{

printf\_s("Input a = ");

double a = input();

printf\_s("Input x = ");

double x = input();

printf\_s("the answer is %lf ", function(a, x));

return 0;

}

double input()

{

double temp;

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

if (res != 1)

{

errno = EIO;

perror("wrong input");

abort();

}

return temp;

}

double function(double a, double x)

{

if (a \* x < DBL\_EPSILON)

{

errno = EIO;

perror("wrong input");

abort();

}

if (a \* x - 1 < DBL\_EPSILON)

{

return pow(a, x) - log10(a \* x);

}

else

{

return pow(a, x) + log10(a \* x);

}

}

* 1. Решение различных тестовых примеров на C

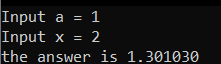


Рисунок  – Вывод программы 2.2 на C в случае a\*x-1>0 и при соблюдении ОДЗ

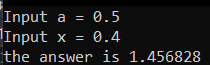


Рисунок  – Вывод программы 2.2 на C в случае a\*x-1<0 и при соблюдении ОДЗ

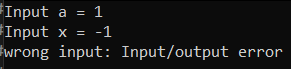


Рисунок  – Вывод программы 2.2 на C при нарушении условия ОДЗ a\*x<0

* 1. Решение различных тестовых примеров в Excel

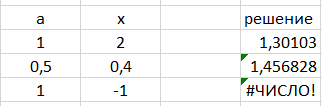


Рисунок  – Решение задания 2.2 в Excel

* 1. Подтверждение Approve 2.2

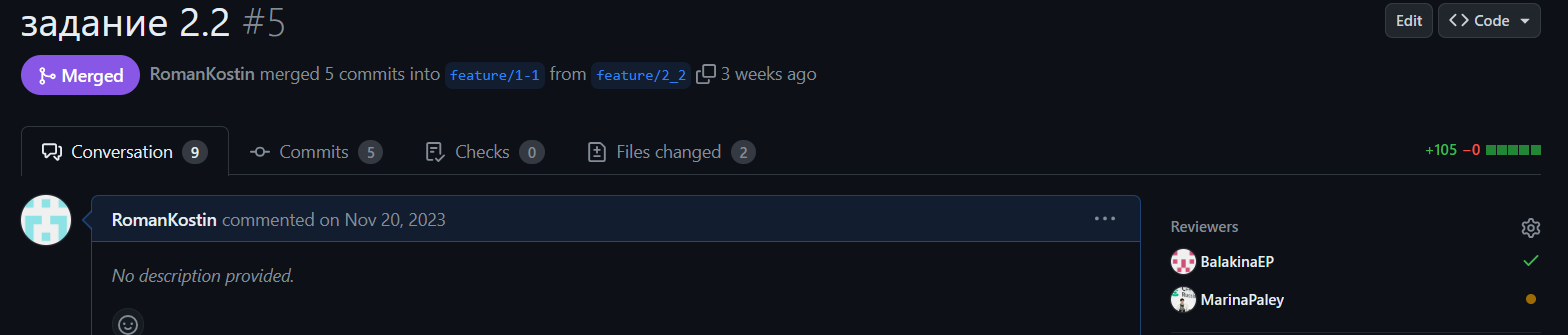


Рисунок  – Approve задачи 2.2

1. решение задания 2.3
   1. Формулировка задачи

Создать консольное приложение с организацией разветвлённого процесса для решения задачи из таблицы. Данные для решения вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

Таблица  – Условие задачи 2.3

|  |  |
| --- | --- |
| Вариант | Задача |
| 7 | По заданным трем числам определить, является ли сумма каких-либо из них положительной. |

* 1. Блок-схема алгоритма

Блок-схема алгоритма представлена ниже (Рисунок 17, Рисунок 18, Рисунок 19)

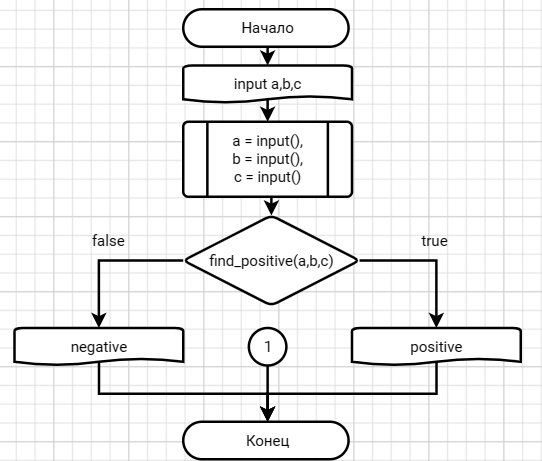


Рисунок  – Блок-схема функции main()

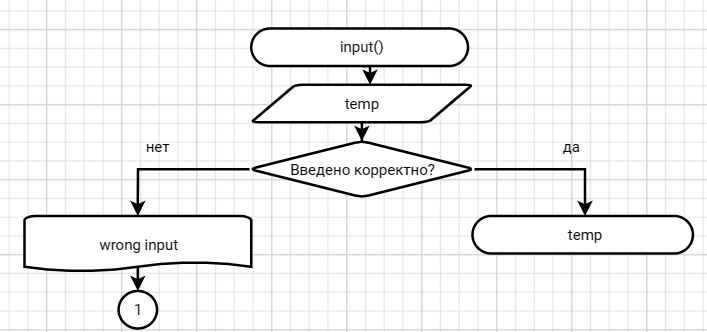


Рисунок  – Блок-схема функции input()

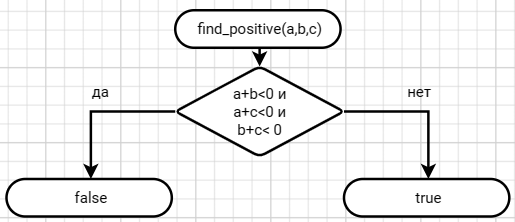


Рисунок  – Блок-схема функции find\_positive()

* 1. Код для задания 2.3

#include <stdio.h>

#include <float.h>

#include <errno.h>

#include <stdbool.h>

/\*\*

\* @brief function checks if there is any positive combination. If sum of any of them is more than DBL\_EPSILON it prints positive, otherwise no positive

\* @param a user's input

\* @param b user's input

\* @param c user's input

\* @return 0 if False; 1 if True

\*/

bool find\_positive(double a, double b, double c);

/\*\*

\* @brief function used to get value from user

\* @return temp - result of user's input

\*/

double input();

/\*\*

\* @brief entrance point into the programm

\* @return 0 if programm works correctly

\*/

int main()

{

printf\_s("input a, b ,c ");

double a = input(),

b = input(),

c = input();

if (find\_positive(a, b, c))

{

printf\_s("positive");

}

else

{

printf\_s("negative");

}

return 0;

}

bool find\_positive(double a, double b, double c)

{

return (a + b > DBL\_EPSILON || a + c > DBL\_EPSILON || b + c > DBL\_EPSILON);

}

double input()

{

double temp;

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

if (res != 1)

{

errno = EIO;

perror("wrong input");

abort();

}

return temp;

}

* 1. Решение различных тестовых примеров на C



Рисунок  – Вывод программы 2.3 на C при вводе только положительных значений



Рисунок  – Вывод программы 2.3 на C при вводе одного отрицательного значения



Рисунок  – Вывод программы 2.3 на C при вводе двух отрицательных значений, одна из сумм которых больше нуля



Рисунок  – Вывод программы 2.3 на C в случае, когда нет ни одной положительной пары чисел



Рисунок  – Вывод программы 2.3 на C при некорректном вводе значений

* 1. Решение различных тестовых задач в Excel

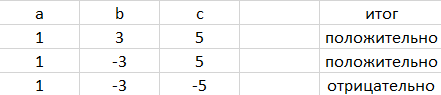


Рисунок  – решение задания 2.3 в Excel

* 1. Подтверждение Approve 2.3

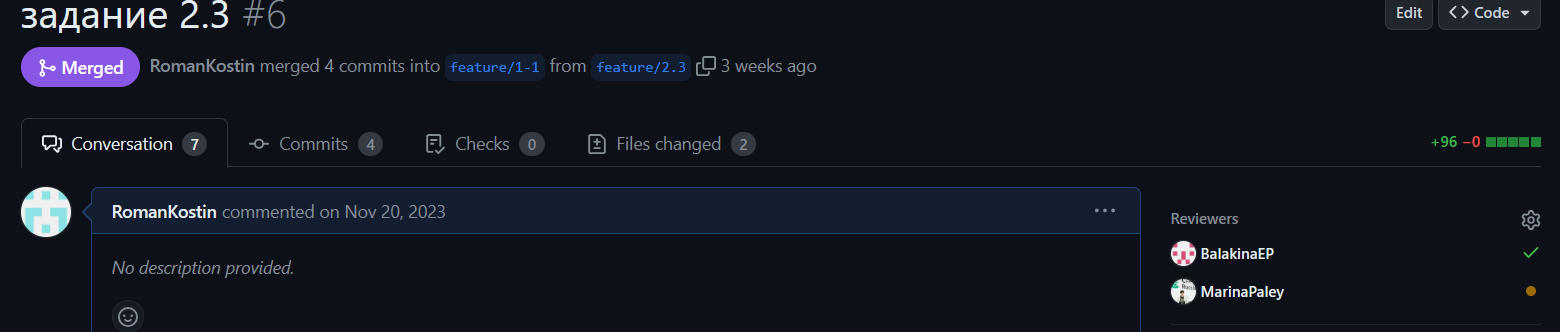


Рисунок  – Approve задания 2.3