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 2

Выполнил: ст. гр. ТКИ - 142

Бахров Кирилл Сергеевич

Проверил: к.т.н., доц. Васильева М. А Проверил: к.т.н., доц. Балакина Е. П.

Москва 2023

**Оглавление**

**Решение задачи 2.13**

Формулировка задания3

Блок-схема алгоритма4-5

Текст программы на языке C6-7

Результат выполнения программы8

Выполнение тестовых примеров9

Отметка о выполнении задания в веб-хостинге системы контроля версий10

**Решение задачи 2.211**

Формулировка задания11

Блок-схема алгоритма12-13

Текст программы на языке C14-15

Результат выполнения программы16

Выполнение тестовых примеров17

Отметка о выполнении задания в веб-хостинге системы контроля версий18

**Решение задачи 2.3**19

Формулировка задания19

Блок-схема алгоритма20

Текст программы на языке C21-22

Результат выполнения программы23

Выполнение тестовых примеров24

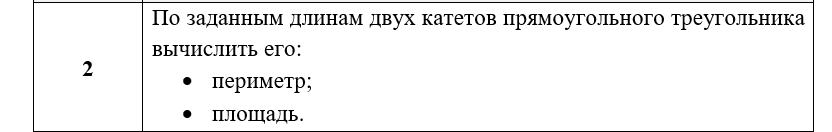
Отметка о выполнении задания в веб-хостинге системы контроля версий25

1. РЕШЕНИЕ ЗАДАЧИ 2.1

1.1 ФОРМУЛИРОВКА ЗАДАНИЯ

Создать консольное приложение для решения задачи, представленной в таблице, с использованием перечислимого типа. (Таблица 1) 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 1 – Исходные данные



1.2 БЛОК-СХЕМА АЛГОРИТМА

Блок-схема основного алгоритма представлена ниже (Рисунок 1). Блок-схемы функций представлены ниже (Рисунки 2-3).

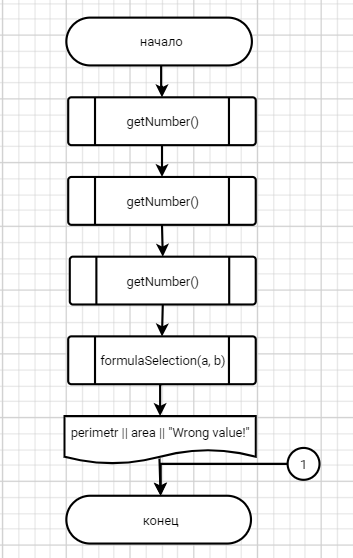


Рисунок 1 ­ Блок-схема основного алгоритма

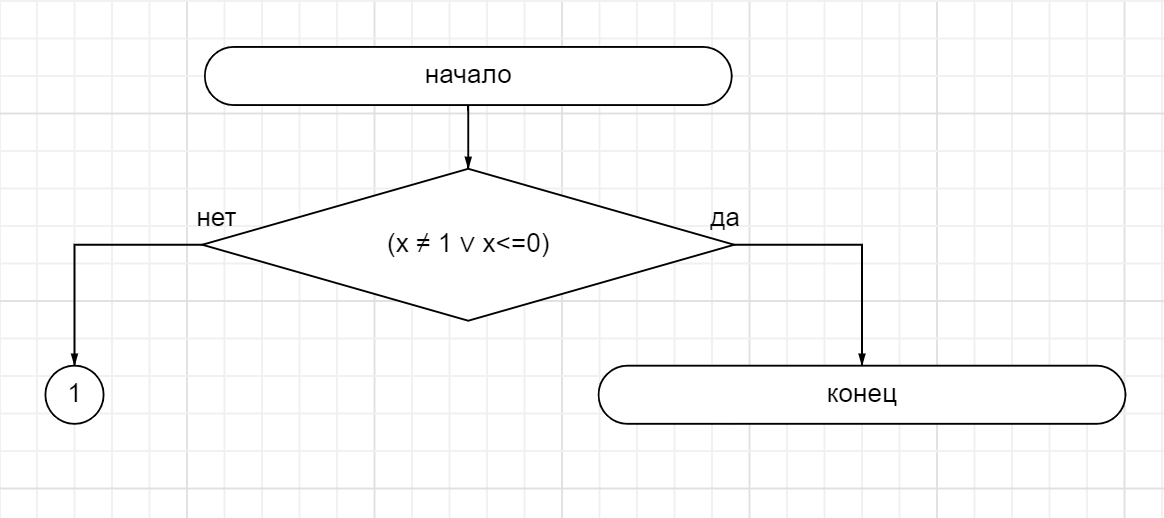


Рисунок 2 – Блок-схема используемых функций

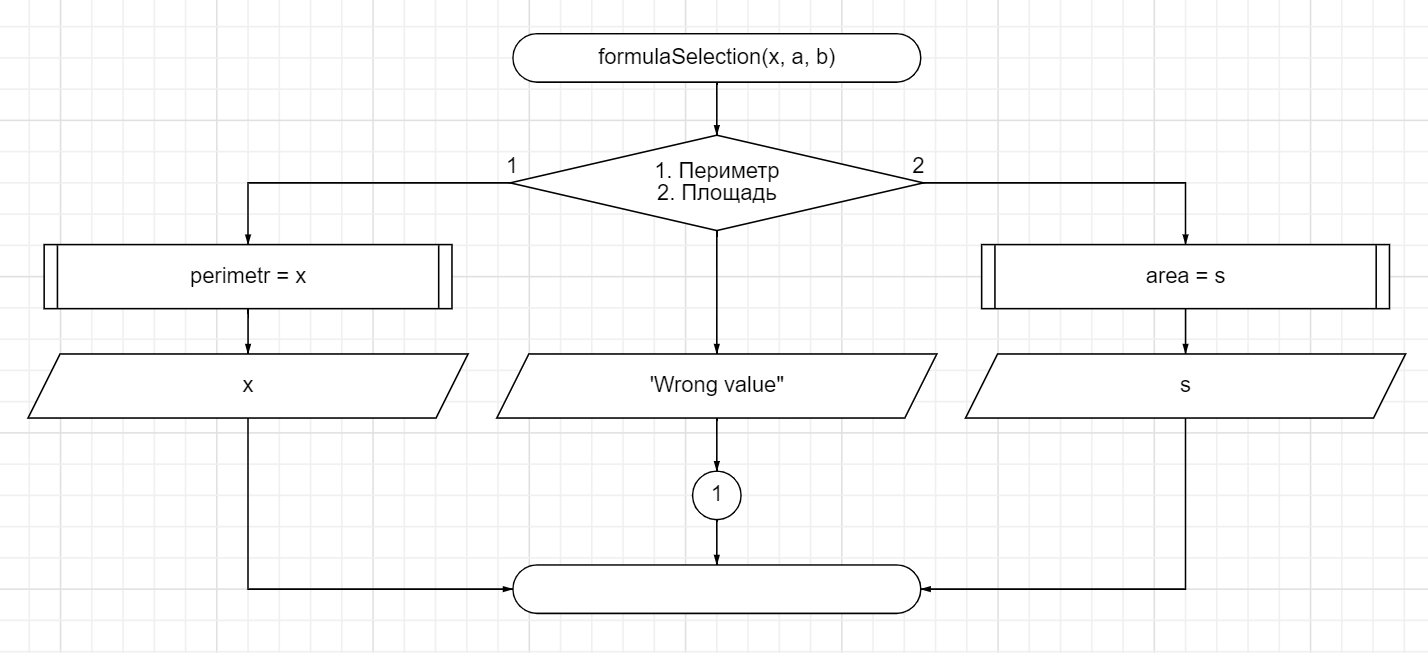


Рисунок 3 – Блок-схема используемых функций

1.3 ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ C

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

/\*\*

\* @brief считывает и проверяет ввод переменной типа float

\* @return возвращает считанное значение

\*/

float getNumber();

/\*\*

\* @brief вычисляет выбранную пользователем величину

\* @param x выбранная формула

\* @param a длина катета a

\* @param b длина катета b

\* @return возвращает посчитанную величину

\*/

float formulaSelection(int x, float a, float b);

enum Values

{

perimetrTriangle = 1,

areaTriangle = 2,

};

/\*\*

\* @brief точка входа в программу

\* @return возвращает 0, если программа выполнена корректно

\*/

int main()

{

printf("Enter the length of the catheter a: \n");

float a = getNumber();

printf("Enter the length of the catheter b: \n");

float b = getNumber();

printf("Please write down what you want to find: the perimeter of the triangle or the area\n");

printf("If you want to find the perimeter of the triangle press 1: \n");

printf("If you want to find the area of the triangle press 2: \n");

int x = getNumber();

formulaSelection(x, a, b);

return 0;

}

float getNumber()

{

float x;

if (scanf("%f", &x) !=1 || x <= 0)

{

puts("Wrong value\n");

abort();

}

return x;

}

float formulaSelection(int x, float a, float b){

enum Values perimetr = perimetrTriangle, area = areaTriangle;

if(x == perimetr){

float c = pow( (pow(a, 2) + pow(b, 2) ), 0.5 );

float x = (a + b + c );

printf("The perimeter of the triangle is equal to: %f\n", x);

}

else if(x == area){

float c = pow( (pow(a, 2) + pow(b, 2) ), 0.5 );

float p = (a + b + c)/2;

float s = ( pow( p \* (p-a) \* (p-b) \* (p-c), 0.5 ) );

printf("The area of the triangle is equal to: %f\n", s);

}

else{

printf("Wrong value!");

}

}

}

1.4 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Результаты выполнения программы представлены ниже (Рисунок 3).

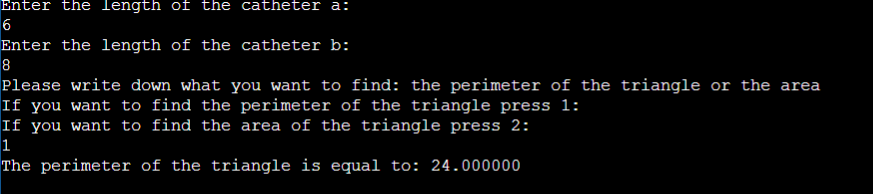


Рисунок 4 – Результаты выполнения программы

1.5 ВЫПОЛНЕНИЕ ТЕСТОВЫХ ПРИМЕРОВ

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

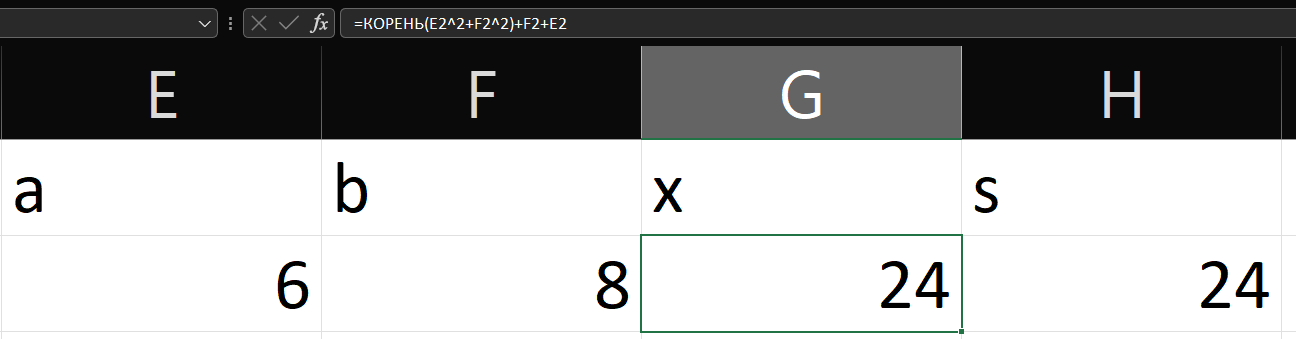


Рисунок 5 – Результат расчета переменной x

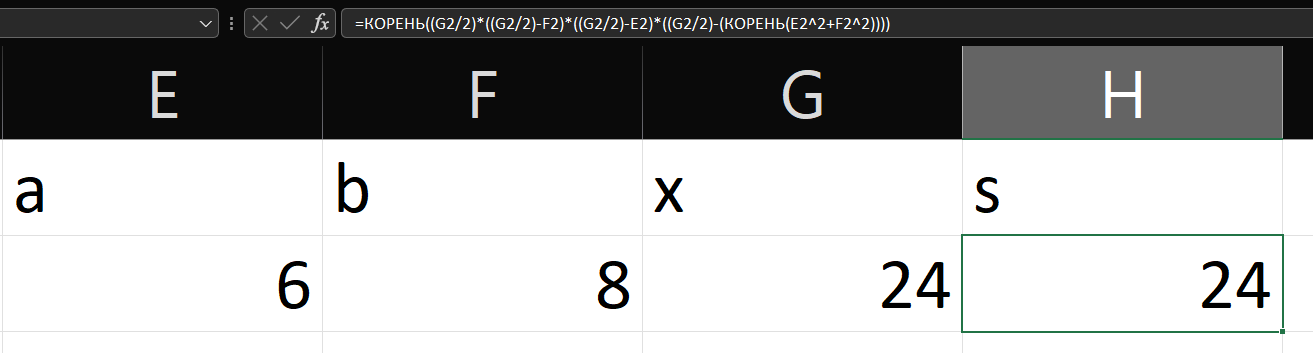


Рисунок 6 – Результат расчета переменной s

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

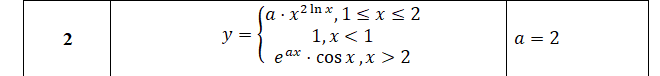
Рисунок 7 – Отметка о выполнении задания в веб-хостинге системы контроля версий

1. РЕШЕНИЕ ЗАДАЧИ 2.2

2.1 ФОРМУЛИРОВКА ЗАДАНИЯ

Создать консольное приложение с организацией разветвлённого процесса для решения заданной в таблице функции. (Таблица 2) Данные для решения вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

Таблица 2 – Исходные данные



2.2 Блок-схема алгоритма

Блок-схема основного алгоритма представлена ниже (Рисунок 8). Блок-схемы функций расчета представлены ниже (Рисунки 9 - 10).

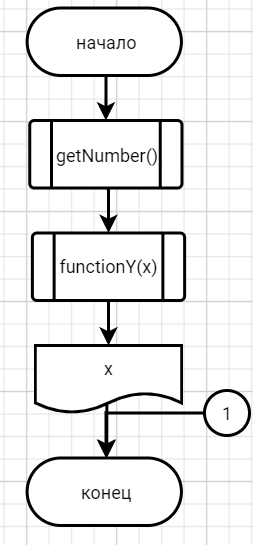


Рисунок 8 ­ Блок-схема основного алгоритма

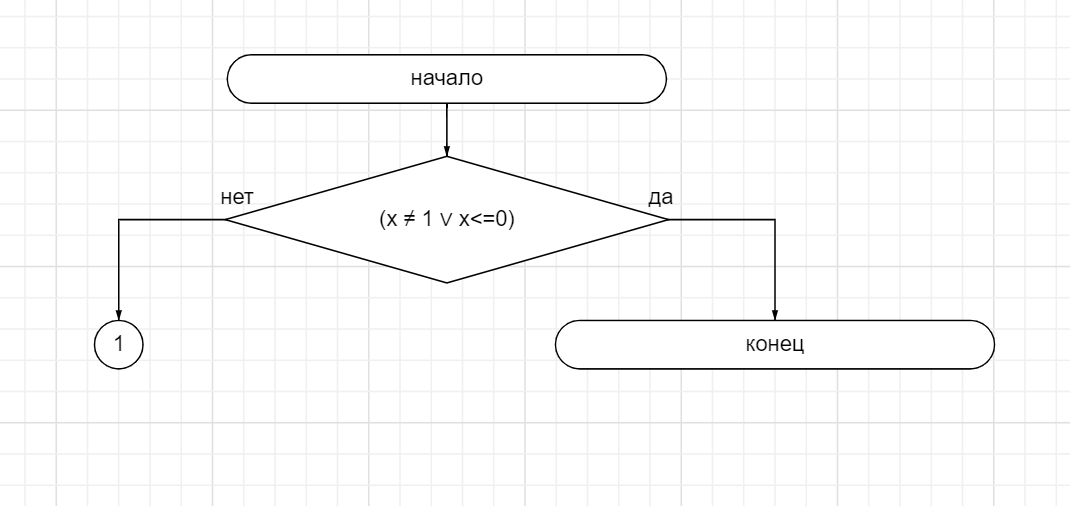


Рисунок 9 – Блок-схема используемых функций

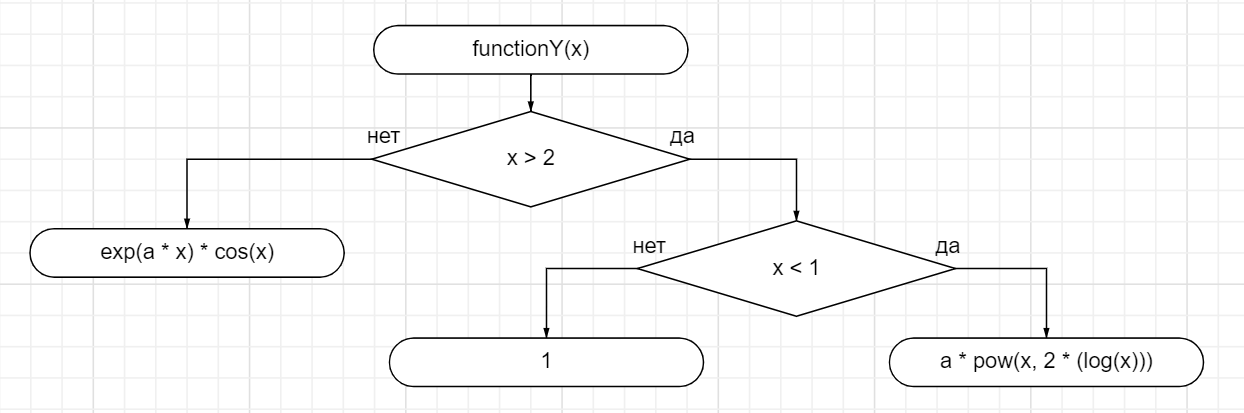


Рисунок 10 – Блок-схема используемых функций

2.3 Текст программы на языке C

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <float.h>

/\*\*

\* @brief считывает и проверяет ввод переменной типа float

\* @return возвращает считанное значение

\*/

float getNumber();

/\*\*

\* @brief решает функцию

\* @param x значение, которое вводит пользователь для расчета функции

\* @return выводит значение посчитанной функции

\*/

float functionY(const float x);

int main()

{

printf("Enter the X value ");

float x = getNumber();

printf("y = %f", functionY(x) );

return 0;

}

float getNumber()

{

float x;

if (scanf("%f", &x) !=1 || x <= 0)

{

puts("Wrong value");

abort();

}

return x;

}

float functionY(const float x)

{

float a = 2, e = 2.71828;

if ( 1 <= x + DBL\_EPSILON && x + DBL\_EPSILON <= 2)

{

return ( a \* (pow(x, 2 \* (log(x) ) ) ) );

}

else if (x + DBL\_EPSILON < 1)

{

return 1;

}

else if (x + DBL\_EPSILON > 2)

{

return (( exp (a \* x) \* cos(x) ));

}

}

2.4 Результаты выполнения программы

Результаты выполнения программы представлены ниже (Рисунок 11).

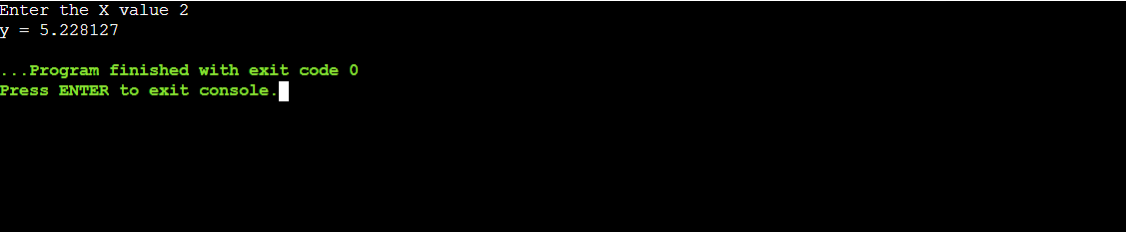


Рисунок 11 - Результаты выполнения программы

2.5 Выполнение тестовых примеров

В программе MS Excel выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 12, Рисунок 13, Рисунок 14).

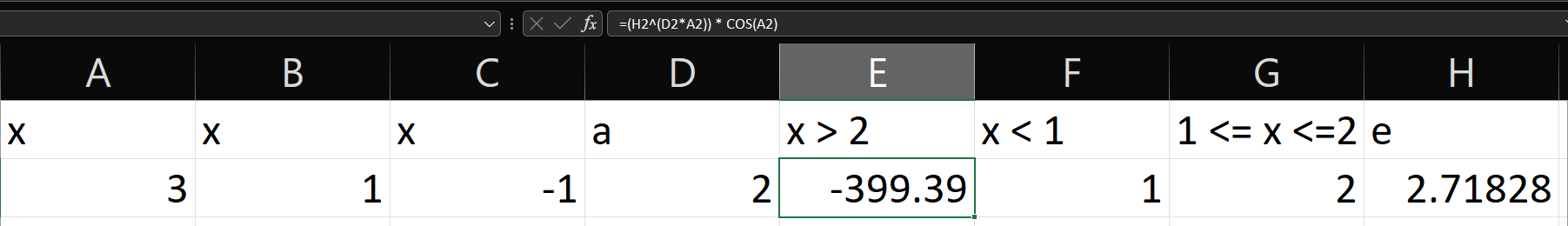


Рисунок 12 – Результат расчета при x>2

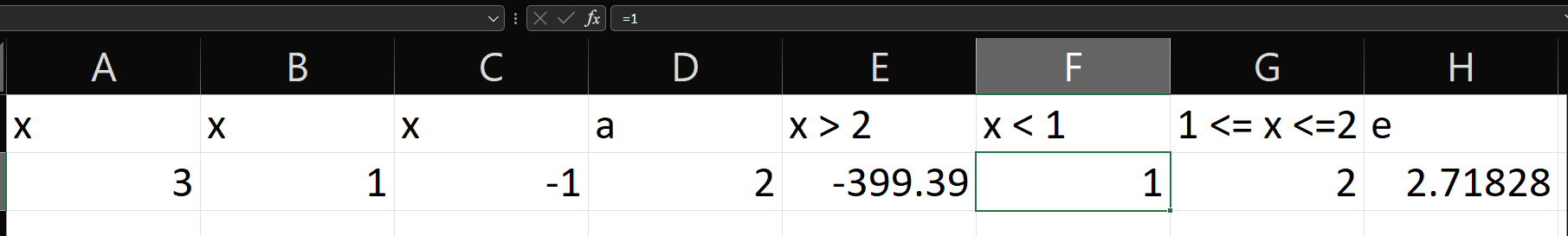


Рисунок 13 – Результат расчета при x<1

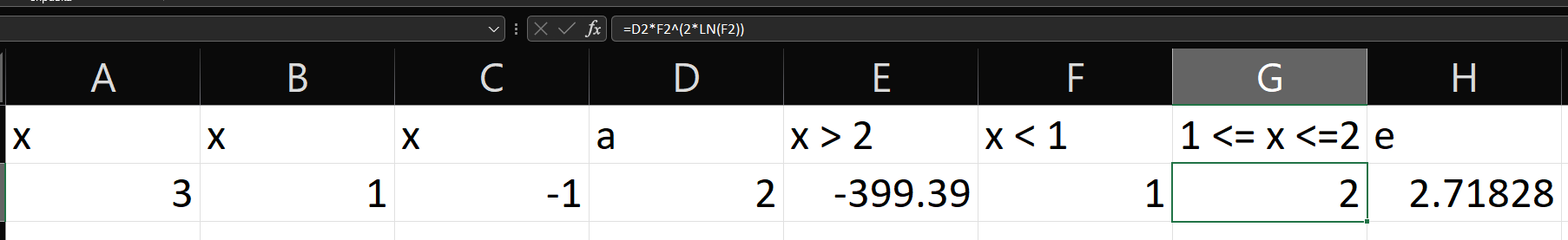


Рисунок 14 – Результат расчета при 1 <= x <= 2

2.6 Отметка о выполнении задания в веб-хостинге системы контроля версий

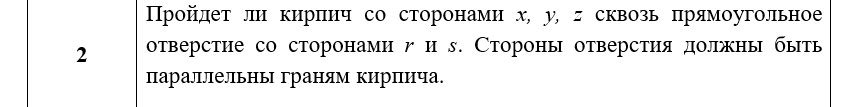


Рисунок 15 – Отметка о выполнении задания в веб-хостинге системы контроля версий

РЕШЕНИЕ ЗАДАЧИ 2.3  
3.1 Формулировка задания

Создать консольное приложение с организацией разветвлённого процесса для решения задачи из таблицы. (Таблица 3) Данные для решения вводит пользователь. Ввод необходимо проверять на правильность. Все результаты вывести на экран. Дополнить свой отчёт блок-схемой алгоритма.

Таблица 3 – Исходные данные



3.2 Блок-схема алгоритма

Блок-схема основного алгоритма представлена ниже (Рисунок 16). Блок-схемы используемых функций представлены ниже (Рисунок 17).

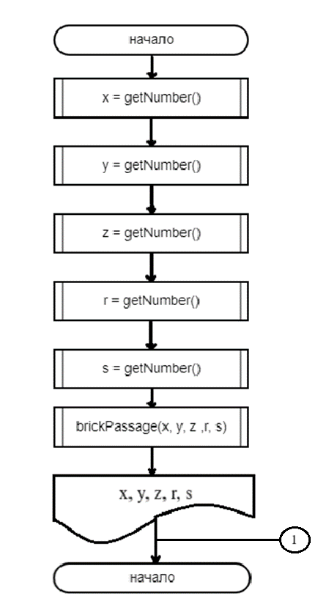


Рисунок 16 ­ Блок-схема основного алгоритма

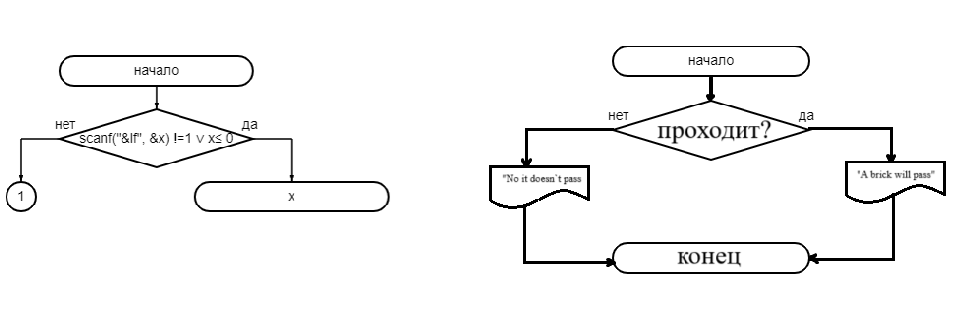


Рисунок 17 – Блок-схема используемых функций

3.3 Текст программы на языке C

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

/\*\*

\* @brief считывает и проверяет ввод переменной типа float

\* @return возвращает считанное значение

\*/

float getNumber();

/\*\*

\* @brief определяет пройдет ли кирпич в отверстие

\* @param x сторона кирпича

\* @param y сторона кирпича

\* @param z сторона кирпича

\* @param r сторона прямоугольного отверстия

\* @param s сторона прямоугольного отверстия

\* @return возвращает результат проверки

\*/

char brickPassage(float r, float s, float x, float y, float z);

/\*\*

\* @brief точка входа в программу

\* @return возвращает 0, если программа выполнена корректно

\*/

int main()

{

printf("The sides of the brick are equal: ");

float x = getNumber();

float y = getNumber();

float z = getNumber();

printf("Dimensions of the rectangular hole: ");

float r = getNumber();

float s = getNumber();

printf("%c", brickPassage(x,y,z,r,s));

return 0;

}

float getNumber()

{

float v;

if (scanf("%f", &v) !=1 || v <= 0)

{

puts("Wrong value");

abort();

}

return v;

}

char brickPassage(float r, float s, float x, float y, float z)

{

if ( ( ( ( r >= x ) && ( ( s >= y ) ||( s >= z ) ) ) || ( ( r >= y ) && ( ( s >= x ) ||( s >= z ) ) ) || ( ( r >= z ) && ( ( s >= x ) ||( s >= y ) ) ) ) ==1){

printf("A brick with such dimensions will pass through the hole");

}

else{

printf("No it doesn't pass ");

}

}

3.4 Результаты выполнения программы

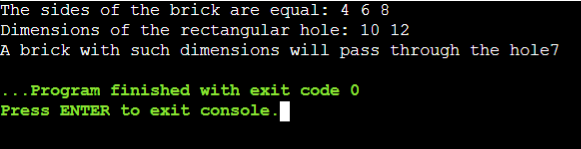


Рисунок 18 - Результаты выполнения программы

3.5 Выполнение тестовых примеров

В программе MS Excel выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 19).

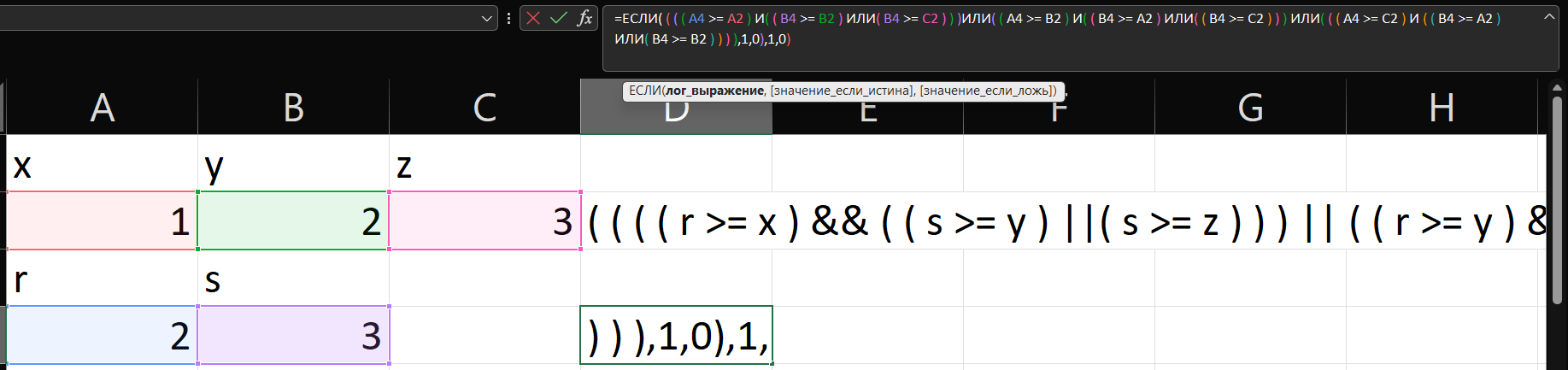


Рисунок 19 – Результат расчета функции

3.6 Отметка о выполнении задания в веб-хостинге системы контроля версий



Рисунок 20 – Отметка о выполнении задания в веб-хостинге системы контроля версий