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 12

Выполнил: ст. гр. ТКИ-142

Наседкин Тимофей Филиппович

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

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

Москва 2023

**ОГЛАВЛЕНИЕ**

**1.Решение задачи 2-1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.3**

**1.1. Формулировка задачи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.3**

**1.2. Блок-схема алгоритма\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.4-5**

**1.3. Код программы на языке C\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.6-7**

**1.4. Результаты выполнения программы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.8**

**1.5. Выполнение тестовых примеров\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.9**

**1.6. Отметка о выполнении задания в веб-хостинге системы контроля версий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.10**

**2.Решение задачи 2-2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.11**

**2.1. Формулировка задачи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.11**

**2.2. Блок-схема алгоритма\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.12-13**

**2.3. Код программы на языке C\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.14-15**

**2.4. Результаты выполнения программы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.16**

**2.5. Выполнение тестовых примеров \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.17**

**2.6. Отметка о выполнении задания в веб-хостинге системы контроля версий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_стр.18**

* 1. **Решение задачи 2-1**
  2. **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 1 – Условие задачи 2-1

| Вариант | Задачи |  |
| --- | --- | --- |
| 12 | Вывести числа *a*, *b*, *c* на экран в порядке:   * возрастания; * убывания. |  |

* 1. **Блок-схема алгоритма**

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

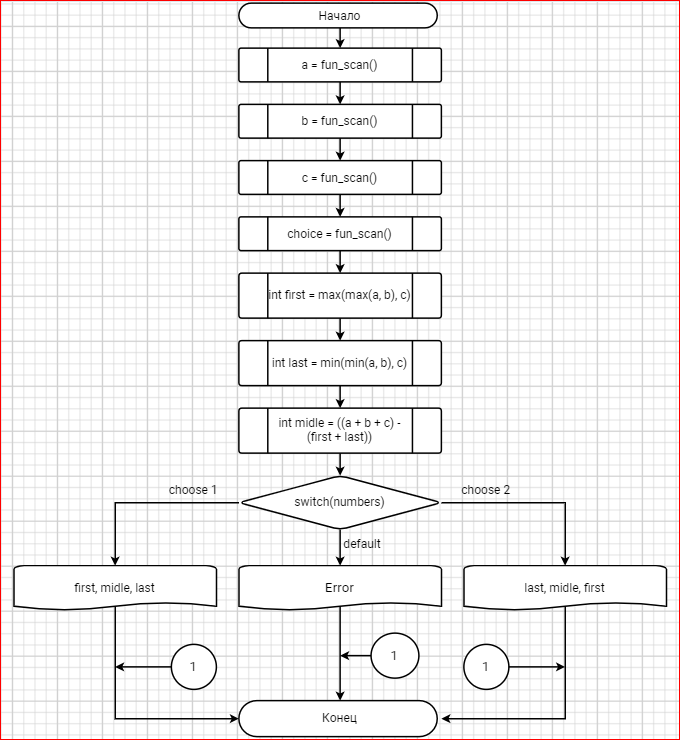


Рисунок 1 - Блок-схема основного алгоритма

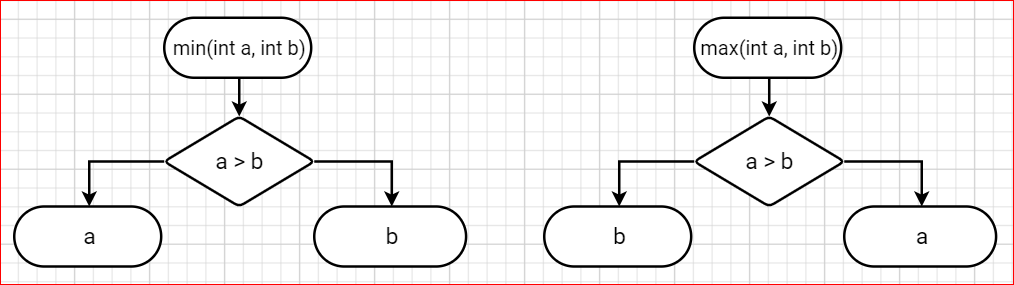


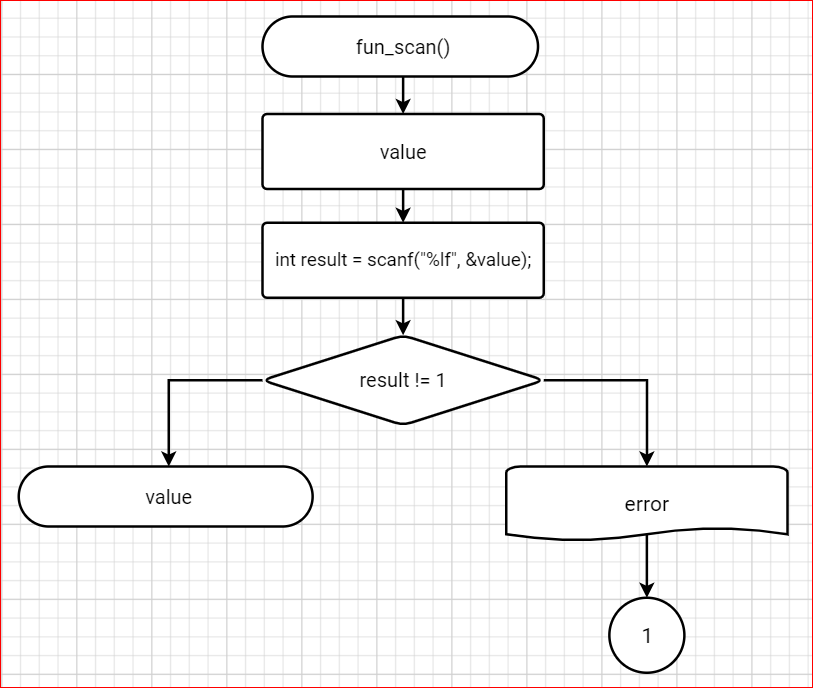
Рисунок 2 – Блок-схемы функций нахождения наибольшего и наименьшего значений двух чисел.

Рисунок 3 - Блок схема функции fun\_scan().

* 1. **Код программы на языке C**

#include <stdio.h>

#include <stdlib.h>

#include <errno.h>

/\*\*

\* @brief Структура хранит константы, указывающие выбор функции пользователем

\* @brief choose1 хранит значение выбора вывода чисел в порядке убывания

\* @brief choose2 хранит значение выбора вывода чисел в порядке возрастания

\*/

enum formula {choose1 = 1, choose2 = 2};

/\*\*

\* @brief Функция проверки ввода на число

\* @return Результат проверки

\*/

double fun\_scan();

/\*\*

\* @brief Функция нахождения минимума двух вводимых чисел

\* @param a, b значения вводимых параметров

\* @return Результат функции

\*/

int min(int a, int b);

/\*\*

\* @brief Функция нахождения максимума двух вводимых чисел

\* @param a, b значения вводимых параметров

\* @return Результат функции

\*/

int max(int a, int b);

/\*\*

\* @brief Вводимое значение для входа в программу

\* @return Вывод 0, если программа неверна. Иначе вывод 1

\*/

int main()

{

int a = fun\_scan();

int b = fun\_scan();

int c = fun\_scan();

int choice = fun\_scan();

int first = max(max(a, b), c);

int last = min(min(a, b), c);

int midle = ((a + b + c) - (first + last));

enum formula numbers = (enum formula)(choice);

switch(numbers)

{

case choose1:

{

printf("%d %d %d ", first, midle, last);

}

case choose2:

{

printf("%d %d %d ", last, midle, first);

}

default:

{

errno = EIO;

perror("error");

return 1;

}

}

return 0;

}

int min(int a, int b)

{

if (a > b)

{

return b;

}

return a;

}

int max(int a, int b)

{

if (a > b)

{

return a;

}

return b;

}

double fun\_scan()

{

double value;

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

if (result != 1)

{

printf("error io");

abort();

}

return value;

}

* 1. **Результаты выполнения программы**

Результаты выполнения программы представлены ниже (Рисунок 4).

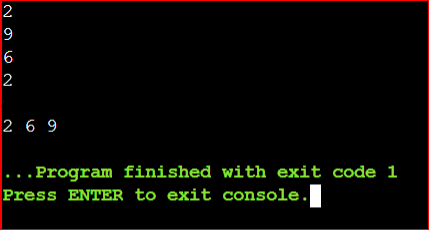


Рисунок 4 – Результаты выполнения программы

* 1. **Выполнение тестовых примеров**

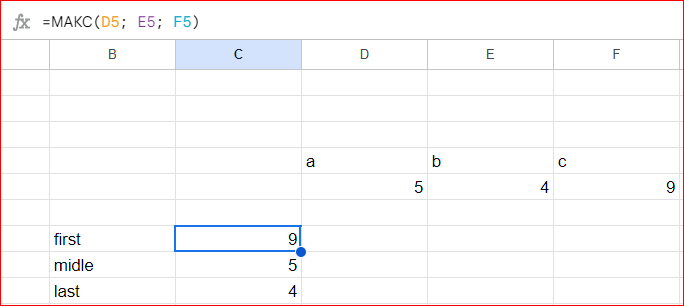
В программе MS Excel выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 5, Рисунок 6, Рисунок 7).

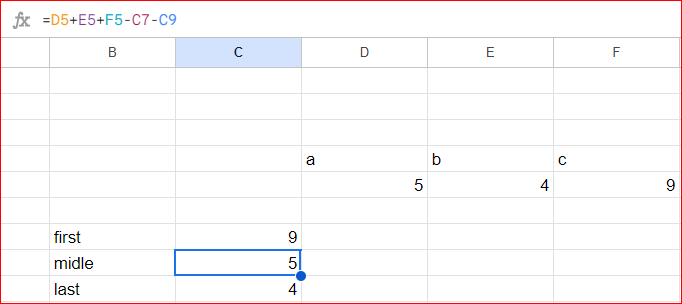
Рисунок 5 - Результат нахождения максимального элемента.

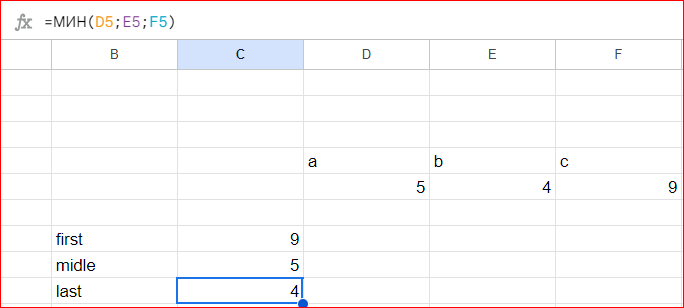
Рисунок 6 - Результат нахождения среднего элемента.

Рисунок 7 - Результат нахождения минимального элемента.

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

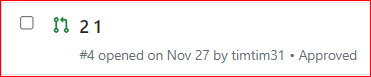


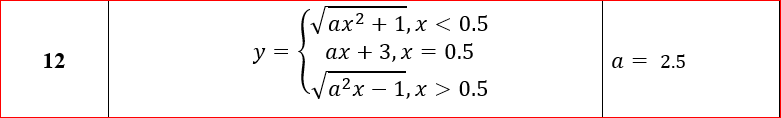
Рисунок 8 – Approve задачи 2-1

1. **Решение задачи 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.

Таблца 2 – Условие задачи 2-2

| Вариант | Функция | Константы |
| --- | --- | --- |



1. **Блок-схема алгоритма**

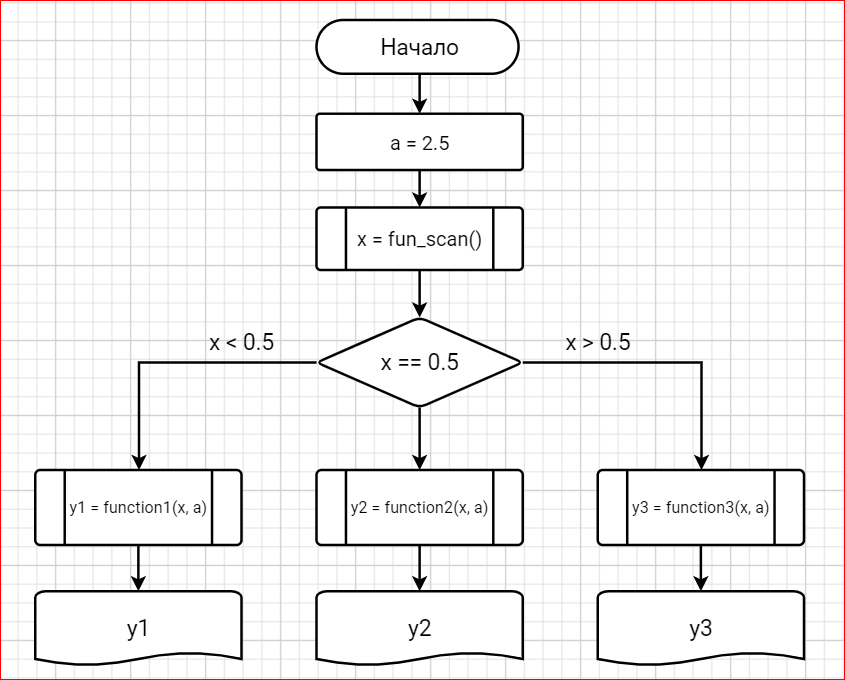
Блок-схема основного алгоритма представлена ниже (Рисунок 1). Блок-схемы дополнительных функций представлены ниже (Рисунок 2, Рисунок 3).

Рисунок 1 - Блок-схема основного алгоритма

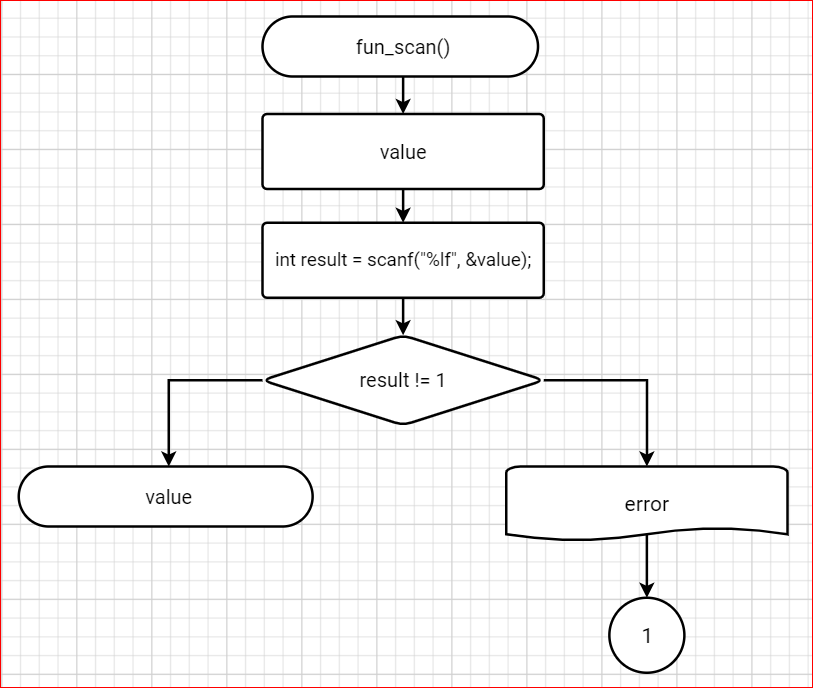


Рисунок 2 - Блок-схема функции fun\_scan()

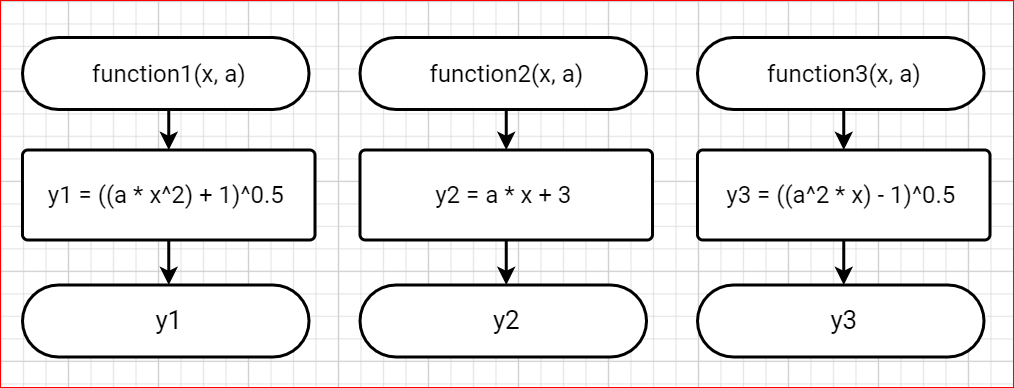


Рисунок 3- Блок-схемы функций function()

1. **Код программы на языке C**

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <float.h>

/\*\*

\* @brief Функция расчета формулы

\* @param x, a значения вводимого параметра и константы

\* @return Вывод значения вычислений

\*/

float function1(float x, float a);

/\*\*

\* @brief Функция расчета формулы

\* @param x, a значения вводимого параметра и константы

\* @return Вывод значения вычислений

\*/

float function2(float x, float a);

/\*\*

\* @brief Функция расчета формулы

\* @param x, a значения вводимого параметра и константы

\* @return Вывод значения вычислений

\*/

float function3(float x, float a);

/\*\*

\* @brief Функция проверки ввода на число

\* @return Результат проверки

\*/

double fun\_scan();

/\*\*

\* @brief Вводимое значение для входа в программу

\* @return Вывод 0, если программа неверна. Иначе вывод 1

\*/

int main()

{

const float a = 2.5;

float x = fun\_scan();

if (x - 0.5 < DBL\_EPSILON)

{

printf("y=%f\n", function1(x, a));

}

else

{

if (x - 0.5 == DBL\_EPSILON)

{

printf("y=%f\n", function2(x, a));

}

else

{

printf("y=%f\n", function3(x, a));

}

}

return 0;

}

float function1(float x, float a)

{

return (pow((a \* pow(x, 2) + 1), 0.5));

}

float function2(float x, float a)

{

return (a \* x + 3);

}

float function3(float x, float a)

{

return (pow((a \* pow(x, 2) - 1), 0.5));

}

double fun\_scan()

{

double value;

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

if (result != 1)

{

printf\_s("error io");

abort();

}

return value;

}

* 1. **Результаты выполнения программы**

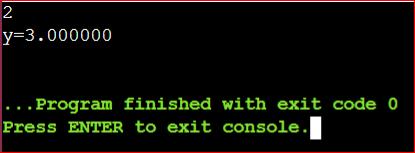
Результаты выполнения программы представлены ниже (Рисунок 4)

Рисунок 4 - результаты выполнения программы

* 1. **Выполнение тестовых примеров**

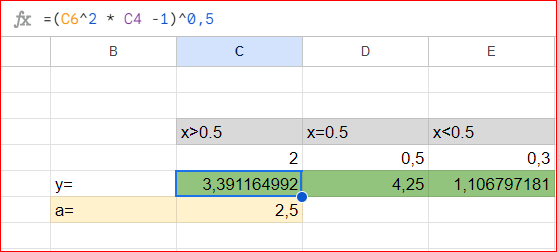
В программе MS Excel выполнены тестовые примеры. Результаты их выполнения представлены ниже (Рисунок 5), (Рисунок 6), (Рисунок 7).

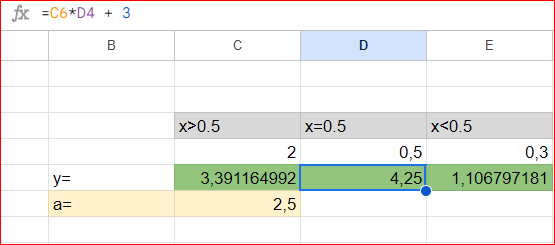
Рисунок 5 - Результат вычисления программы при X > 0.5

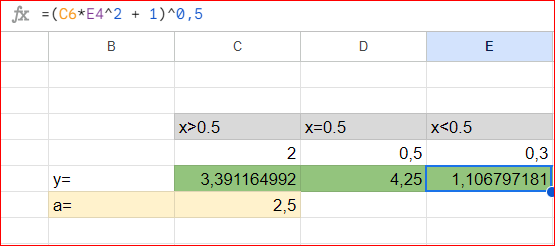
Рисунок 6 - Результат вычисления программы при X = 0.5

Рисунок 7 - Результат вычисления программы при X < 0.5

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

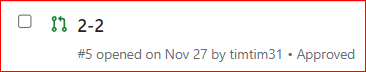


Рисунок 8 – Approve задачи 2-2