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 ПРАКТИЧЕСКОЙ РАБОТЕ №2

По дисциплине «Введение в языки программирования»

Выполнил: ст. гр. ТКИ – 111

Панаргин В.М.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

Оглавление

[**Задание 2–1** 3](#_Toc91434256)

[**Условие задания** 3](#_Toc91434257)

[**Блок-схема алгоритма** 4](#_Toc91434258)

[**Программа на языке С++** 5](#_Toc91434259)

[**Решение тестовых примеров на С++** 8](#_Toc91434260)

[**Задание 2–2** 9](#_Toc91434261)

[**Условие задания** 9](#_Toc91434262)

[**Блок-схема алгоритма** 10](#_Toc91434263)

[**Программа на языке С++** 11](#_Toc91434264)

[**Решение тестовых примеров на С++** 13](#_Toc91434265)

[**Решение тестовых примеров в Excel** 14](#_Toc91434266)

[**Задание 2–3** 15](#_Toc91434267)

[**Условие задания** 15](#_Toc91434268)

[**Блок-схема алгоритма** 16](#_Toc91434269)

[**Программа на языке С++** 17](#_Toc91434270)

[**Решение тестовых примеров на С++** 20](#_Toc91434271)

[**Решение тестовых примеров в Калькуляторе** 21](#_Toc91434272)

# **Задание 2–1**

## **Условие задания**

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**

|  |  |
| --- | --- |
| Номер варианта | Задача |
| 12 | Вывести числа a,b,c на экран в порядке:   * возрастания * убывания |

## **Блок-схема алгоритма**

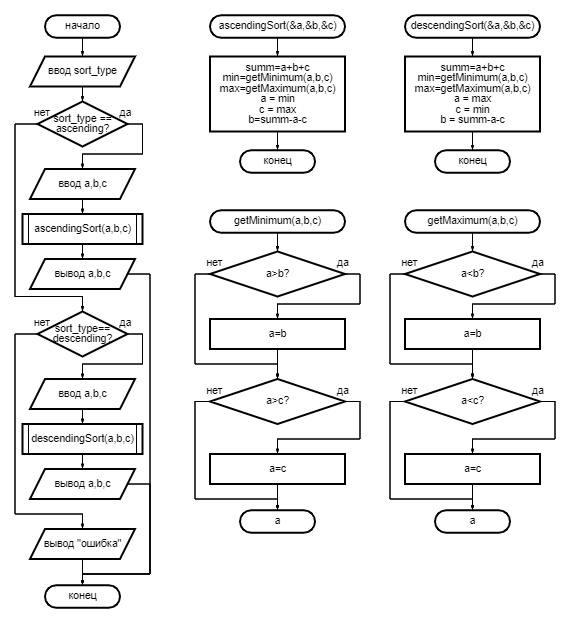


Рисунок 1 - Блок-схема 2-1

## **Программа на языке С++**

#include <iostream>

#include <string>

using namespace std;

/\*\*

\* \brief Перечислимый тип видов сортировок.

\*/

enum class SortType

{

/\*\*

\* \brief сортировка по возрастанию.

\*/

ascending,

/\*\*

\* \brief сортировка по убыванию.

\*/

descending

};

/\*\*

\* \brief Функция сортировки по возрастанию трех чисел.

\* \param a первое число.

\* \param b второе число.

\* \param c третье число.

\* \return 0 в случае успеха.

\*/

int ascendingSort(double &a, double &b, double &c);

/\*\*

\* \brief Функция сортировки по убыванию трех чисел.

\* \param a первое число.

\* \param b второе число.

\* \param c третье число.

\* \return 0 в случае успеха.

\*/

int descendingSort(double &a, double &b, double &c);

/\*\*

\* \brief Функция нахождения минимального из трех чисел.

\* \param a первое число.

\* \param b второе число.

\* \param c третье число.

\* \return минимальное из трех чисел.

\*/

double getMinimum(double a, double b, double c);

/\*\*

\* \brief Функция нахождения максимального из трех чисел.

\* \param a первое число.

\* \param b второе число.

\* \param c третье число.

\* \return максимальное из трех чисел.

\*/

double getMaximum(double a, double b, double c);

/\*\*

\* \brief Точка вход в программу

\* \return Код ошибки, если 0 - успешное выполнение

\*/

int main()

{

setlocale(LC\_ALL, "Russian");

cout << "Выберите как сортировать числа: " << static\_cast<int>(SortType::ascending) << " - по возрастанию, " << static\_cast<int>(SortType::descending) << " - по убыванию ";

int sort\_type;

cin >> sort\_type;

switch (sort\_type)

{

case static\_cast<int>(SortType::ascending):

{

cout << "Введите первое число: ";

double a;

cin >> a;

cout << "Введите второе число: ";

double b;

cin >> b;

cout << "Введите третье число: ";

double c;

cin >> c;

ascendingSort(a, b, c);

cout << a << " " << b << " " << c << endl;

break;

}

case static\_cast<int>(SortType::descending):

{

cout << "Введите первое число: ";

double a;

cin >> a;

cout << "Введите второе число: ";

double b;

cin >> b;

cout << "Введите третье число: ";

double c;

cin >> c;

descendingSort(a, b, c);

cout << a << " " << b << " " << c << endl;

break;

}

default:

cout << "Ошибка!";

}

return 0;

}

int ascendingSort(double &a, double &b, double &c)

{

const double summ = a + b + c;

const double min = getMinimum(a, b, c);

const double max = getMaximum(a, b, c);

a = min;

c = max;

b = summ - a - c;

return 0;

}

int descendingSort(double &a, double &b, double &c)

{

const double summ = a + b + c;

const double min = getMinimum(a, b, c);

const double max = getMaximum(a, b, c);

c = min;

a = max;

b = summ - a - c;

return 0;

}

double getMinimum(double a, double b, double c)

{

if (a > b) a = b;

if (a > c) a = c;

return a;

}

double getMaximum(double a, double b, double c)

{

if (a < b) a = b;

if (a < c) a = c;

return a;

}if (a > b) a = b;

if (a > c) a = c;

return a;

}

double getMaximum(double a, double b, double c)

{

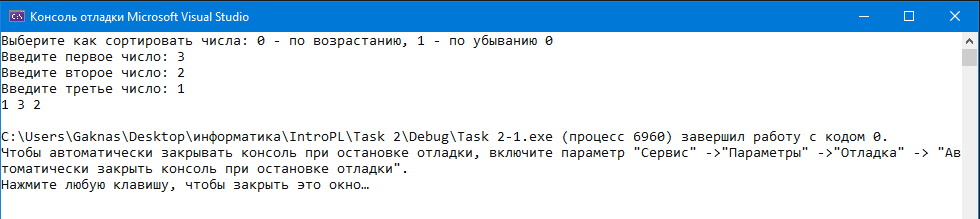
if (a < b) a = b;

if (a < c) a = c;

return a;

}}

## **Решение тестовых примеров на С++**



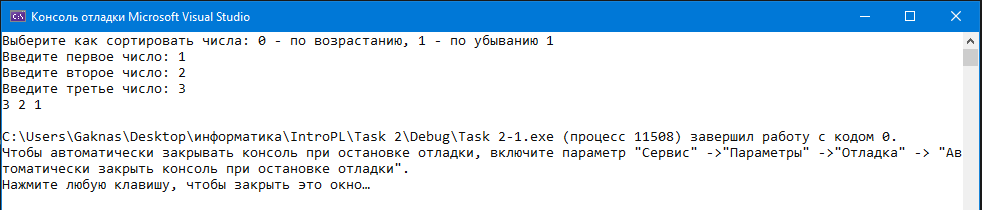


Рисунок 2 - Решение тестового примера для задания 2-1 на С++

# **Задание 2–2**

## **Условие задания**

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 2**

|  |  |  |
| --- | --- | --- |
| Вариант | Функция | Константы |
| 12 |  | a = 2.5 |

## **Блок-схема алгоритма**

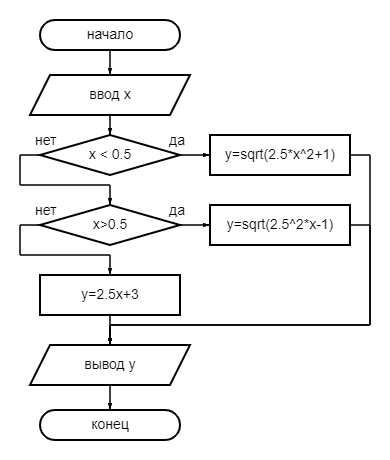


Рисунок 3 - Блок-схема 2-2

## **Программа на языке С++**

#include <cmath>

#include <iostream>

const double A = 2.5;

/\*\*

\brief Функция рассчитывает значение в точке

/param value значение в точке

/return Значение функции

\*/

double getFunctionLessTreshold(const double value);

/\*\*

\brief Функция рассчитывает значение в точке

/param value значение в точке

/return Значение функции

\*/

double getFunctionEqualTreshold(const double value);

/\*\*

\brief Функция рассчитывает значение в точке

/param value значение в точке

/return Значение функции

\*/

double getFunctionAboveTreshold(const double value);

/\*\*

\brief Точка входа в программу

/return 0 в случае успеха

\*/

int main()

{

setlocale(LC\_ALL, "Russian");

const double THRESHOLD = 0.5;

std::cout << "Введите значение x = ";

double x;

std::cin >> x;

double y;

if (x < THRESHOLD)

{

y = getFunctionLessTreshold(x);

}

else if (x > THRESHOLD)

{

y = getFunctionAboveTreshold(x);

}

else

{

y = getFunctionEqualTreshold(x);

}

std::cout << "y = " << y;

return 0;

}

double getFunctionLessTreshold(const double value)

{

return sqrt(A \* pow(value, 2) + 1);

}

double getFunctionEqualTreshold(const double value)

{

return A \* value + 3;

}

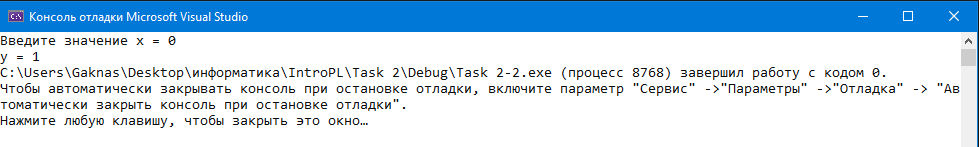
double getFunctionAboveTreshold(const double value)

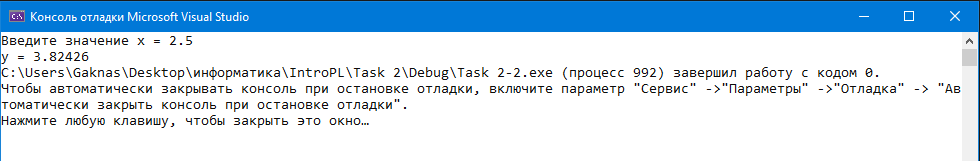
{

return sqrt(pow(A, 2) \* value - 1);

}

## **Решение тестовых примеров на С++**





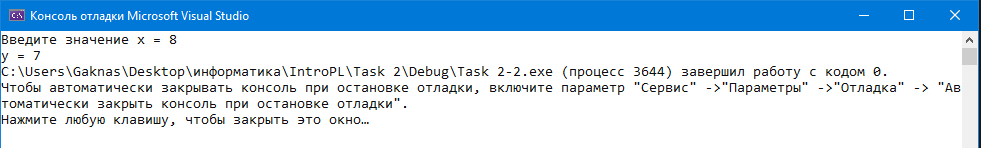


Рисунок 4 - Решение тестовых примеров для задания 2-2 на С++

## **Решение тестовых примеров в Excel**

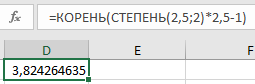


Рисунок 5 - решение тестового примера для задания 2-2 в Excel

# **Задание 2–3**

## **Условие задания**

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.

|  |  |
| --- | --- |
| Номер варианта | Задача |
| 12 | Дано число x. Вывести в порядке возрастания числа: sin x, cos x, ln x. Если при каком либо x некоторые из выражений не имеют смысла, вывести сообщение об этом и сравнивать значения только тех, которые имеют смысл. |

## **Блок-схема алгоритма**

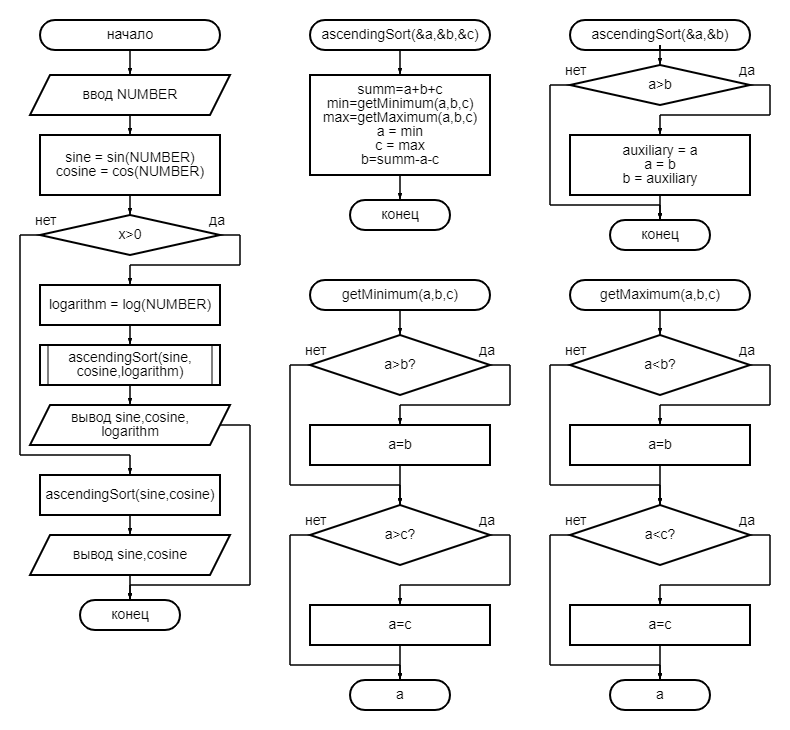


Рисунок 6 - Блок-схема 2-3

## **Программа на языке С++**

#include <iostream>

/\*\*

\* \brief Считывает натуральное число из консоли.

\* \param message Мотивационное сообщение для пользователя.

\* \return Натуральное число.

\*/

int read\_number(const std::string& message);

/\*\*

\* \brief Функция сортировки по возрастанию трех чисел.

\* \param a первое число.

\* \param b второе число.

\* \param c третье число.

\* \return 0 в случае успеха.

\*/

int ascendingSort(double &a, double &b, double &c);

/\*\*

\* \brief Функция сортировки по возрастанию двух чисел.

\* \param a первое число.

\* \param b второе число.

\* \return 0 в случае успеха.

\*/

int ascendingSort(double &a, double &b);

/\*\*

\* \brief Функция нахождения минимального из трех чисел.

\* \param a первое число.

\* \param b второе число.

\* \param c третье число.

\* \return минимальное из трех чисел.

\*/

double getMinimum(double a, double b, double c);

/\*\*

\* \brief Функция нахождения максимального из трех чисел.

\* \param a первое число.

\* \param b второе число.

\* \param c третье число.

\* \return максимальное из трех чисел.

\*/

double getMaximum(double a, double b, double c);

/\*\*

\* \brief Узнает, существует ли логарифм в заданной точке \a x.

\* \param x Заданная точка.

\* \return true, если значение логарифма в заданной точке \a x существует.

\*/

bool isCalculated(double x);

/\*\*

\* \brief Точка входа в программу.

\* \return 0 в случае успеха.

\*/

int main()

{

setlocale(LC\_ALL, "Russian");

try

{

const auto NUMBER = read\_number("Введите число ");

double sine = sin(NUMBER);

std::cout << "Синус числа равен " << sine << "\n";

double cosine = cos(NUMBER);

std::cout << "Косинус числа равен " << cosine << "\n";

if (isCalculated(NUMBER)) {

const double logarithm = log(NUMBER);

std::cout << "Логарифм числа равен " << logarithm << "\n";

ascendingSort(sine, cosine, logarithm);

std::cout << sine << " " << cosine << " " << logarithm << std::endl;

}

else {

std::cout << "Логарифм не вычисляется" << "\n";

ascendingSort(sine, cosine);

std::cout << sine << " " << cosine << std::endl;

}

return 0;

}

catch (...)

{

return 1;

}

}

double getMinimum(double a, double b, double c)

{

if (a > b) a = b;

if (a > c) a = c;

return a;

}

double getMaximum(double a, double b, double c)

{

if (a < b) a = b;

if (a < c) a = c;

return a;

}

int ascendingSort(double &a, double &b, double &c)

{

const double summ = a + b + c;

const double min = getMinimum(a, b, c);

const double max = getMaximum(a, b, c);

a = min;

c = max;

b = summ - a - c;

return 0;

}

int ascendingSort(double &a, double &b)

{

if (a > b) {

double auxiliary = a;

a = b;

b = auxiliary;

}

return 0;

}

int read\_number(const std::string& message)

{

std::cout << message;

int number = 0;

std::cin >> number;

return number;

}

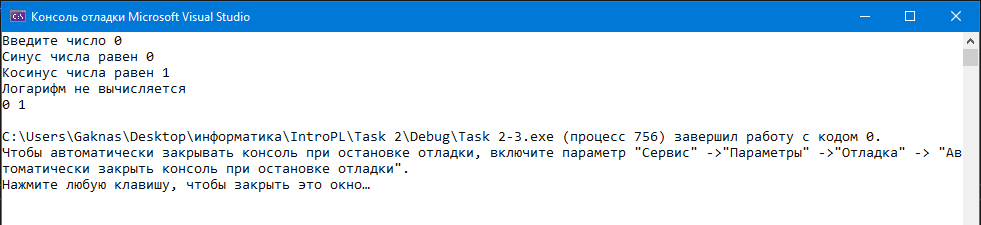
bool isCalculated(const double x)

{

return x > 0;

}

## **Решение тестовых примеров на С++**



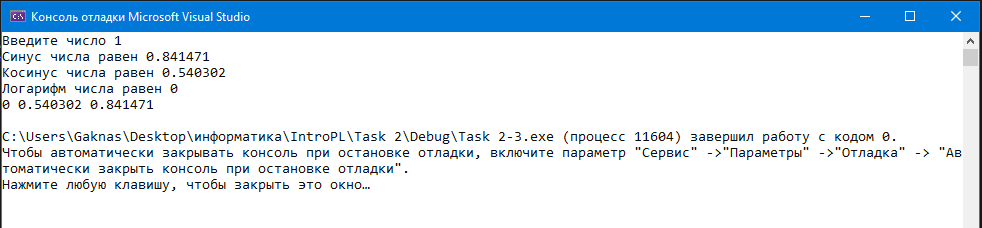


Рисунок 7 - Решение тестовых примеров для задания 2-3 на С++

## **Решение тестовых примеров в Калькуляторе**

