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 «Процедурное программирование»

Выполнил: ст. гр. ТКИ – 112

Зинченко Б.А.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

**Оглавление**

[ЗАДАНИЕ 2.1 3](#_Toc72361031)

[ЗАДАНИЕ 2.2 6](#_Toc72361033)

[ЗАДАНИЕ 2.3 9](#_Toc72361035)

ЗАДАНИЕ 2.1

1. Условие 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.

Поменяйте между собой значения переменных a и b,

1. Воспользовавшись третьей переменной с.
2. Без использования переменной.
3. Блок-схема алгоритма

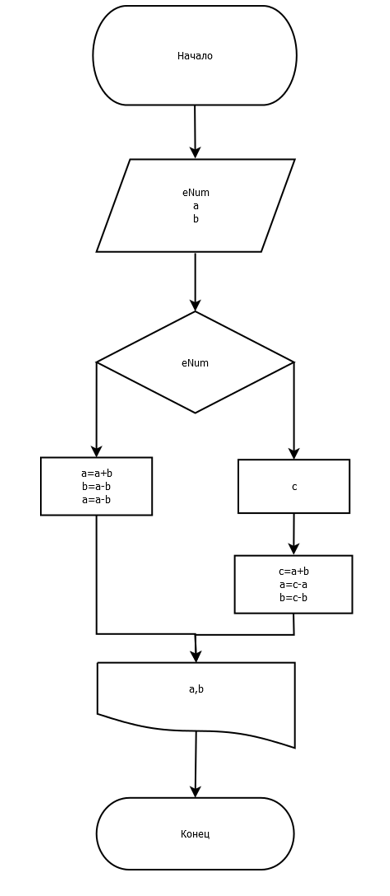


Рисунок 1 – Блок-схема 2.1

1. Текст программы на языке C++

#define \_USE\_MATH\_DEFINES // for C++

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

/\*\*

\* \brief Выбор замены переменных

\*\*\*/

enum class ActionChoice

{

two,

three

};

int main() {

setlocale(LC\_ALL, "Russian");

double a, b, c;

std::cout << "Введите первую переменную a и вторую переменную b ";

std::cin >> a >> b;

std::cout << "Выберите способ перемены значений (без новой переменной=0 , с использованием новой переменной=1" << endl;

int input;

std::cin >> input;

const auto choice = static\_cast<ActionChoice>(input);

switch (choice)

{

case ActionChoice::two:

{

a = a + b;

b = a - b;

a = a - b;

std::cout << "Переменные были заменены друга на друга, теперь a= " << a << ", а b= " << b;

break;

}

case ActionChoice::three:

{

c = a + b;

a = c - a;

b = c - b;

std::cout << "Переменные были заменены друга на друга, теперь a= " << a << ", а b= " << b;

break;

}

}

return 0;

}Результаты работы программы

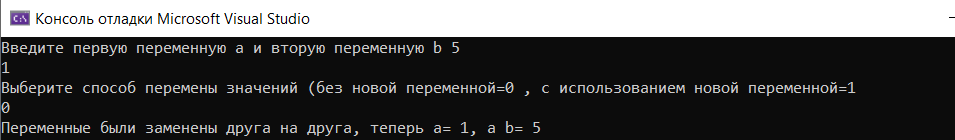


Рисунок 2 ˗ Вывод программы 2.1

1. Проверка расчётов в Excel

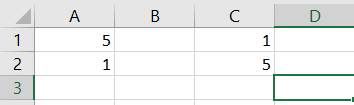
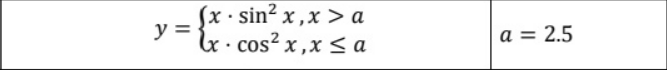


Рисунок 3 ˗ Проверка Расчёта программы 2.1

ЗАДАНИЕ 2.2

1. Условие 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.



1. Блок-схема алгоритма

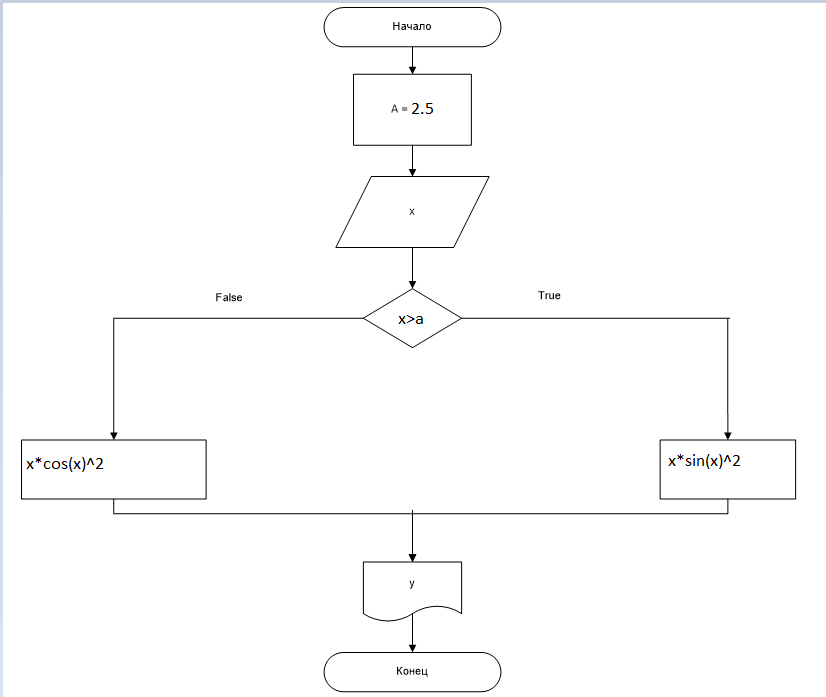


Рисунок 4 ˗ Блок-схема 2.2

1. Текст программы на языке C++

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

/\*\*

\* \brief Функциия для расчета y1.

\* \param x Параметр x.

\* \return Возвращает значение y1.

\*/

double GetFirst(const double x);

/\*\*

\* \brief Функциия для расчета y2.

\* \param x Параметр x.

\* \return Возвращает значение y2.

\*/

double GetSecond(const double x);

/\*\*

\* \brief Точка входа в программу

\* \return Возвращает нуль, в случае успеха

\*/

int main()

{

setlocale(LC\_ALL, "Russian");

const double a = 2.5;

double x;

cout << "Введите число x \n";

cin >> x;

if (x > a)

{

auto first = GetFirst(x);

cout << first;

}

else

{

auto second = GetSecond(x);

cout << second;

}

}

double GetFirst(double x) {

return x \* pow(sin(x), 2);

}

double GetSecond(double x) {

return x \* pow(cos(x), 2);

} Результаты работы программы

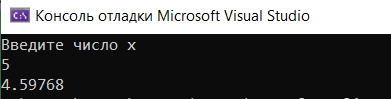


Рисунок 5 ˗ Вывод программы 2.2

1. Проверка расчётов в Excel

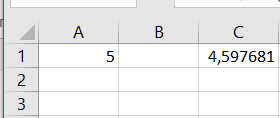


Рисунок 6 ˗ Проверка Расчёта программы 2.2

ЗАДАНИЕ 2.3

1. Условие 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.

Даны три точки A(x1, y1), B(x2, y2) и C(x3, y3). Определить, будут ли  
они расположены на одной прямой. Если нет, то вычислить угол B.

Блок-схема алгоритма

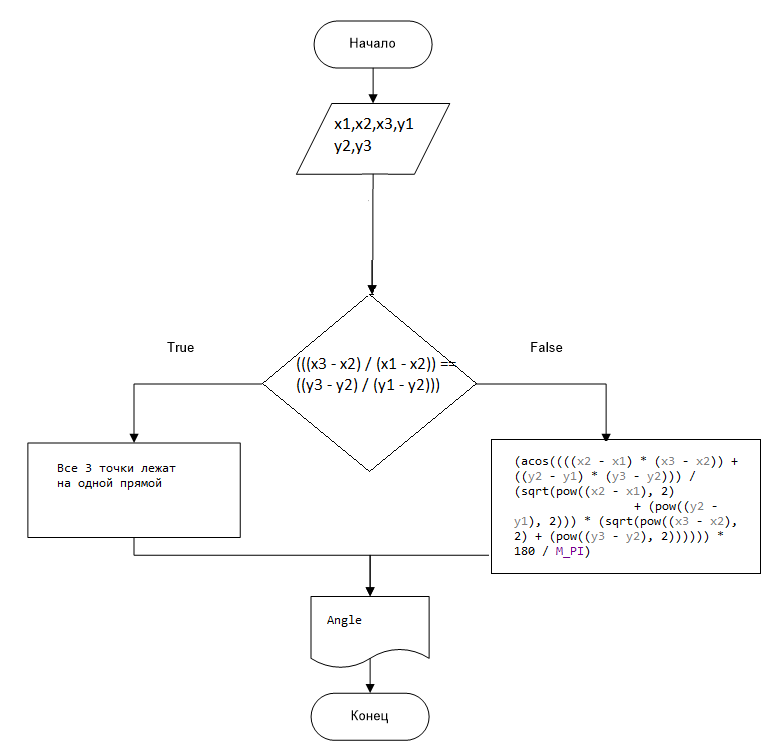


Рисунок 7 ˗ Блок-схема 2.3

1. Текст программы на языке C++

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

/\*\*

\* \brief Функциия для расчета угла.

\* \param x1,x2,x3,y1,y2,y3 Параметр x1,x2,x3,y1,y2,y3.

\* \return Возвращает значение angle.

\*/

double GetAngle(const double x1, const double x2, const double x3, const double y1, const double y2, const double y3);

/\*\*

\* \brief Точка входа в программу

\* \return Возвращает нуль, в случае успеха

\*/

int main()

{

setlocale(LC\_ALL, "Russian");

double x1, y1;

cout << "Введите координаты точки A (Xa,Ya) \n";

cin >> x1 >> y1;

double x2, y2;

cout << "Введите координаты точки B (Xb,Yb) \n";

cin >> x2 >> y2;

double x3, y3;

cout << "Введите координаты точки C (Xc,Yc) \n";

cin >> x3 >> y3;

if (((x3 - x2) / (x1 - x2)) == ((y3 - y2) / (y1 - y2)))

{

cout << "Все 3 точки лежат на одной прямой";

}

else

{

cout << "Точки не лежат на одной прямой" << endl;

auto angle = GetAngle(x1, x2, x3, y1, y2, y3);

cout << "Угол ABC равен " << angle << " градусов";

}

}

double GetAngle(double x1, double x2, double x3,

double y1, double y2, double y3) {

return (acos((((x2 - x1) \* (x3 - x2)) + ((y2 - y1) \* (y3 - y2))) / (sqrt(pow((x2 - x1), 2)

+ (pow((y2 - y1), 2))) \* (sqrt(pow((x3 - x2), 2) + (pow((y3 - y2), 2)))))) \* 180 / M\_PI);

}Результаты работы программы

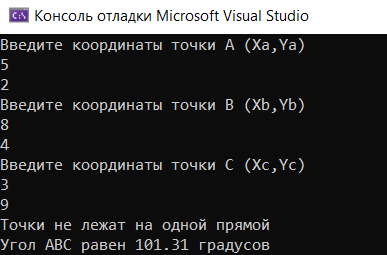


Рисунок 8 ˗ Вывод программы 2.3

1. Проверка расчётов в Excel

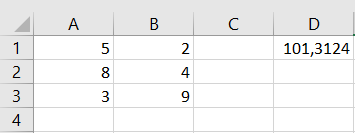


Рисунок 9 ˗ Проверка Расчёта программы 2.3