Практическая работа №7

Тема: «Функции»

**Задание 7-12**

Написать функцию, которая вычисляет сопротивление цепи, состоящей из трех проводников. Параметрами ее являются значения сопротивлений, а также тип соединения – последовательное или параллельное (цифрами 1 или 2). Проверить ее в работе, написав программу с ее использованием.

#include <iostream>

// Функция для вычисления сопротивления в зависимости от типа соединения

double calculateResistance(double R1, double R2, double R3, int connectionType) {

if (connectionType == 1) {

// Последовательное соединение

return R1 + R2 + R3;

} else if (connectionType == 2) {

// Параллельное соединение

return 1 / (1 / R1 + 1 / R2 + 1 / R3);

} else {

// Неподдерживаемый тип соединения

std::cerr << "Ошибка: Неподдерживаемый тип соединения." << std::endl;

return -1; // Возвращаем значение -1 в случае ошибки

}

}

int main() {

setlocale(LC\_ALL, "RU");

double resistance1, resistance2, resistance3;

int connectionType;

// Ввод значений сопротивлений и типа соединения

std::cout << "Введите значения сопротивлений R1, R2, R3: ";

std::cin >> resistance1 >> resistance2 >> resistance3;

std::cout << "Введите тип соединения (1 - последовательное, 2 - параллельное): ";

std::cin >> connectionType;

// Вычисление сопротивления и вывод результата

double totalResistance = calculateResistance(resistance1, resistance2, resistance3, connectionType);

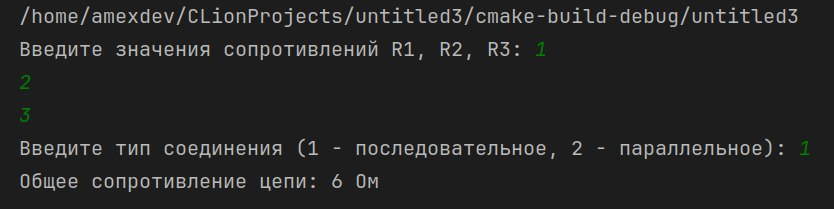
if (totalResistance != -1) {

std::cout << "Общее сопротивление цепи: " << totalResistance << " Ом" << std::endl;

}

return 0;

}



**Задание 7-17**

Даны четыре натуральных числа. Найти наибольший общий делитель (НОД) для этих четырех чисел. Поиск НОД двух чисел оформить функцией.

#include <iostream>

// Функция для нахождения НОД двух чисел

int findGCD(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

// Функция для нахождения НОД четырех чисел

int findGCDFourNumbers(int a, int b, int c, int d) {

// Находим НОД для первых двух чисел

int gcdAB = findGCD(a, b);

// Нахождение НОД для третьего числа и результата предыдущего шага

int gcdABC = findGCD(gcdAB, c);

// Нахождение НОД для четвертого числа и результата предыдущего шага

int gcdABCD = findGCD(gcdABC, d);

return gcdABCD;

}

int main() {

setlocale(LC\_ALL, "RU");

// Ввод четырех натуральных чисел

int num1, num2, num3, num4;

std::cout << "Введите четыре натуральных числа: ";

std::cin >> num1 >> num2 >> num3 >> num4;

// Нахождение НОД четырех чисел

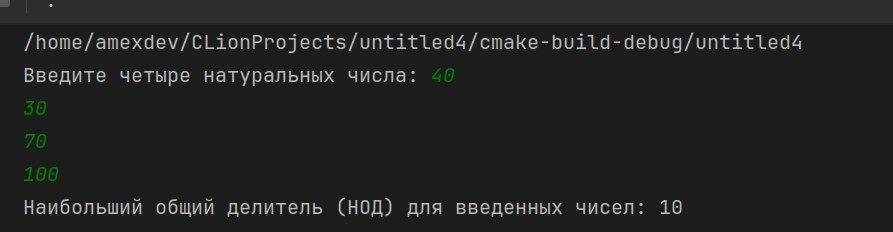
int result = findGCDFourNumbers(num1, num2, num3, num4);

// Вывод результата

std::cout << "Наибольший общий делитель (НОД) для введенных чисел: " << result << std::endl;

return 0;

}



**Задание 7-23**

Составить программу для умножения двух обыкновенных дробей *a/b\*c/d*.

#include <iostream>

int main() {

setlocale(LC\_ALL, "RU");

// Ввод дробей a/b и c/d

int a, b, c, d;

std::cout << "Введите числитель и знаменатель первой дроби (a/b): ";

if (!(std::cin >> a) || !(std::cin.ignore() >> b) || b == 0) {

std::cerr << "Ошибка ввода первой дроби." << std::endl;

return 1;

}

std::cout << "Введите числитель и знаменатель второй дроби (c/d): ";

if (!(std::cin >> c) || !(std::cin.ignore() >> d) || d == 0) {

std::cerr << "Ошибка ввода второй дроби." << std::endl;

return 1;

}

// Умножение дробей

int numerator = a \* c;

int denominator = b \* d;

// Вывод результата

std::cout << "Результат умножения: " << numerator << "/" << denominator << std::endl;

return 0;

}

