Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Специальность «Программная инженерия»

Кафедра инженерной психологии и эргономики

Учебная дисциплина «Основы алгоритмизации и программирования»

ОТЧЕТ

по лабораторной работе No8

«Функции»

Вариант 7

Подготовил: Гавлев Н.В.

Проверил: Усенко Ф.В.

Минск 2024

**Цель работы:** сформировать навыки и умения обработки структурированных типов данных, организованных в виде функций.

Задание 1. Напишите программу нахождения, следующего за данным совершенного числа. Совершенным называется число, сумма делителей которого, не считая самого числа, равна этому числу. Первое совершенное число 6 (6 = 1 + 2 + 3).

Задание 2. Используя перегрузку методов, создайте программу согласно варианту

а) для деления целых чисел; б) для деления комплексных чисел.

1 Задание:

#include <iostream>

#include <cmath>

using namespace std;

bool check\_prime\_number(unsigned long long number\_for\_check)

{

for (long long i = 2; i <= (number\_for\_check / 2); i++)

{

if (number\_for\_check % i == 0)

{

return false;

}

}

return true;

}

unsigned long long check\_perfect\_number(int base\_number)

{

bool result\_of\_prime = check\_prime\_number(pow(2, base\_number) - 1);

if (result\_of\_prime == true)

return(pow(2, base\_number - 1) \* (pow(2, base\_number) - 1));

else

return 0;

}

int main()

{

setlocale(LC\_ALL, "RU");

int user\_base\_perfect\_number;

cout << "Введите совершенное число начиная с которого вы хотите начать нахождение новых совершенных чисел ";

cin >> user\_base\_perfect\_number;

int max\_base\_number = { 1000 };

setlocale(LC\_ALL, "RU");

for (int i = 2; i < max\_base\_number; i++)

{

if (check\_perfect\_number(i) != 0 and check\_perfect\_number(i) > user\_base\_perfect\_number)

cout << "Число " << pow(2, i - 1) \* (pow(2, i) - 1) << " совершенное" << endl;

}

}

2 Задание:

#include <iostream>

#include <cmath>

#include <complex>

using namespace std;

double division(int x1, int x2)

{

return static\_cast<double> (x1) / x2;

}

complex<double> division(complex<double> z1, complex<double> z2)

{

return (z1 / z2);

}

void printcomplex(complex<double> z)

{

double real = z.real();

double imag = z.imag();

if (imag >= 0)

cout << real << " + " << imag << "i";

else

cout << real << " - " << abs(imag) << "i";

}

int main()

{

string answer;

setlocale(LC\_ALL, "RU");

while (true)

{

cout << "Если вы хотите делить целые числа введите 0, если комплексные 1: ";

cin >> answer;

if (answer == "0")

{

int x1, x2;

cout << "Введите 1 и 2 число через пробел ";

cin >> x1 >> x2;

cout << division(x1, x2) << endl;

}

else if (answer == "1")

{

double real1, imag1, real2, imag2;

cout << "Введите 1 комплексное число в виде x + iy, где x - ваш первый ввод, а y - второй (разделяются пробелом) ";

cin >> real1 >> imag1;

cout << "Введите 2 комплексное число в виде x + iy, где x - ваш первый ввод, а y - второй (разделяются пробелом) ";

cin >> real2 >> imag2;

complex<double> result = division(complex<double>(real1, imag1), complex<double>(real2, imag2));

cout << "Результат деления комплексных чисел: ";

printcomplex(result);

cout << endl;

}

else

{

cout << "Введено неккоректное значение" << endl;

}

}

}

На рисунках 1-3 показаны скриншоты работающей программы.

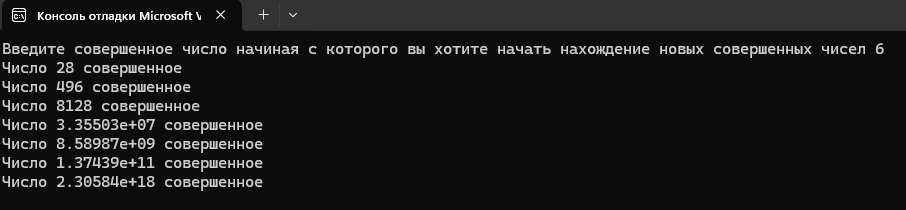


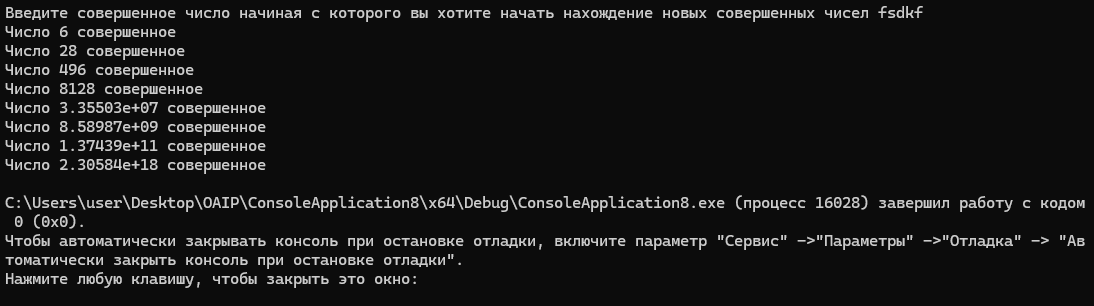
Рисунок 1 – Поиск совершенных чисел начиная с 6

Рисунок 2 – Поиск совершенных чисел если ввести некорректное значение

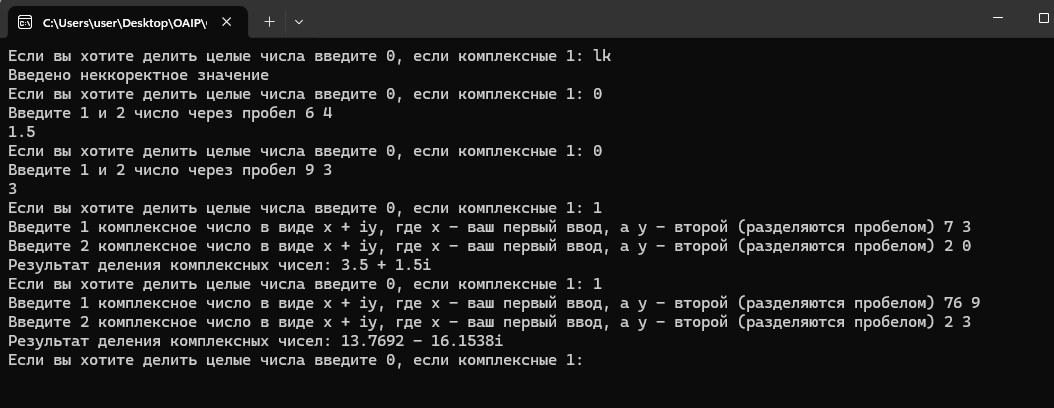


Рисунок 3 – Работа программы с делением целых и комплексных чисел

Далее приведена блок схемы функций составляющих программу:

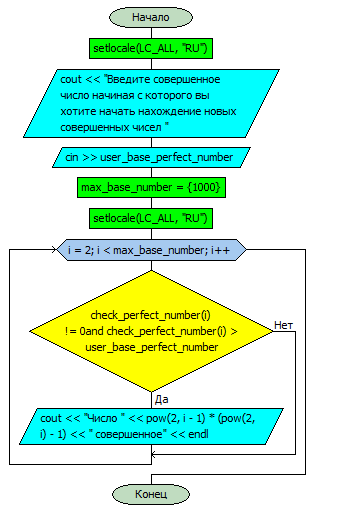


Рисунок 4 – Функция int main() в 1 программе

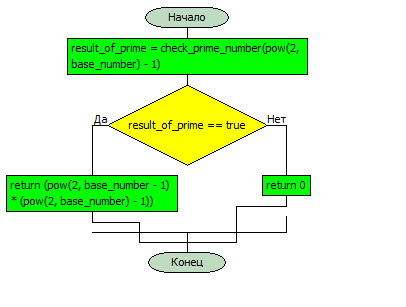


Рисунок 5 – Функция unsigned long long check\_perfect\_number(int base\_number)

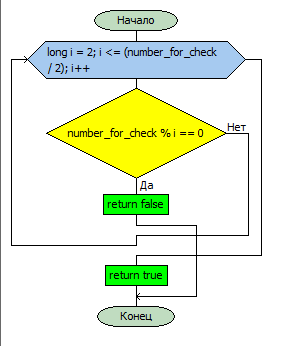


Рисунок 6 – Функция bool check\_prime\_number(unsigned long long number\_for\_check)

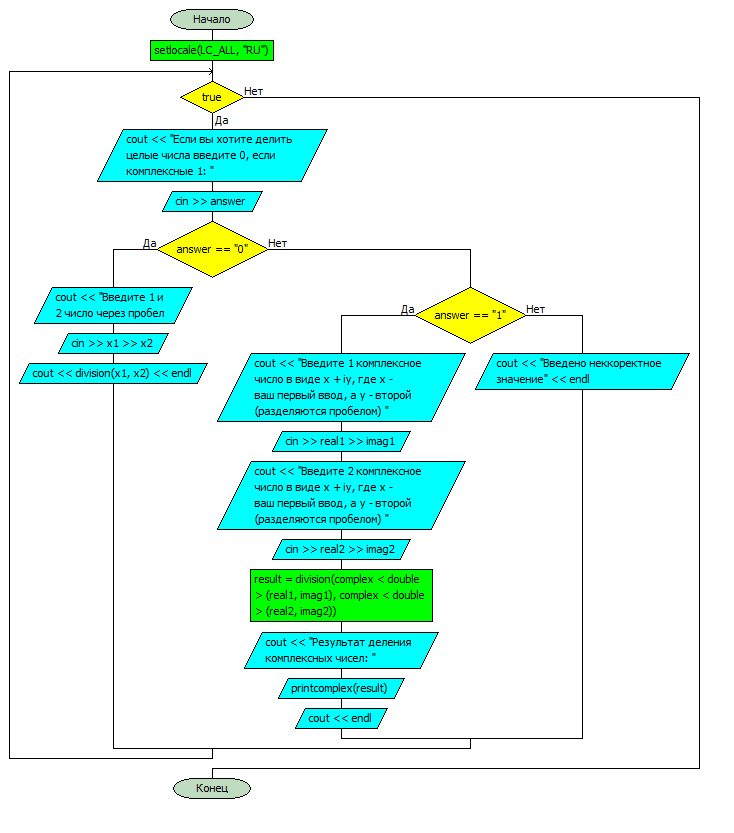


Рисунок 7 – Функция int main() во 2 программе

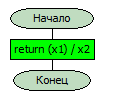


Рисунок 8 – Функция double division(int x1, int x2)

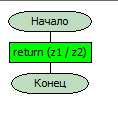


Рисунок 9 ‒ complex<double> division(complex<double> z1, complex<double> z2)

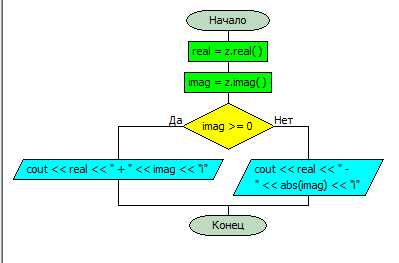


Рисунок 10 - void printcomplex(complex<double> z)

**Вывод:** В ходе выполнения лабораторной работы мы научились создавать функции и работать с ними. В программе по нахождению совершенных чисел использован более эффективный способ нахождения их, чем простым перебором. Мы поняли, что функции можно влаживать друг в друга и тем самым делать код более читабельным. Также мы изучили перегрузку функций и научились работать с комплексными числами в рамках библиотеки <complex> в С++.