**Отчет для лабораторных работ по предмету математическое программирование**

Лабораторная работа №1

**Задание 1** Разработайте три функции (start, dget и iget), используя следующие спецификации:

#include "pch.h"

#include "Auxil.h"

#include <ctime>

namespace auxil

{

void start() // старт генератора сл. чисел

{

srand((unsigned)time(NULL));

};

double dget(double rmin, double rmax) // получить случайное число

{

return ((double)rand() / (double)RAND\_MAX) \* (rmax - rmin) + rmin;

};

int iget(int rmin, int rmax) // получить случайное число

{

return (int)dget((double)rmin, (double)rmax);

};

}

Листинг файла Auxil.cpp

#pragma once

#include <cstdlib>

namespace auxil

{

void start(); // старт генератора сл. чисел

double dget(double rmin, double rmax); // получить случайное число

int iget(int rmin, int rmax); // получить случайное число

};

Листинг файла Auxil.h

Разработаны три функции (start, dget и iget).

**Задание 2** Для проверки работоспособности разработанных функций и приобретения навыков замера продолжительности процесса вычисления реализуйте программу, приведенную в примере 2.

int \_tmain(int argc, \_TCHAR\* argv[])

{

double av1 = 0, av2 = 0;

clock\_t t1 = 0, t2 = 0;

setlocale(LC\_ALL, "rus");

auxil::start(); // старт генерации

t1 = clock(); // фиксация времени

for (int i = 0; i < CYCLE; i++)

{

av1 += (double)auxil::iget(-100, 100); // сумма случайных чисел

av2 += auxil::dget(-100, 100); // сумма случайных чисел

}

t2 = clock(); // фиксация времени

cout << endl << "количество циклов: " << CYCLE;

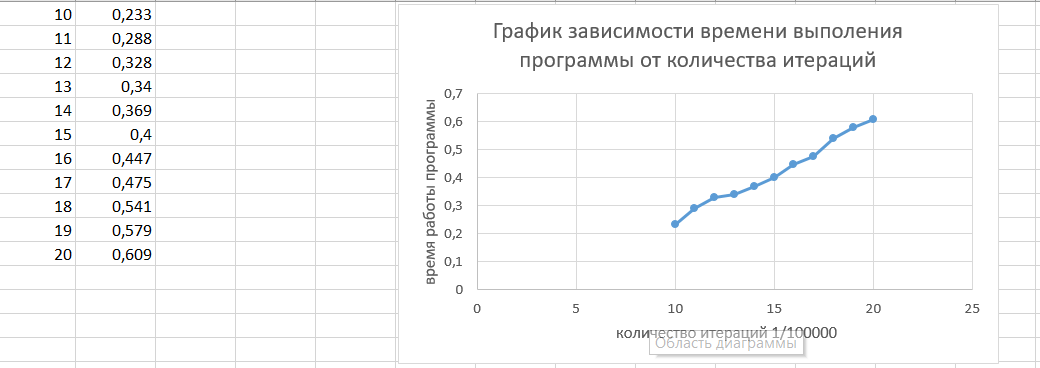
cout << endl << "среднее значение (int): " << av1 / CYCLE;

cout << endl << "среднее значение (double): " << av2 / CYCLE;

cout << endl << "продолжительность (у.е): " << (t2 - t1);

cout << endl << " (сек): " << ((double)(t2 - t1)) / ((double)CLOCKS\_PER\_SEC);

cout << endl;

****

**Вывод:** График зависимости времени выполнения от кол-ва итераций показывает, что время выполнения алгоритма растет линейно при увеличения итераций.

**Задание 3** Проведите необходимые эксперименты и постройте график зависимости (Excel) продолжительности процесса вычисления от количества циклов в примере 2. Проанализируйте характер зависимости. Проведите исследование любого другого рекурсивного алгоритма, например, вычисления факториала или генератора чисел Фибоначчи (прим. – например вычислите каким будет 100-е, 200-е, 300-е и т.д число), и включите в отчет график.

int f(int n)

{

if (n == 1 || n == 2) return (n - 1);

return f(n - 1) + f(n - 2);

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int n;

cout << "n="; cin >> n;

t3 = clock();

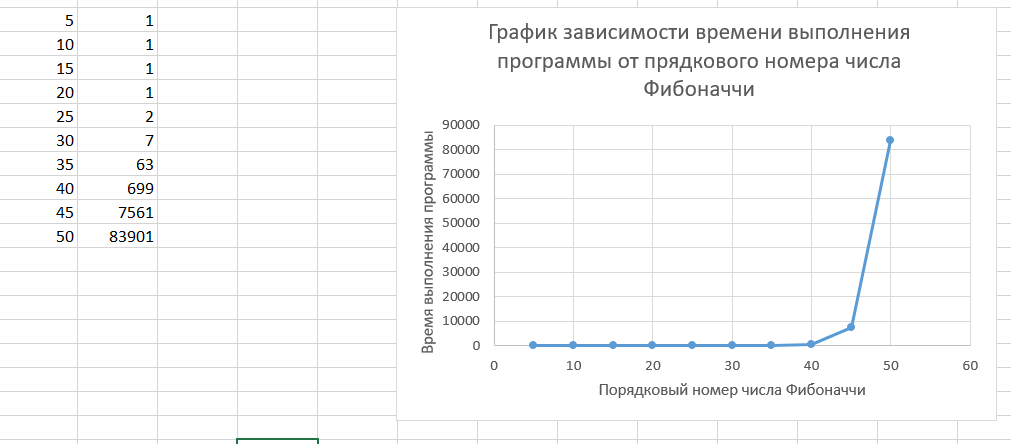
cout << f(n) << "\n";

t4 = clock();

cout << endl << "продолжительность (у.е) " << (t4 - t3);

cout << endl << " (сек): " << ((double)(t4 - t3)) / ((double)CLOCKS\_PER\_SEC);

cout << endl;

}

**Вывод:** График зависимости времени выполнения от порядкового номера числа Фибоначчи показывает, что время выполнения алгоритма растет экспоненциально при увеличения его порядкового номера.