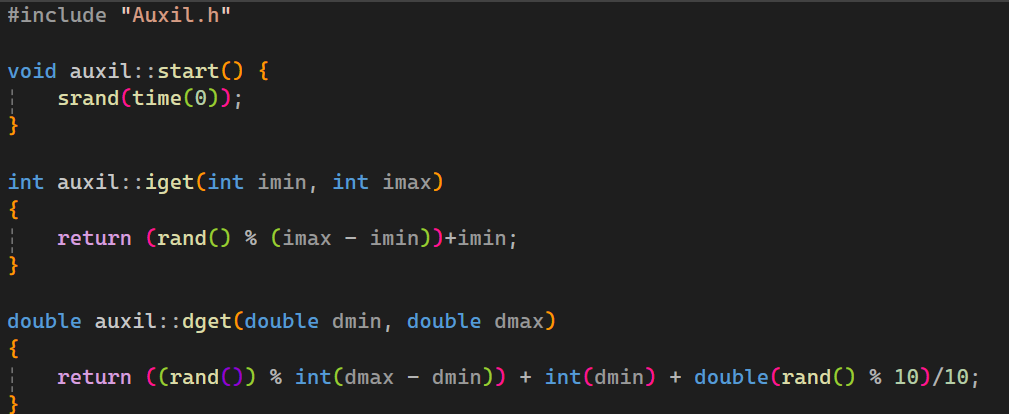
**Лабораторная работа 1. Вспомогательные функции**

**ЦЕЛЬ РАБОТЫ:** приобретение навыков составления и отладки программ с использованием пользовательских функций для замера продолжительности процесса вычисления.

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



//-- установка начального числа для генератора псевдослучайных // чисел

// # include "Auxil.h"

// namespace auxil

**void start();**

// функция устанавливает в качестве начального числа для // генератора псевдослучайных чисел текущее значение // системного времени в фомате функции time()

***Задание 2***

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

**Пример 2.**

#include "stdafx.h"

#include "Auxil.h" // вспомогательные функции

#include <iostream>

#include <ctime>

#include <locale>

#define CYCLE 1000000 // количество циклов

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

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

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

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

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

std::cout<<std::endl<< " (сек): "

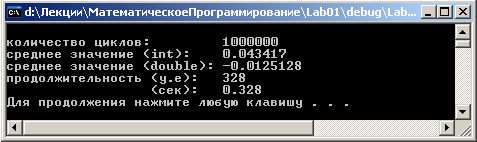
<<((double)(t2-t1))/((double)CLOCKS\_PER\_SEC);

std::cout<<std::endl;

system("pause");

return 0;

}



**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.

**Примечание**: продолжительность вычисления измерять в условных единицах процессорного времени (функция **clock**).

**Пример применения**

#include "stdafx.h"

#include <iostream>

#include <ctime>

#include "Knapsack.h"

#include "Auxil.h"

#define N 100

clock\_t SS[N];

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

{

setlocale(LC\_ALL, "rus");

int V = 1000, // вместимость рюкзака

v[N], // размер предмета каждого типа

c[N]; // стоимость предмета каждого типа

short m[N]; // количество предметов каждого типа {0,1}

int maxcc = 0;

int n = 0;

auxil::start();

for(int i = 0; i < N; i++) v[i] = auxil::iget(10, 100);

for(int i = 0; i < N; i++) c[i] = auxil::iget(5, 50);

for (int n = 10; n < 21; n++)

{

SS[n] = clock();

maxcc = knapsack\_s(V,n, v, c, m ); // измеряемая функция

SS[n]= - (SS[n] - clock());

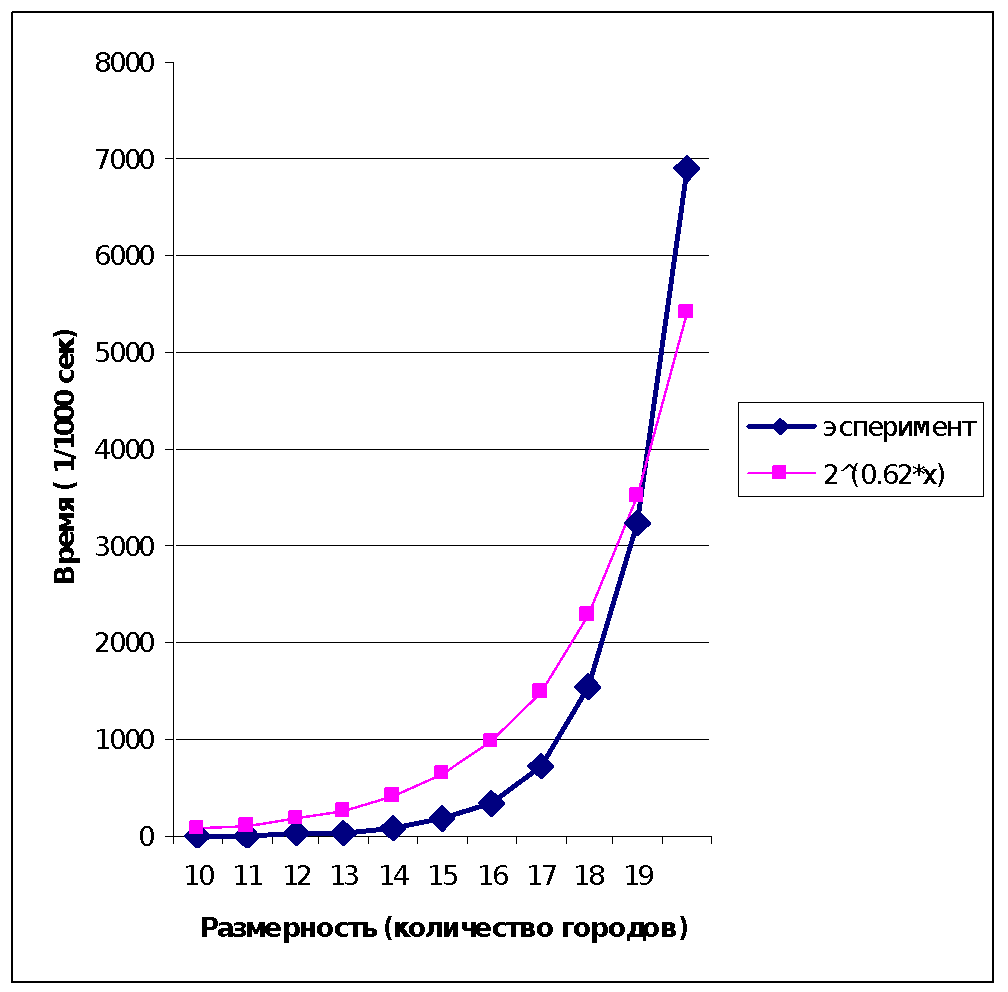
std::cout<< std::endl<<"n = "<<n << " " << SS[n];

}

system("pause");

return 0;

}



**Отчет по лабораторным работам оформите в единый документ. Он понадобится вам перед экзаменом.**