Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Дисциплина «Математическое программирование»

Отчёт по лабораторным работам

Студент: Велютич Д. И.

ФИТ 2 курс 1 группа

Минск 2024

**Лабораторная работа 1**

Вспомогательная функция

**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 (листинг 1):

|  |
| --- |
| #include "pch.h"  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 1 – реализация функций

**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.

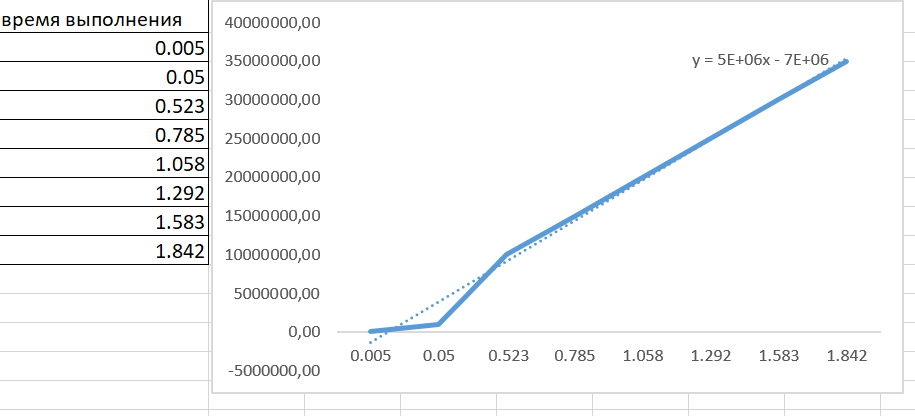
|  |
| --- |
| #include "pch.h"  #include "factorial.h"  #define CYCLE 1 // количество циклов  int main()  {  double av1 = 0, av2 = 0;  clock\_t t1 = 0, t2 = 0, t3 = 0, t4 = 0;  setlocale(LC\_ALL, "rus");  auxil::start(); // старт генерации  t1 = clock(); // фиксация времени  for (int i = 0; i < CYCLE; 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;  return 0;  } |

Листинг 2 – вызов функции и измерение времени выполнения

**Задание №3**

Проведите необходимые эксперименты и постройте график зависимости (Excel) продолжительности процесса вычисления от количества циклов в примере 2.

Примечание: продолжительность вычисления измерять в условных единицах процессорного времени (функция clock). График представлен на изображении 1:

 Изображение 1 – график зависимости

Реализация Факториала в коде демонстрируется на листинге 3:

|  |
| --- |
| #include "pch.h"  #include "factorial.h";  long double factorial(int number) {  if (number == 1) {  return 1;  }  return number \* factorial(number - 1);  } |

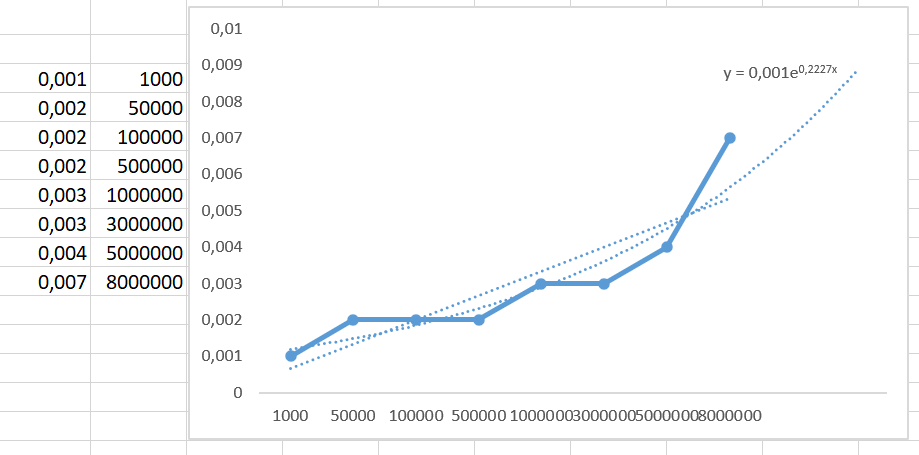
Листинг 3 – реализация факториала

Вызов и подсчет времени для реализации факториала представлен на листинге 4:

|  |
| --- |
| t3 = clock();  std::cout << '\n' << "Результат:" << factorial(100000);  t4 = clock();  std::cout << '\n' << "Время выполнения:" << ((double)(t4 - t3)) / ((double)CLOCKS\_PER\_SEC); |

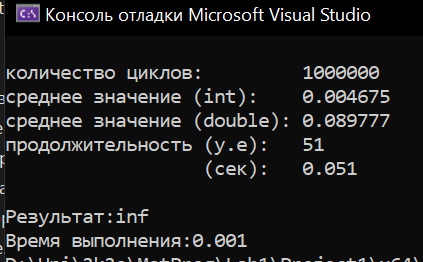
Листинг 4 - Вызов и подсчет времени для реализации факториала

График зависимости:



Изображение 2 – график зависимости для факториала

**Результаты работы программы**

****

Изображение 3 – результат работы программы

**Вывод**: по полученным измерениям, можем сделать следующий вывод, что время выполнения программы линейно зависит от количества циклов или значения факториала.