Учреждение образования

«Белорусский государственный технологический университет»

**Лабораторная работа №1**

Вспомогательные функции

Выполнил:

Студент 2 курса 4 группы ФИТ

Станчик Максим Андреевич

2024 г.

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

**Задание 1**

Используя спецификации, приведенные в условии данной лабораторной работы, мной было разработаны три функции (start, dget и iget). Результат выполнения задания приведен на рисунке 1:

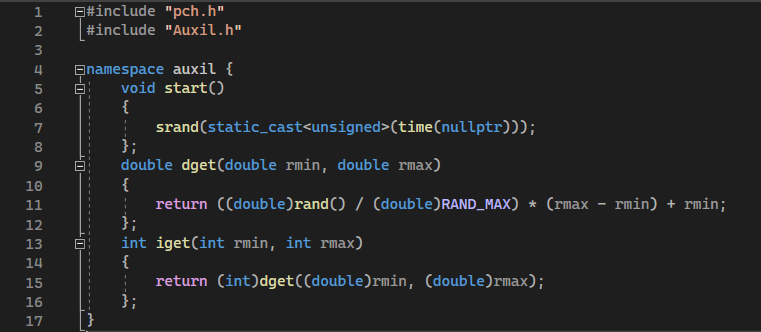


Рисунок 1 – Реализация функций

**Задание 2**

В целях проверки работоспособности разработанных функций и приобретения навыков замера продолжительности процесса вычисления мной также была реализована программа, основанная на примере, приведенном в лабораторной работе. Результат выполнения задания приведен на рисунке 2:

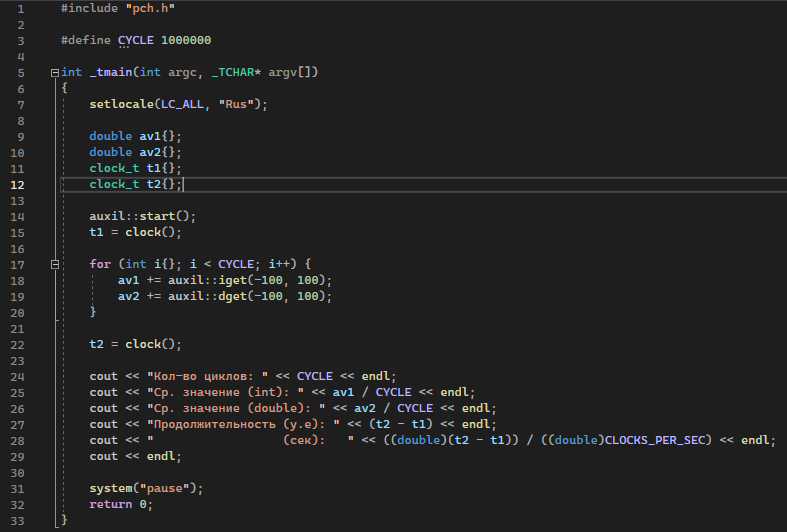


Рисунок 2 – вызов функции и измерение времени выполнения

**Задание 3**

Для выполнения задания 3 мной были проведены необходимые эксперименты и построен график зависимости продолжительности процесса вычисления от количества циклов в программе, реализованной на рисунках 1 и 2. Мной была использована программа для работы с электронными таблицами Microsoft Excel. Проанализировал характер зависимости. Результат выполнения приведен на рисунке 3:

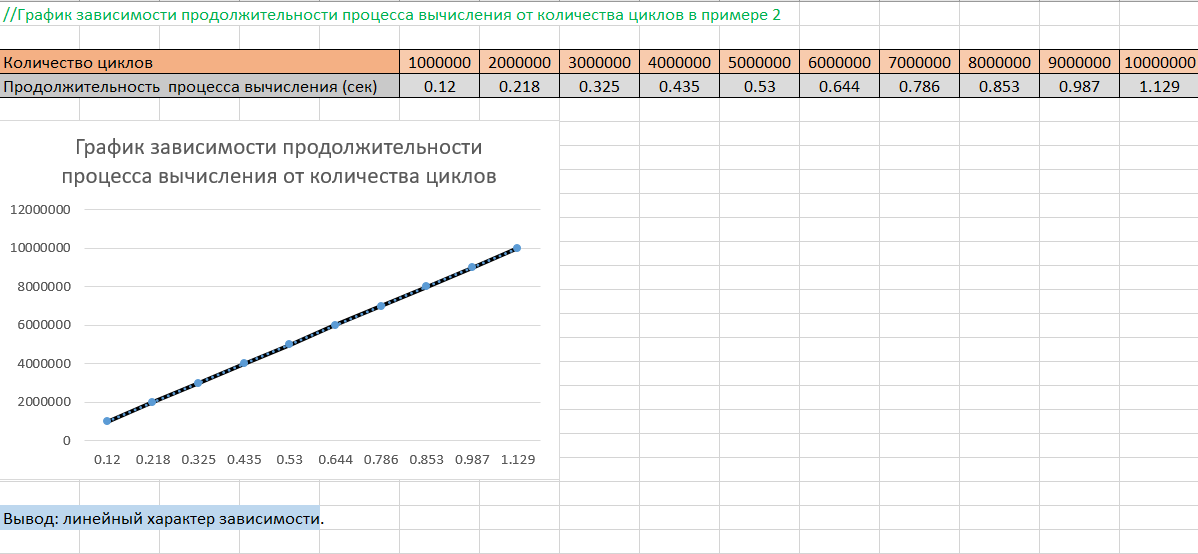


Рисунок 3 – График зависимости продолжительности процесса вычисления от количества циклов в примере 2

Также мной был реализован алгоритм для нахождения факториала. Содержимое файла, где находятся функции для работы данного алгоритма представлен на рисунке 4:

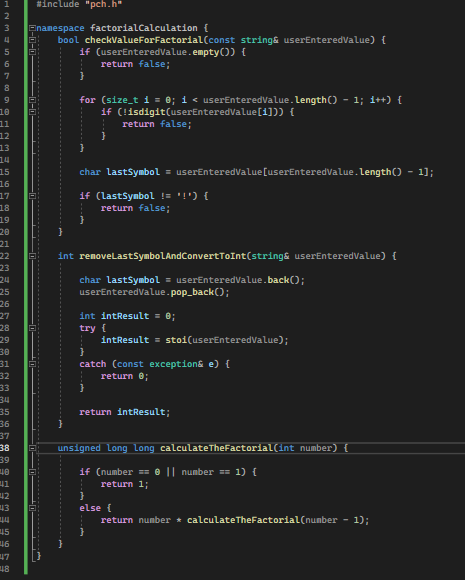


Рисунок 4 – Реализация алгоритма нахождения факториала

Вызов функций и измерение времени выполнения алгоритма находится в файле Main.cpp, после выполнения заданий 1 и 2. Содержимое файла, где находится вызов функций представлен на рисунке 5:

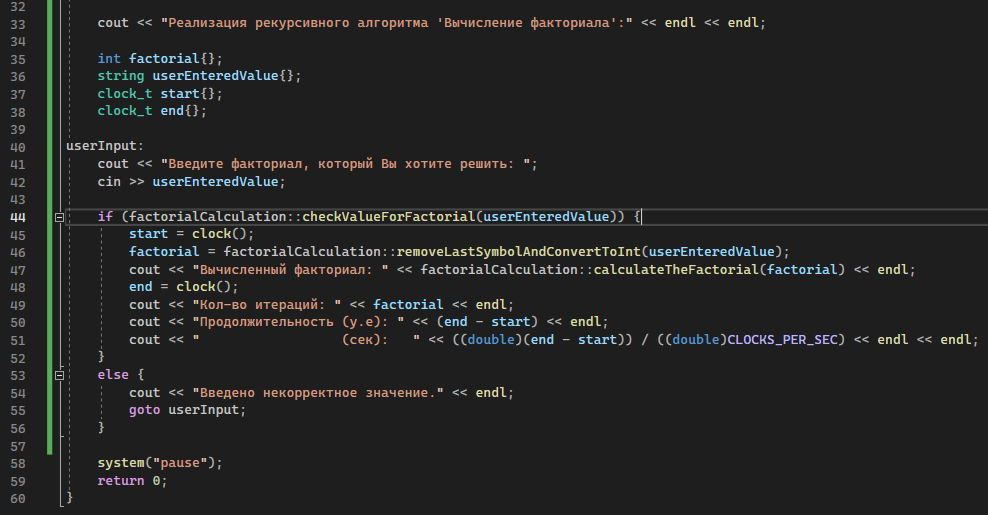


Рисунок 5 – Вызовы функций и измерение времени выполнения алгоритма

Также необходимо было построить график зависимости продолжительности процесса вычисления (сек) от количества итераций. Результат представлен на рисунке 6:

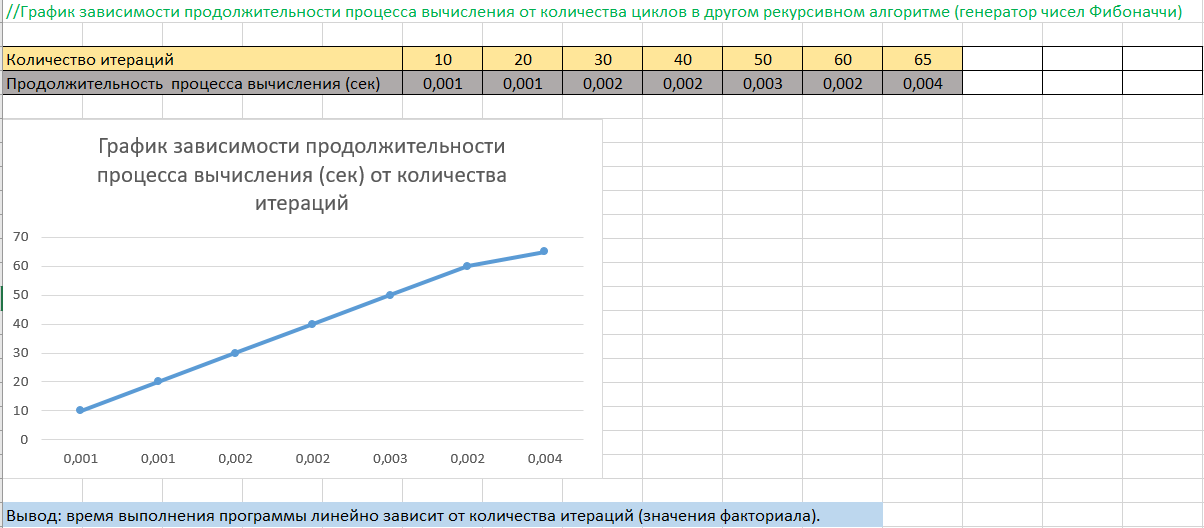


Рисунок 6 – График зависимости продолжительности процесса вычисления (сек) от количества итераций

Работоспособность программы представлена на рисунке 7:

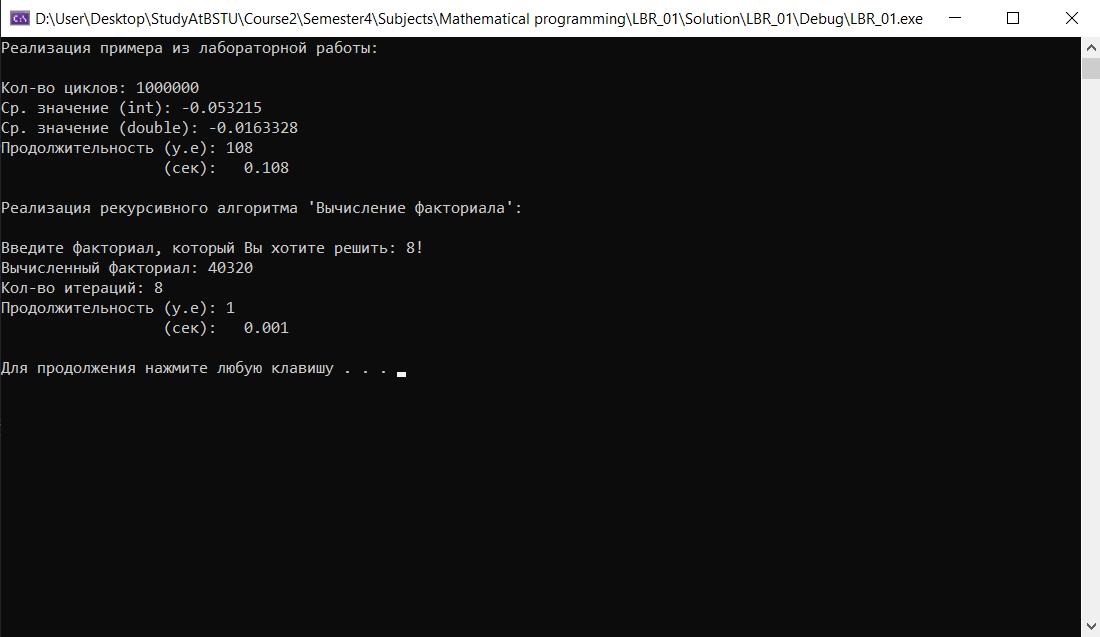


Рисунок 7 – Запуск программы

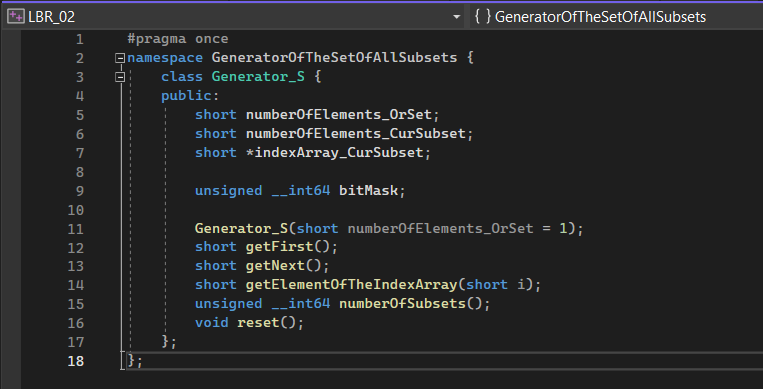
Вывод: в ходе данной лабораторной работы были использованы функции времени, которые нужны для замера чего-либо, например работы алгоритма или программы в целом, разработан алгоритм нахождения факториала, разработаны графики зависимости, первый--график зависимости продолжительности процесса вычисления от количества циклов, второй--график зависимости продолжительности процесса вычисления (сек) от количества итераций.

**Лабораторная работа №2**

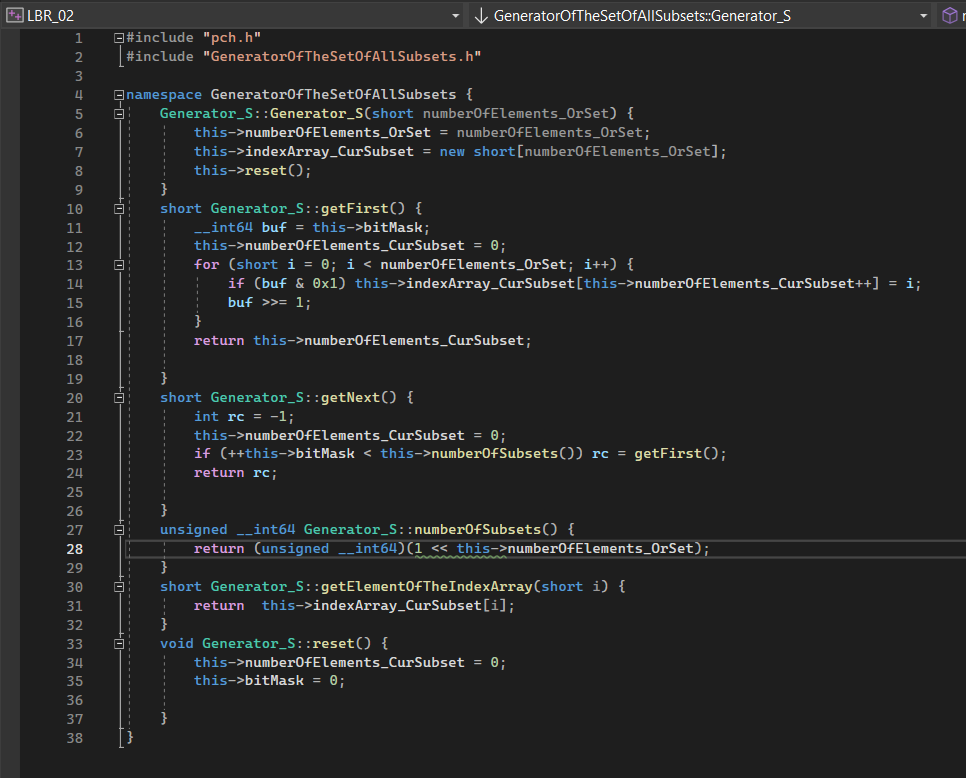
Комбинаторные алгоритмы решения оптимизационных задач

**Задание 1.** Разработать генератор подмножеств заданного множества.

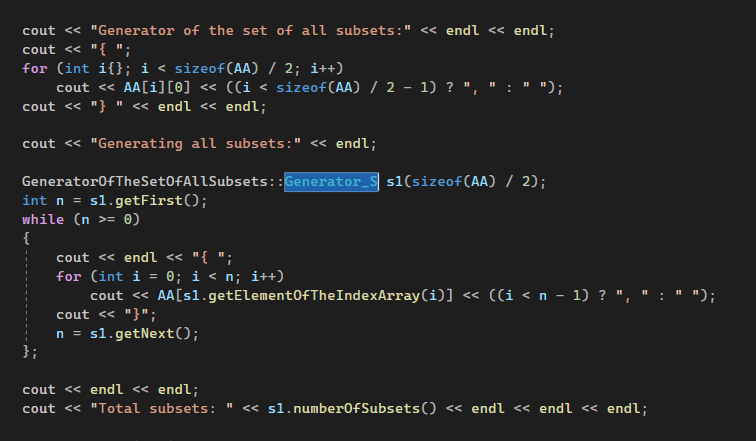
В этом задании было необходимо разработать генератор подмножеств заданного множества. Ниже на рисунках представлены заголовочные файлы и файлы cpp с реализацией поставленной задачи.



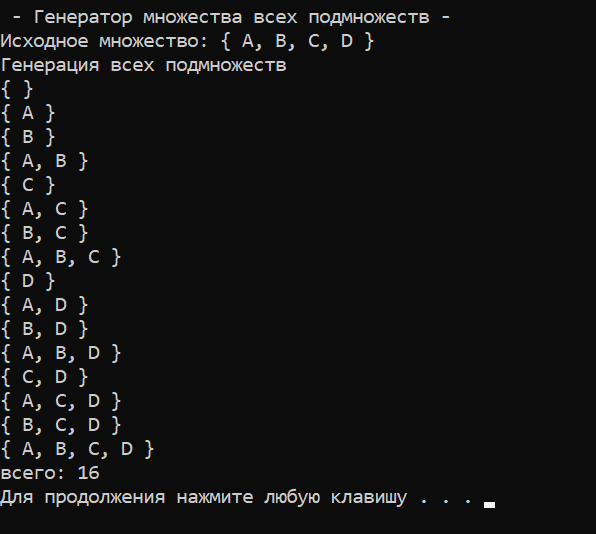
Файл GeneratorOfTheSetOfAllSubsets.h



Файл GeneratorOfTheSetOfAllSubsets.cpp



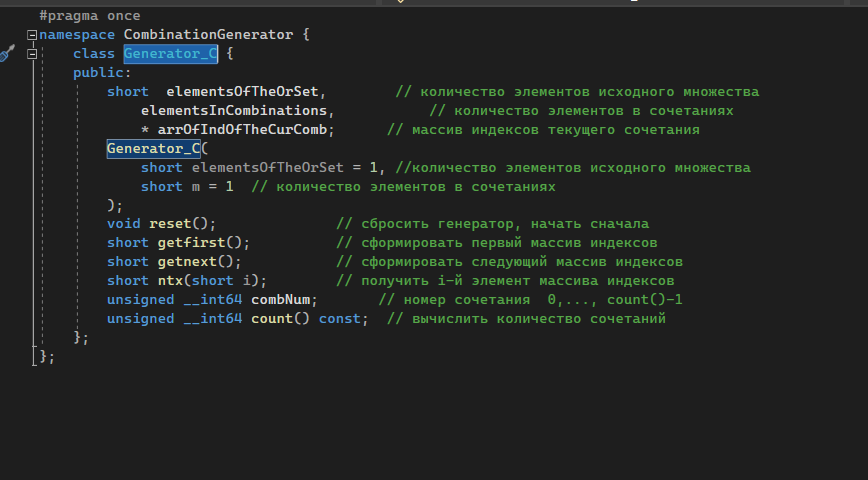
Использование в Main()



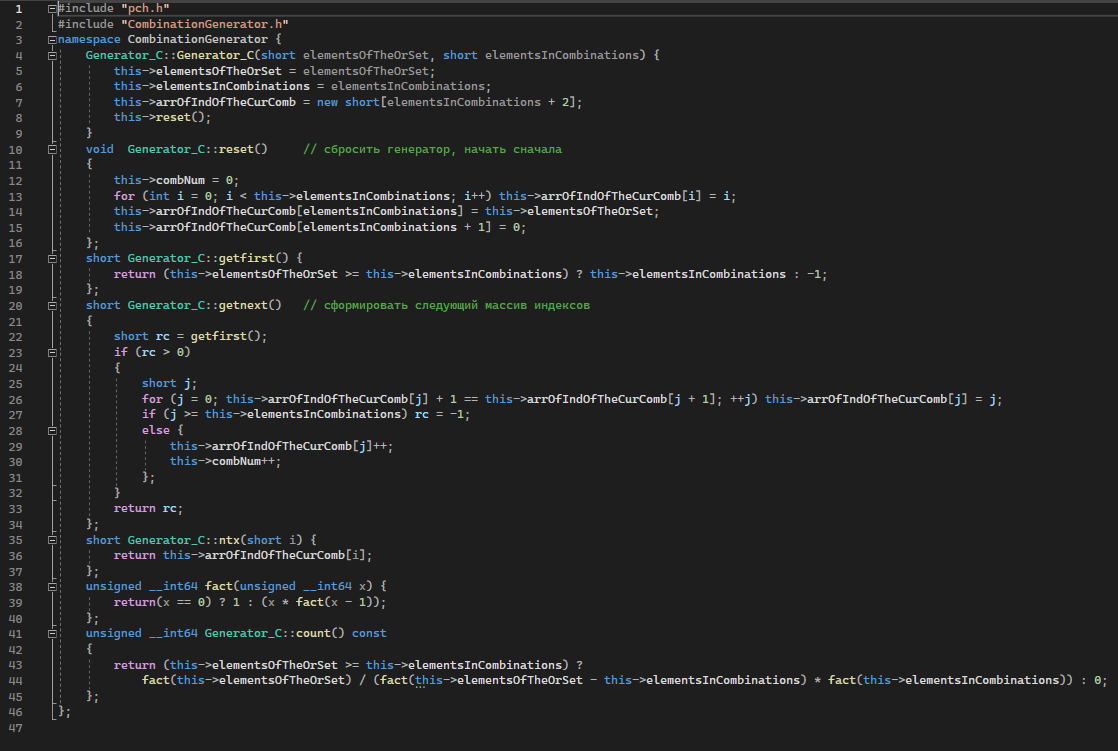
Результат выполнения

**Задание 2.** Разработать генератор сочетаний.

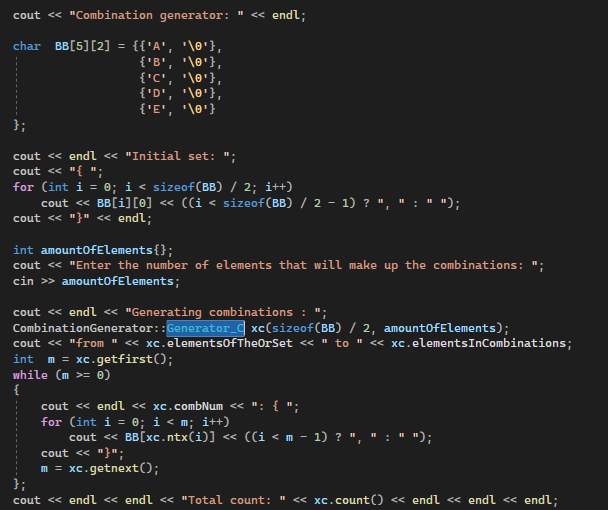
В этом задании было необходимо разработать генератор сочетаний. Ниже на рисунках представлены заголовочные файлы и файлы cpp с реализацией поставленной задачи.



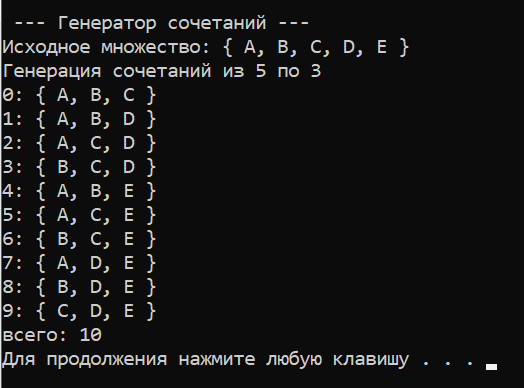
Файл CombinationGenerator.h



Файл CombinationGenerator.cpp



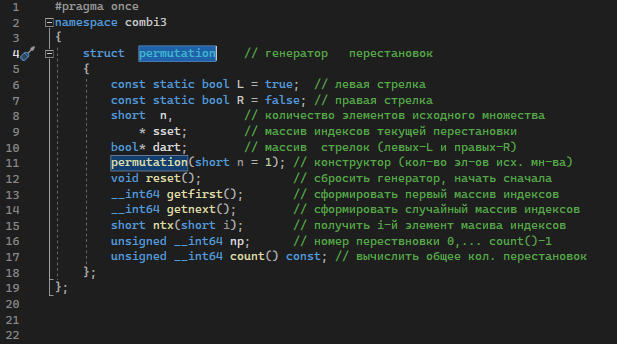
Использование в Main()



Результат выполнения

**Задание 3.** Разработать генератор перестановок.

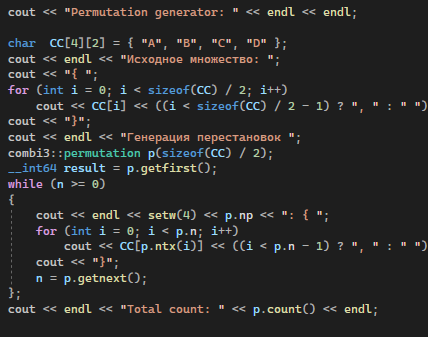
В этом задании было необходимо разработать генератор перестановок. Ниже на рисунках представлены заголовочные файлы и файлы cpp с реализацией поставленной задачи.



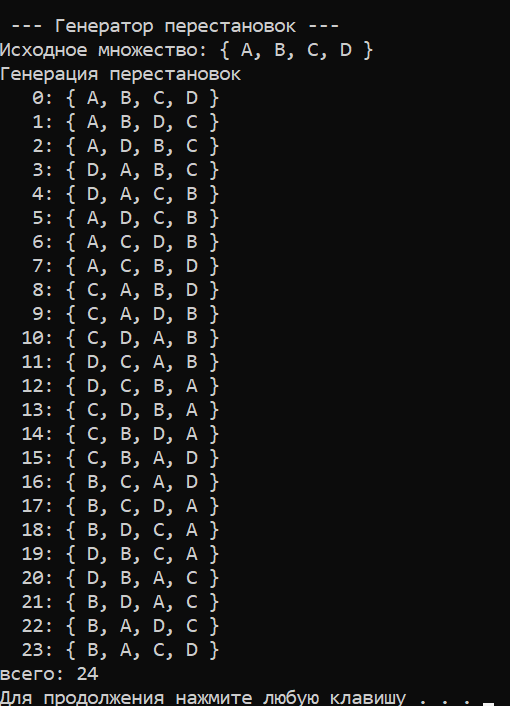
Файл PermutationGenerator.h



Файл PermutationGenerator.cpp



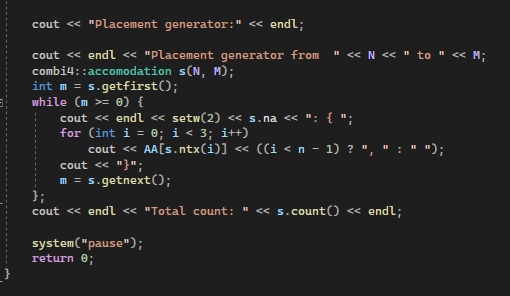
Использование в Main()



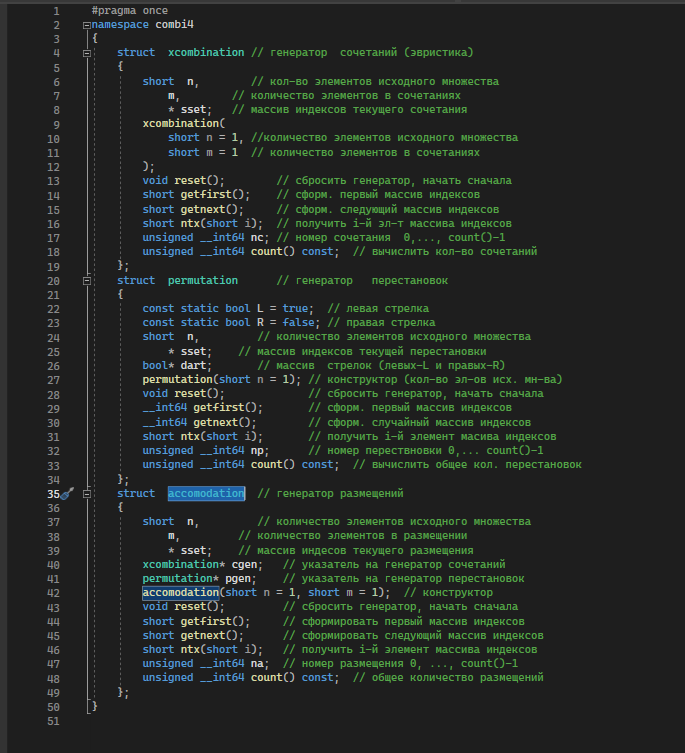
Результат выполнения

**Задание 4.** Разработать генератор размещений.

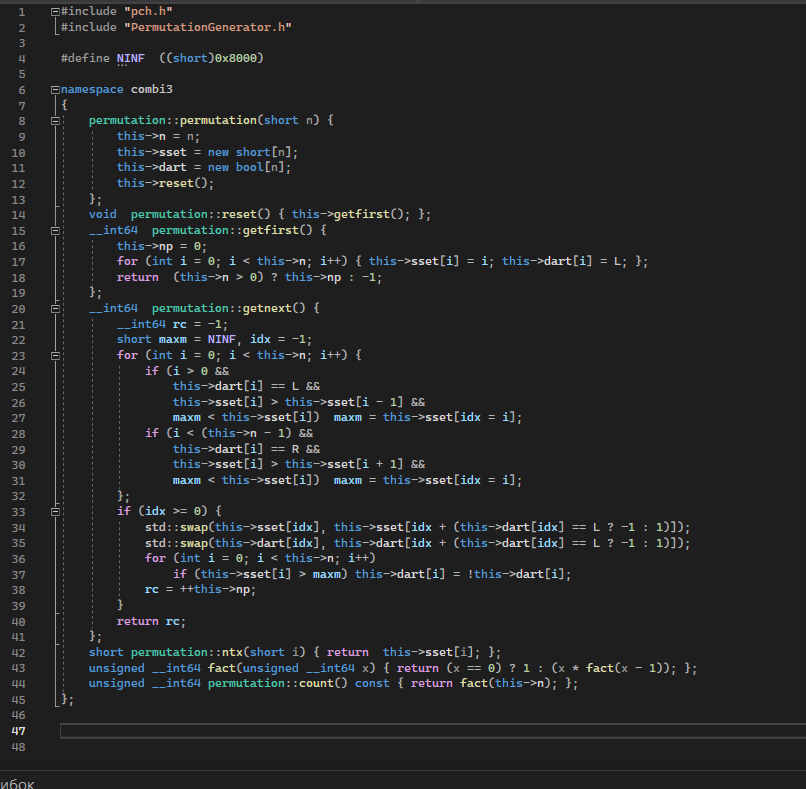
В этом задании было необходимо разработать генератор размещений. Ниже на рисунках представлены заголовочные файлы и файлы cpp с реализацией поставленной задачи.



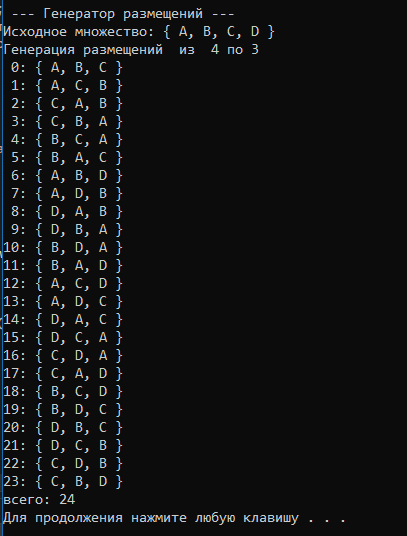
Использование генератора размещений в Main()



Файл PermutationGenerator.h



Файл PermutationGenerator.cpp



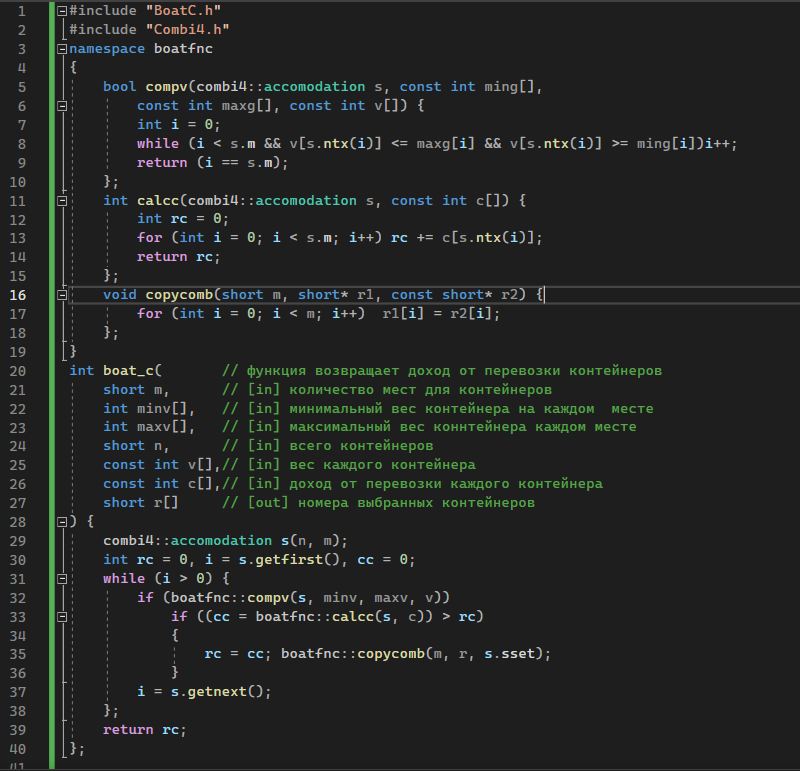
Результат выполнения

**Задание 5.** Решить в соответствии с вариантом задачу и результат занести в отчет. У меня восьмой вариант, поэтому условие задачи следующее:

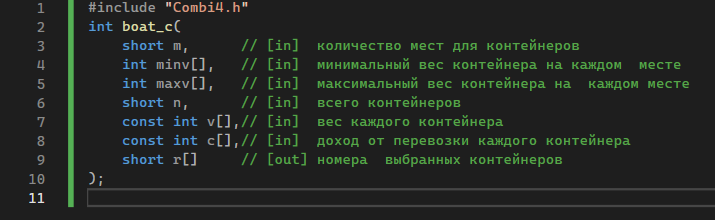
об оптимальной загрузке судна с условием центровки (веса контейнеров сгенерировать случайным образом: количество мест на судне для контейнеров – 5, количество контейнеров 8, веса контейнеров 100 – 200 кг., доход от перевозки 10 – 100 у.е.; минимальный вес контейнера для каждого места 50 – 120 кг, максимальный вес контейнера для каждого места 150 – 850 кг);

Для решения данного задания использовался генератор размещений, схема которого представлена на рисунке:





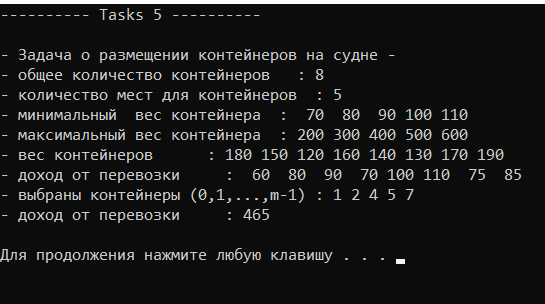
Файл BoatC.cpp



Файл BoatC.h



Использование в Main()



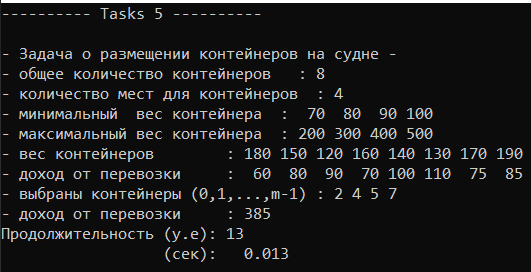
Результат решения задачи

**Задание 6.** Исследовать зависимость времени вычисления необходимое для решения задачи от размерности задачи и результат в виде графика:

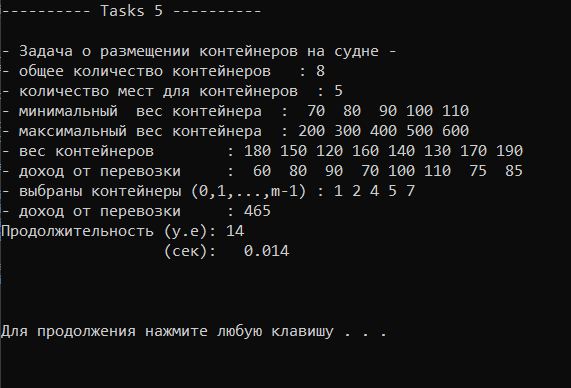
Об оптимальной загрузке судна с условием центровки (количество мест на судне для контейнеров 4 – 8);

Пусть x = количество мест на судне для контейнеров. Тогда:

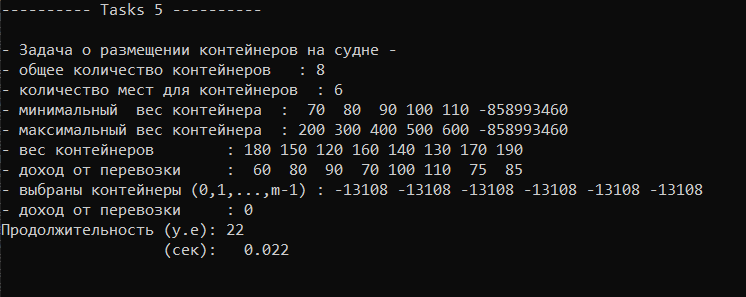
При х = 4:



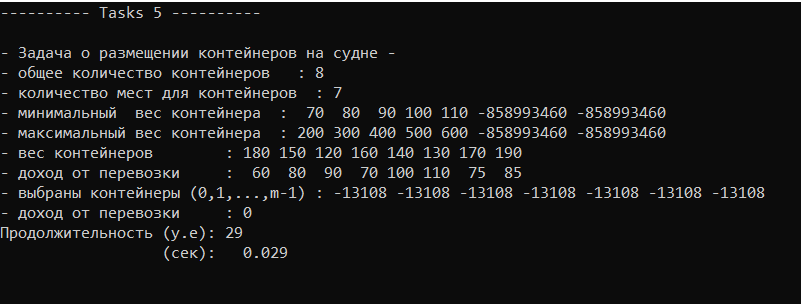
При х = 5:



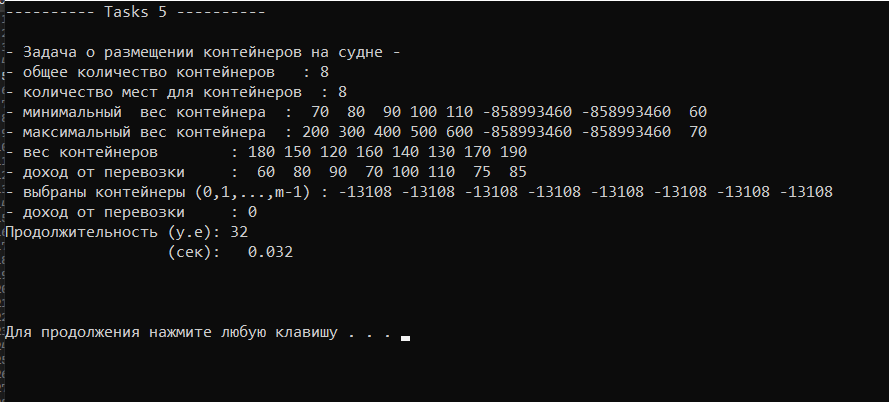
При х = 6:

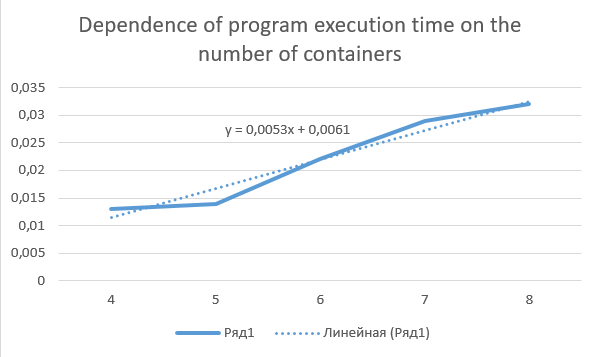


При х = 7:



При х = 8:





Вывод: исходя из полученных данных и графика, можно заметить, что скорость выполнения программы линейно возрастает при добавлении количества мест для контейнеров. Уравнение функции: y = kx +b.

//сочетание размещение перестановки и что такое генератор