**Лабораторная работа №2**

**Тема «Комбинаторные алгоритмы решения оптимизационных задач»**

**Цель**: приобрести навыки разработки генераторов подмножеств, перестановок, сочетаний и размещений на С++; научиться применять разработанные генераторы для решения задач о рюкзаке (упрощенную, коммивояжера, об оптимальной загрузке судна и об оптимальной загрузке судна с центровкой.

**Ход работы:**

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

Рисунок 1.1 – Шаблон структуры генератора множества всех подмножеств (combi1.h)

Рисунок 1.2 – Реализация методов структуры *subset* (combi1.cpp).

Рисунок 1.3 – Пример применения генератора множества всех подмножеств.

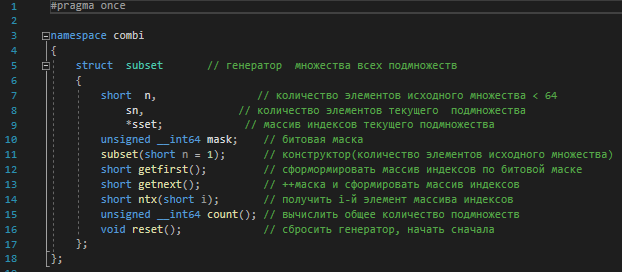
Рисунок 1.4 – Результат выполнения программы.

Рисунок 1.1

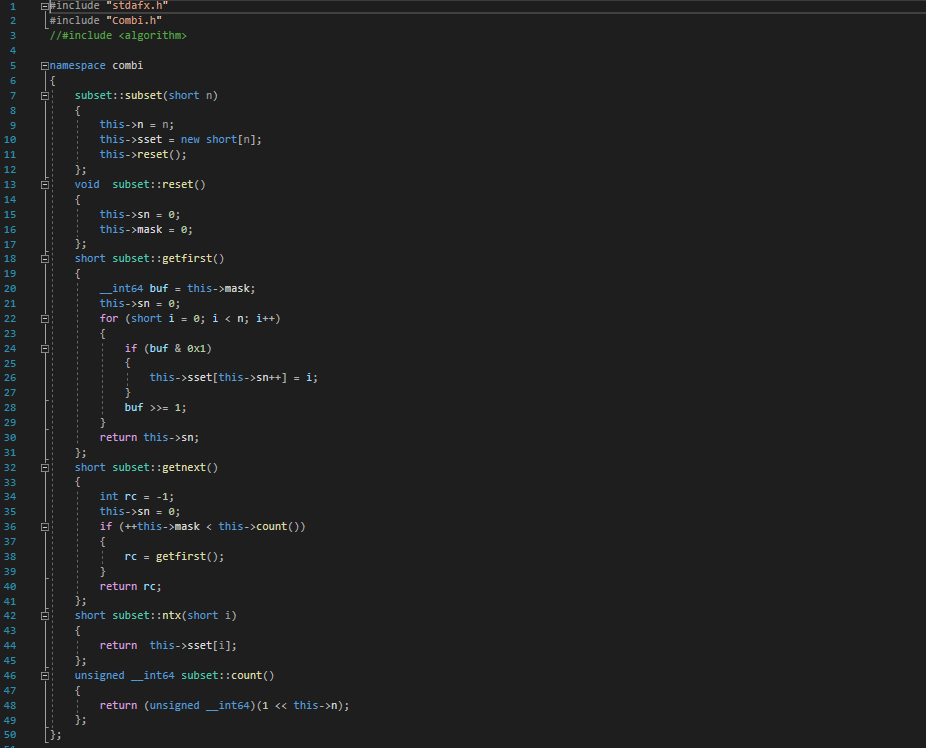
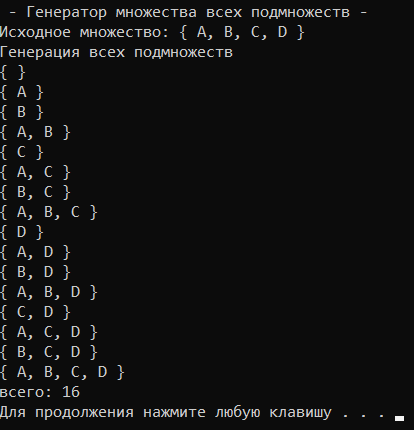
Рисунок 1.2

Рисунок 1.3

 Рисунок 1.4

1. **Разработать генератор сочетаний**

Рисунок 2.1 – Шаблон структуры генератора сочетаний.

Рисунок 2.2 – Реализация функций генератора сочетаний.

Рисунок 2.3 – Main.

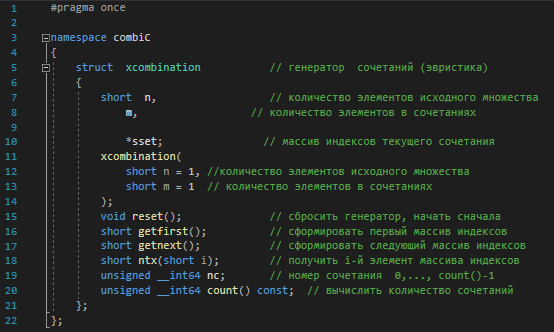
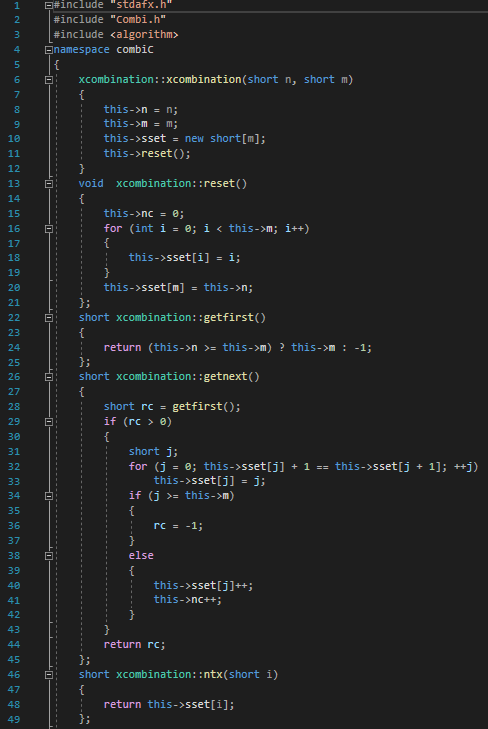
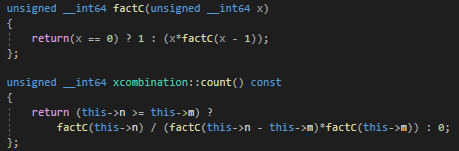
Рисунок 2.4 – Результат.

Рисунок 2.1

Рисунок 2.2

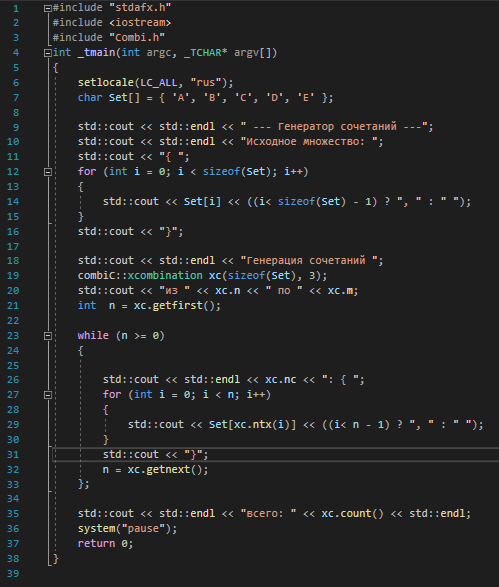


Рисунок 2.3

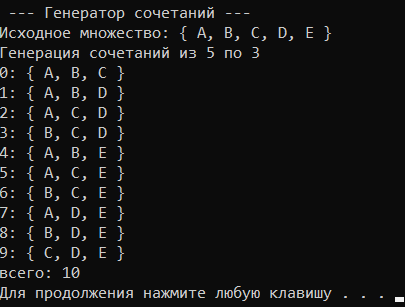


Рисунок 2.4

1. **Разработать генератор перестановок**

Рисунок 3.1 – Шаблон структуры генератора перестановок.

Рисунок 3.2, 3.3 – Реализация функций генератора перестановок.

Рисунок 3.4 – Пример применения генератора перестановок.

Рисунок 3.5 – Результат.

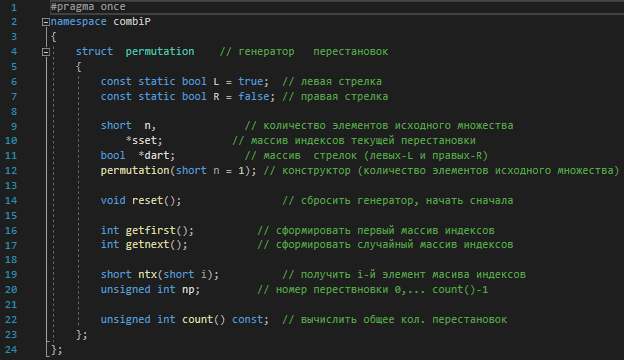
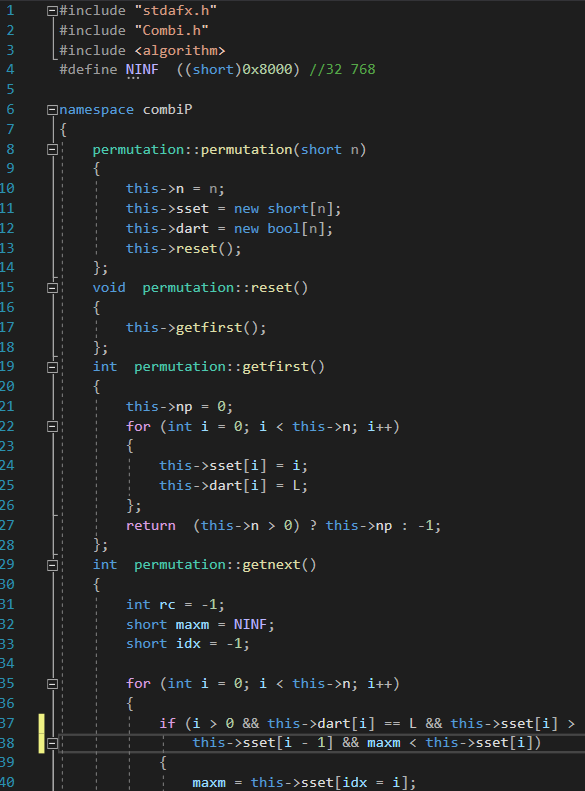
Рисунок 3.1

Рисунок 3.2

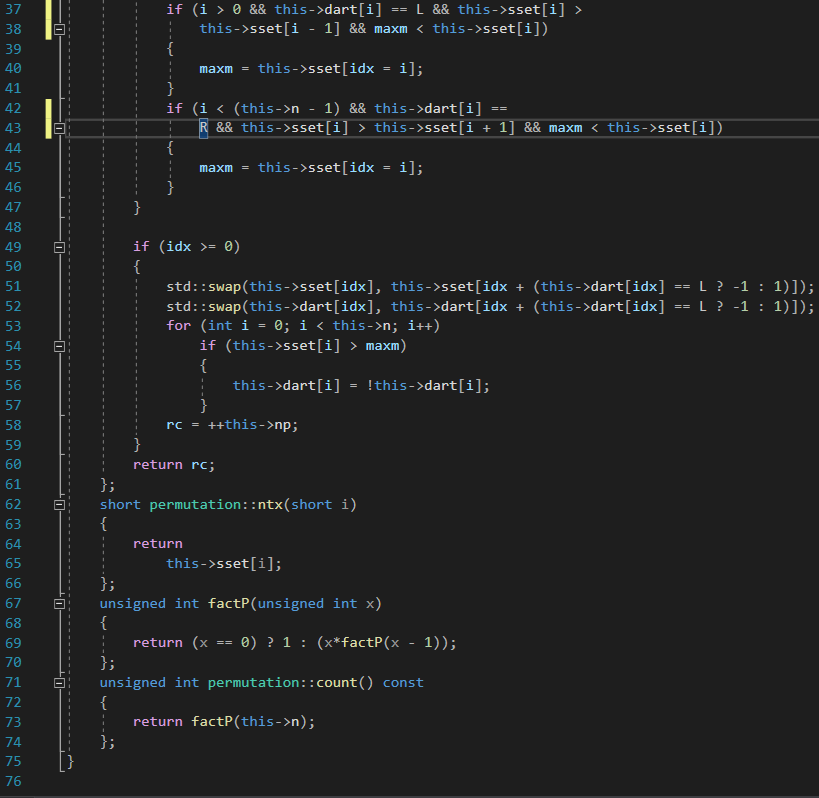
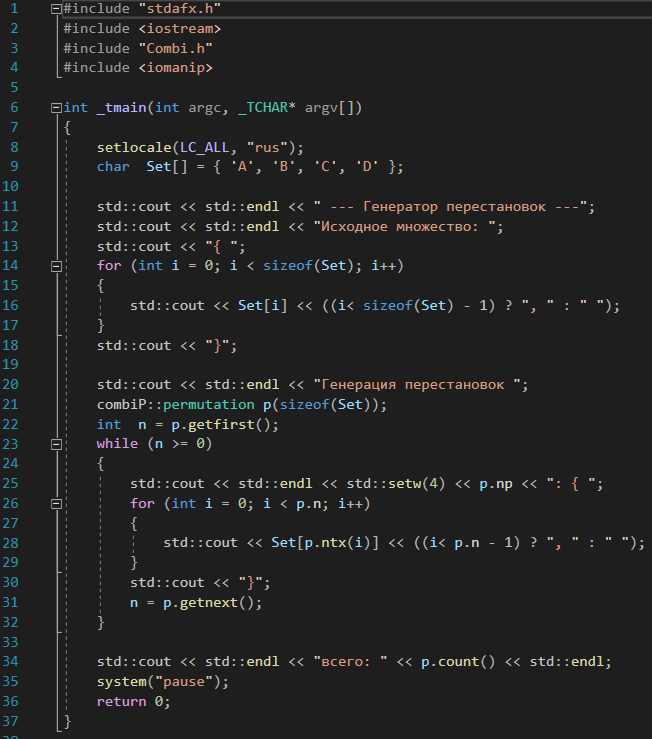
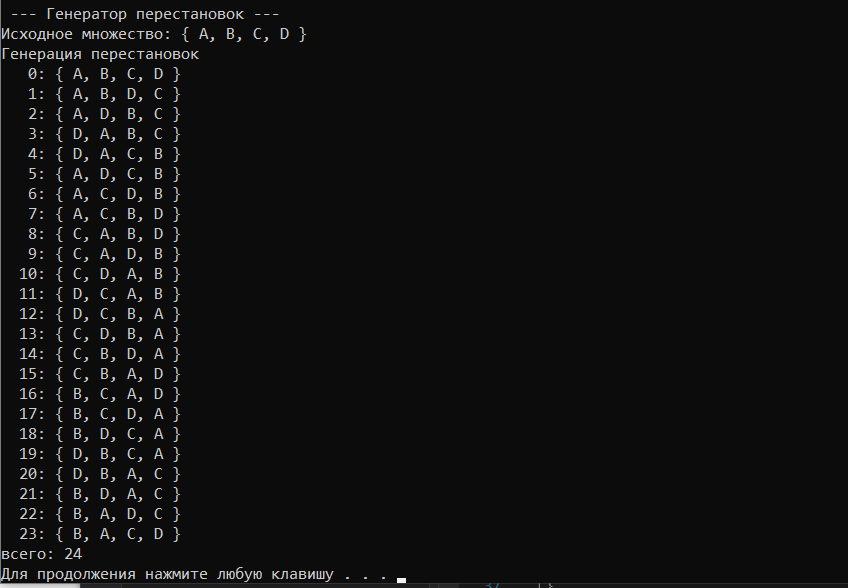
Рисунок 3.3

Рисунок 3.4

Рисунок 3.5

1. **Разработать генератор размещений**

Рисунок 4.1 – Шаблон структуры генератора размещений.

Рисунок 4.2, 4.3 – Реализация функций генератора размещений.

Рисунок 4.4 – Пример использования генератора размещений (Accommodation.cpp).

Рисунок 4.5, 4.6 – (Sochetaniya.cpp).

Рисунок 4.7, 4.8 – (Perestanovki.cpp).

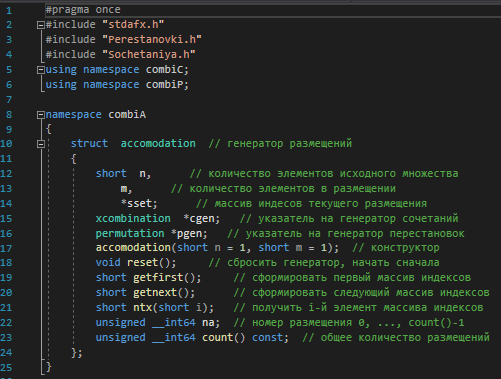
Рисунок 4.9 – Результат.

Рисунок 4.1

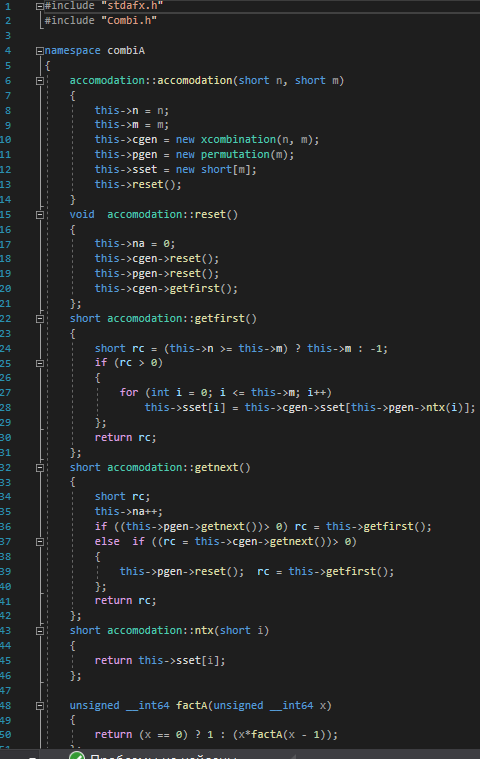
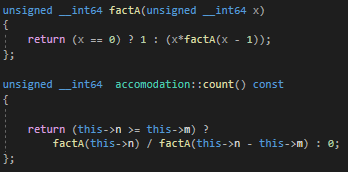
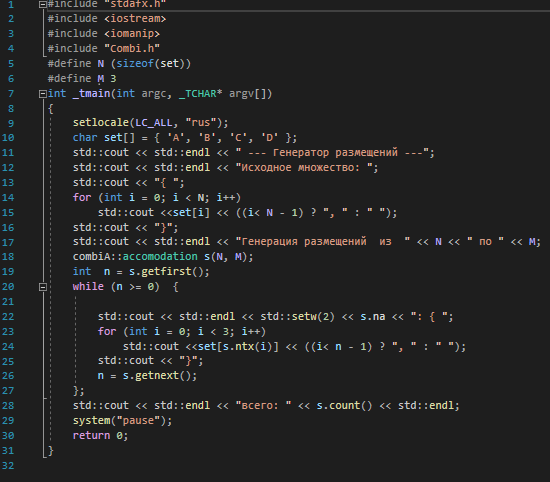
Рисунок 4.2

Рисунок 4.3

Рисунок 4.4

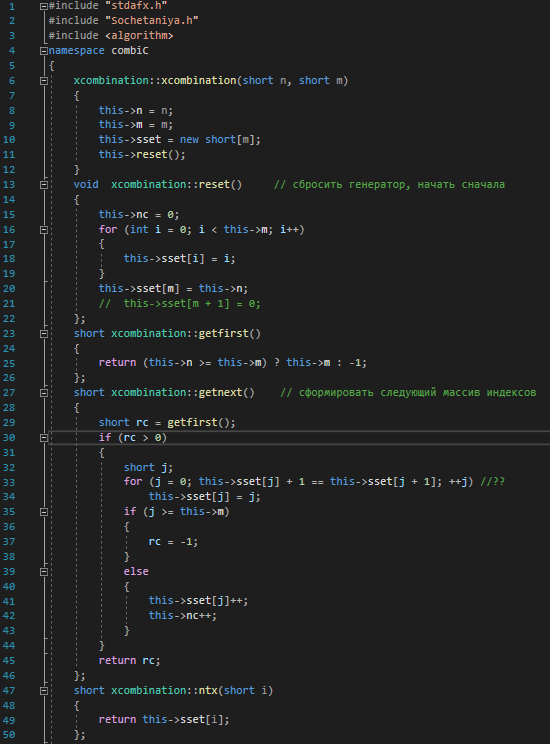


Рисунок 4.5

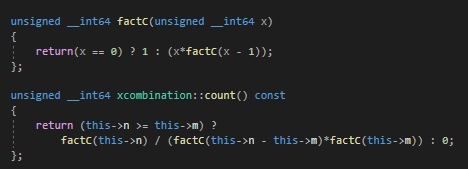


Рисунок 4.6

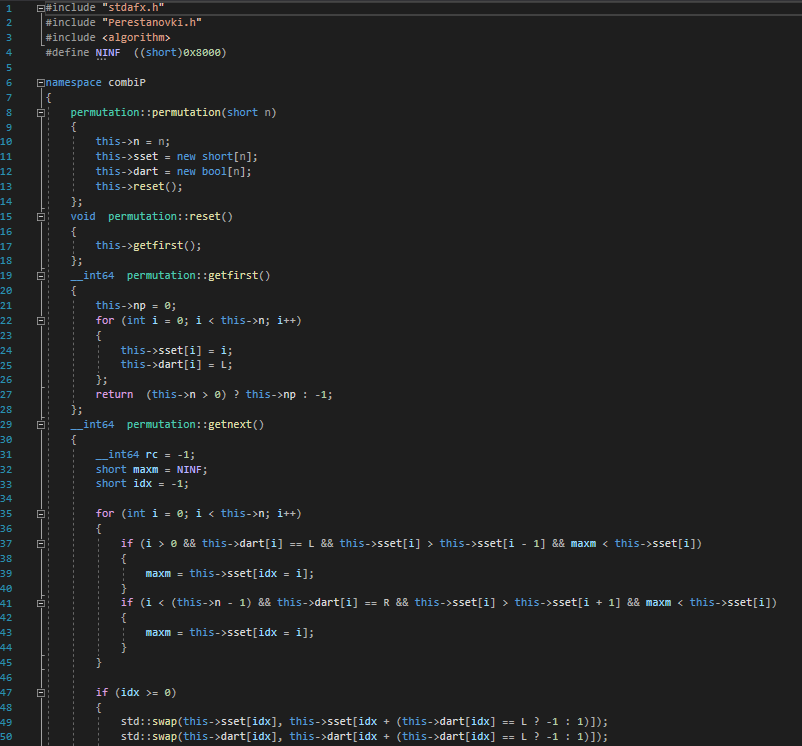


Рисунок 4.7

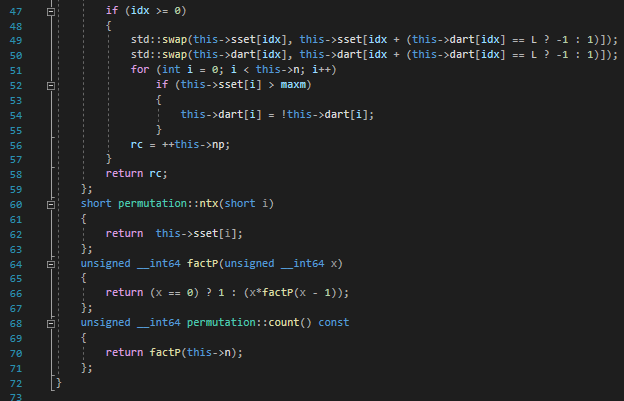


Рисунок 4.8

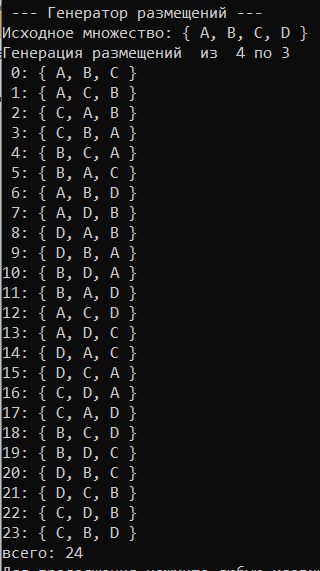


Рисунок 4.9

1. **Решить в соответствии с вариантом задачу:**

(Вариант 1, 5, 9, 13) коммивояжера (расстояния сгенерировать случайным образом: 10 городов, расстояния 10 – 300 км, 3 расстояния между городами задать бесконечными);

Рисунок 5.1 – Функция salesman.

Рисунок 5.2, 5.3 – Реализация функции salesman.

Рисунок 5.4, 5.5 – Пример решения задачи коммивояжера.

Рисунок 5.6 – Результат.

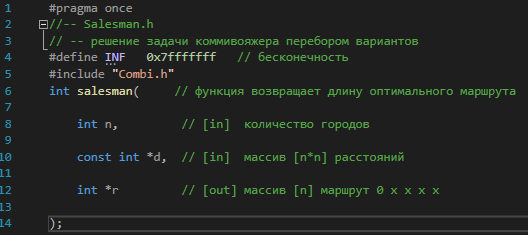


Рисунок 5.1

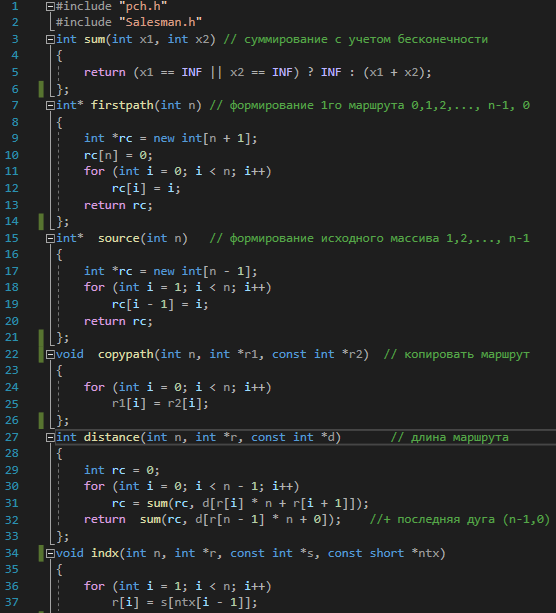


Рисунок 5.2

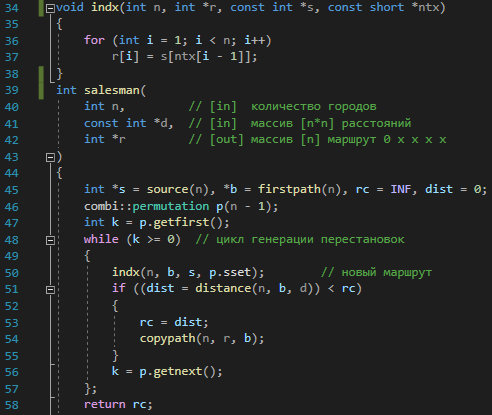


Рисунок 5.3

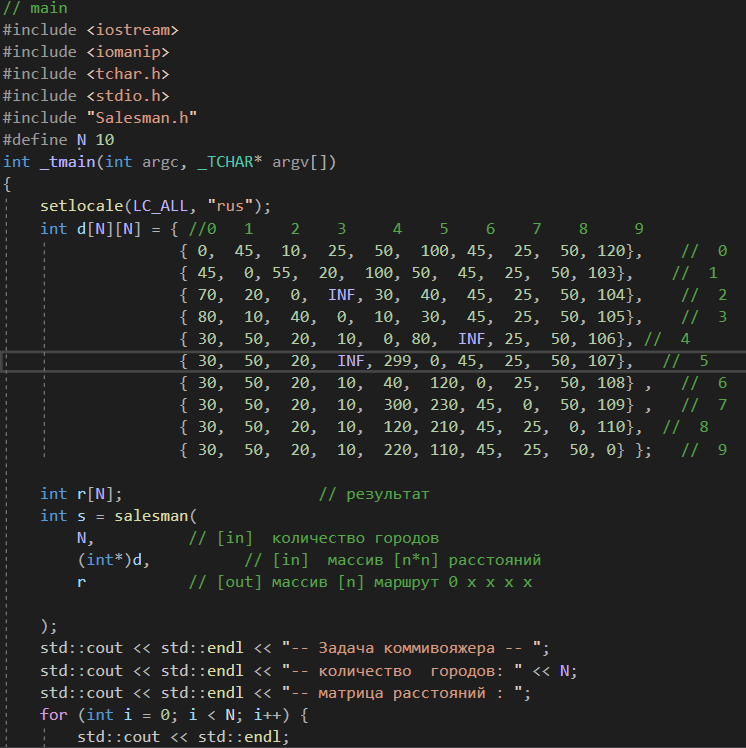
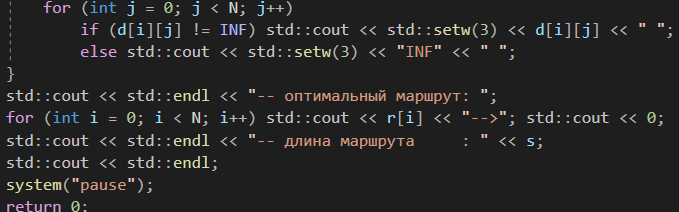


Рисунок 5.4

Рисунок 5.5

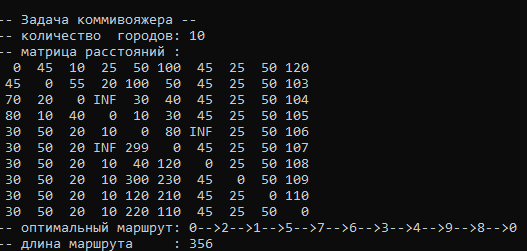
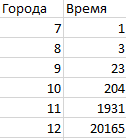


Рисунок 5.6

1. Исследовать зависимость времени вычисления необходимое для решения задачи (в соответствии с вариантом) от размерности задачи: (Вариант 1, 5, 9, 13) коммивояжера (5–12 городов)

Рисунок 6.1 – Зависимость времени от количества городов.

Рисунок 6.2 – Зависимость на графике.

Рисунок 6.1

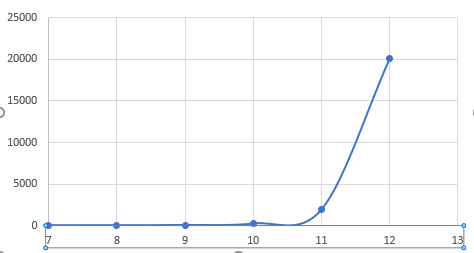


Рисунок 6.2

**Вывод**: в ходе лабораторной работы замерил продолжительность выполнения программы с помощью составления и отладки программ и заметил, что время вычисления, необходимое для решения задачи коммивояжера, резко возросло, когда начало возрастать количество городов.