**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по лабораторной работе №2**

**по дисциплине «Программирование»**

Тема: Одномерные статические массивы

|  |  |  |
| --- | --- | --- |
| Студент(ка) гр. 1323 |  | Захарова Е.В. |
| Преподаватель |  | Глущенко А.Г. |

Санкт-Петербург

2021

**Цель работы.**

Необходимо написать программу, которая:

1)    Создает целочисленный массив размерности *N* = 100. Элементы массивы должны принимать случайное значение в диапазоне от -99 до 99.

2)    Отсортировать заданный в пункте 1 массив […] сортировкой (от меньшего к большему). Определить время, затраченное на сортировку, используя библиотеку chrono.

3)    Найти максимальный и минимальный элемент массива. Подсчитайте время поиска этих элементов в отсортированном массиве и неотсортированном, используя библиотеку chrono.

4)    Выводит среднее значение (если необходимо, число нужно округлить) максимального и минимального значения. Выводит индексы всех элементов, которые равны этому значению, и их количество.

5)    Выводит количество элементов в отсортированном массиве, которые меньше числа *a*, которое инициализируется пользователем.

6)    Выводит количество элементов в отсортированном массиве, которые больше числа *b*, которое инициализируется пользователем.

7)    Выводит информацию о том, есть ли введенное пользователем число в отсортированном массиве. Реализуйте алгоритм бинарного поиска. Сравните скорость его работы с обычным перебором.

8)     Меняет местами элементы массива, индексы которых вводит пользователь. Выведите скорость обмена, используя библиотеку chrono.

Должна присутствовать возможность запуска каждого пункта многократно.

**Основные теоретические положения.**

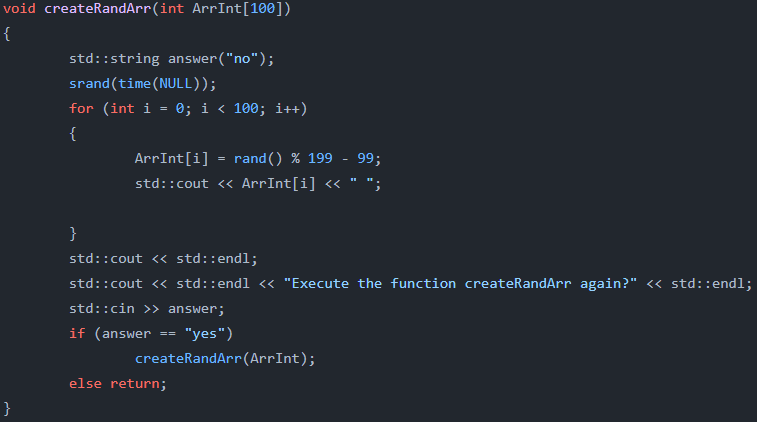


Рисунок 1 (Функция для создания массива из случайных чисел)

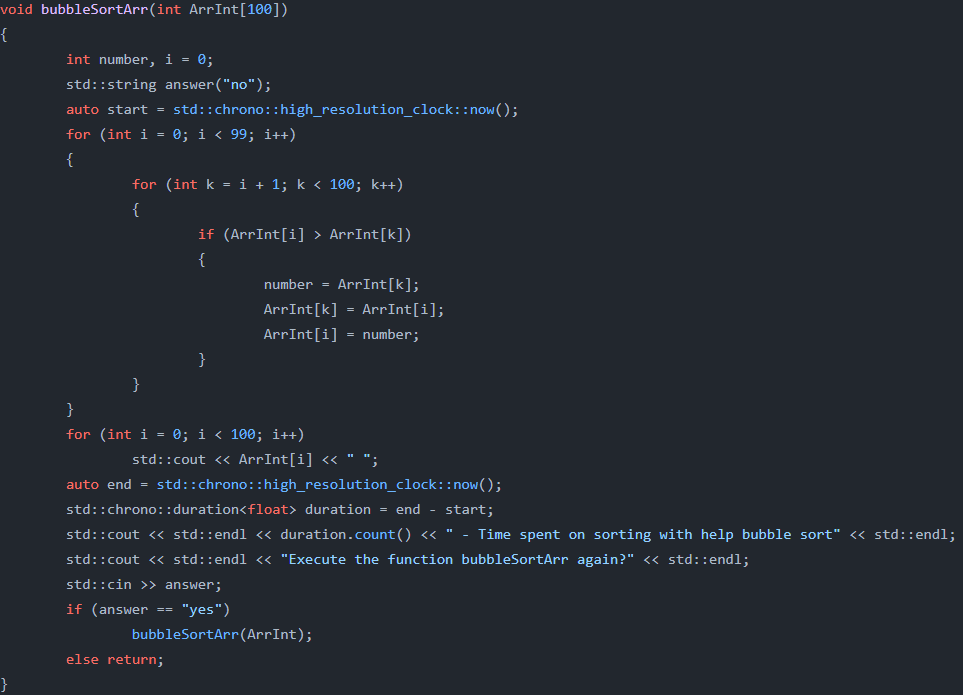


Рисунок 2 (Функция пузырьковой сортировки)



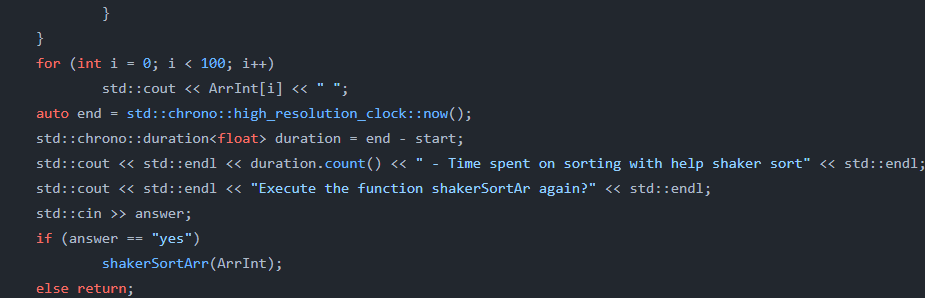


Рисунок 3 (Функция сортировки shaker)

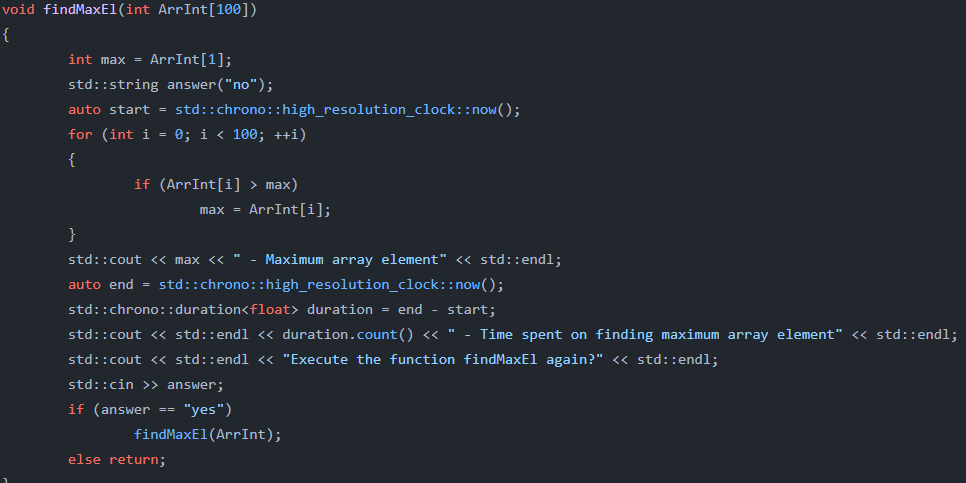


Рисунок 4 (Функция для нахождения максимального элемента в неотсортированном массиве)

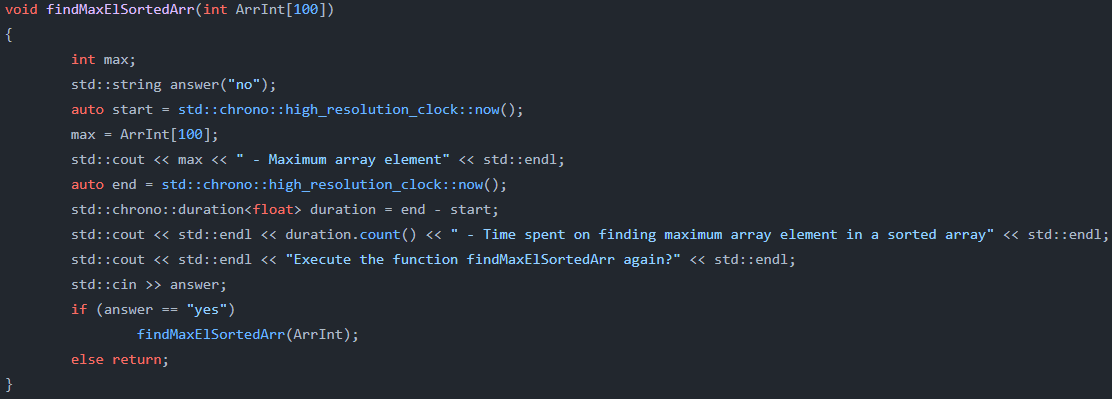


Рисунок 5 (Функция для нахождения максимального элемента в отсортированном массиве)

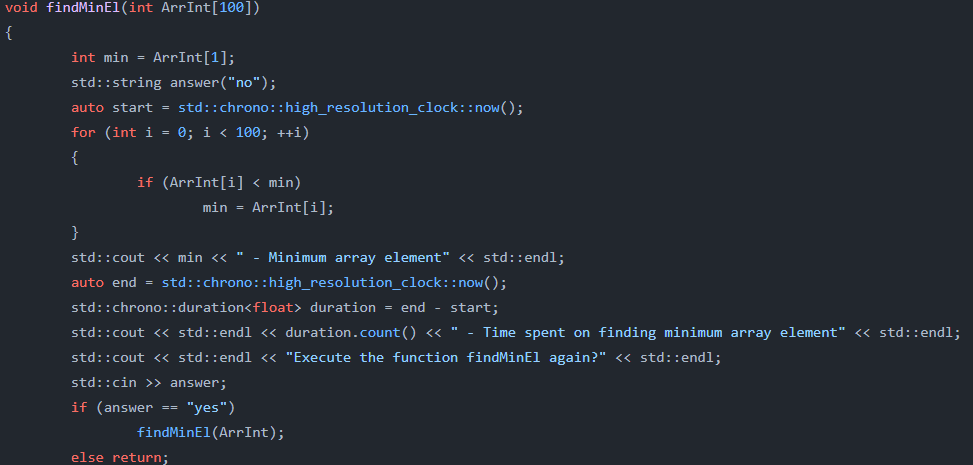


Рисунок 6 (Функция для нахождения минимального элемента в неотсортированном массиве)

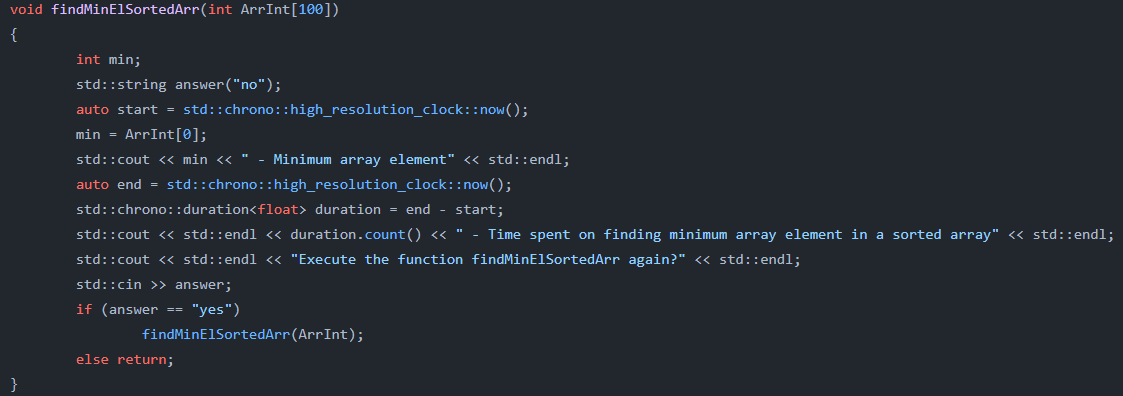


Рисунок 7 (Функция для нахождения минимального элемента в отсортированном массиве)

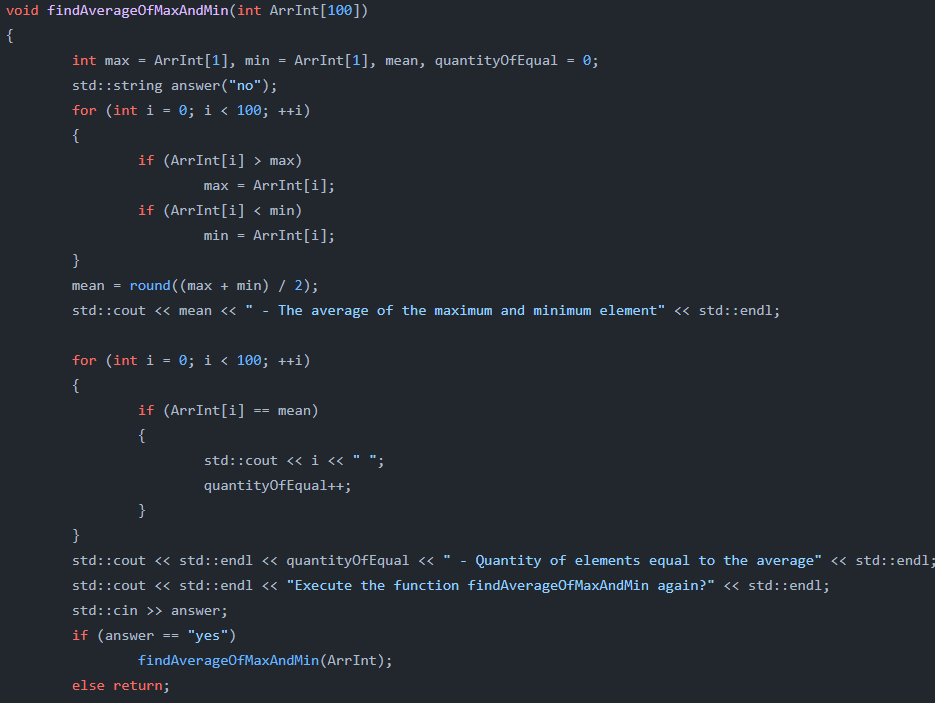


Рисунок 8 (Функция для нахождения среднего значения между максимальным и минимальным значением в массиве и определения количества таких элементов в массиве)

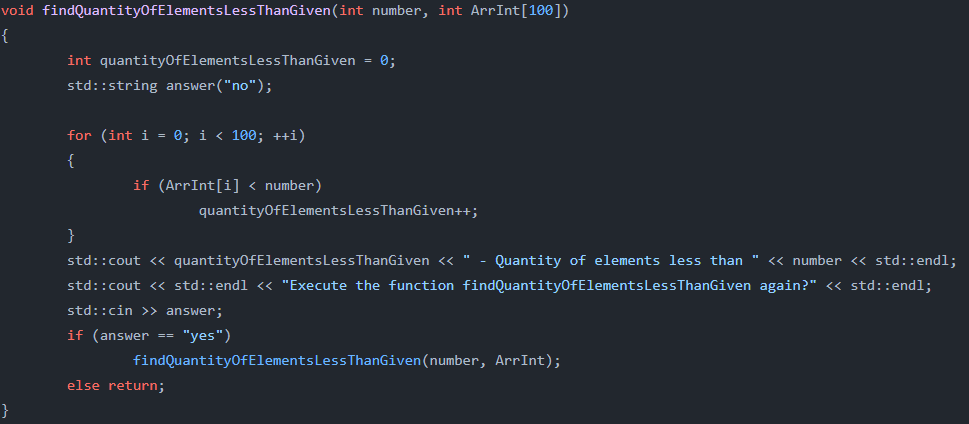


Рисунок 9 (Функция для нахождения количества элементов меньше введенного с клавиатуры)

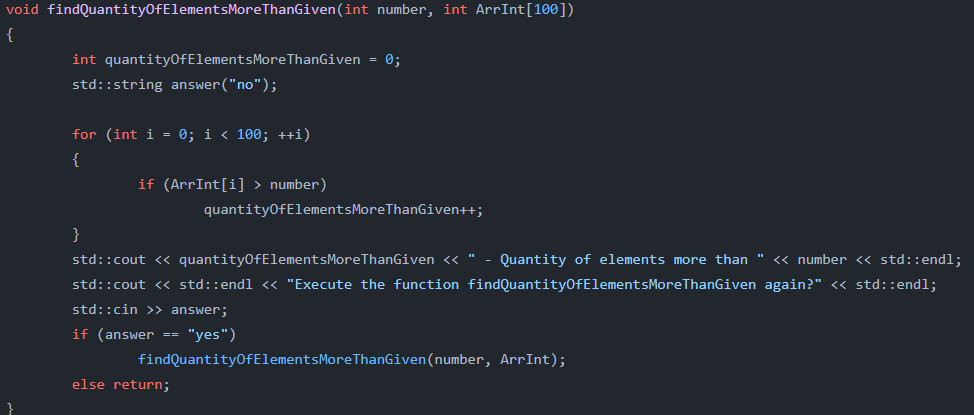


Рисунок 10 (Функция для нахождения количества элементов больше введенного с клавиатуры)

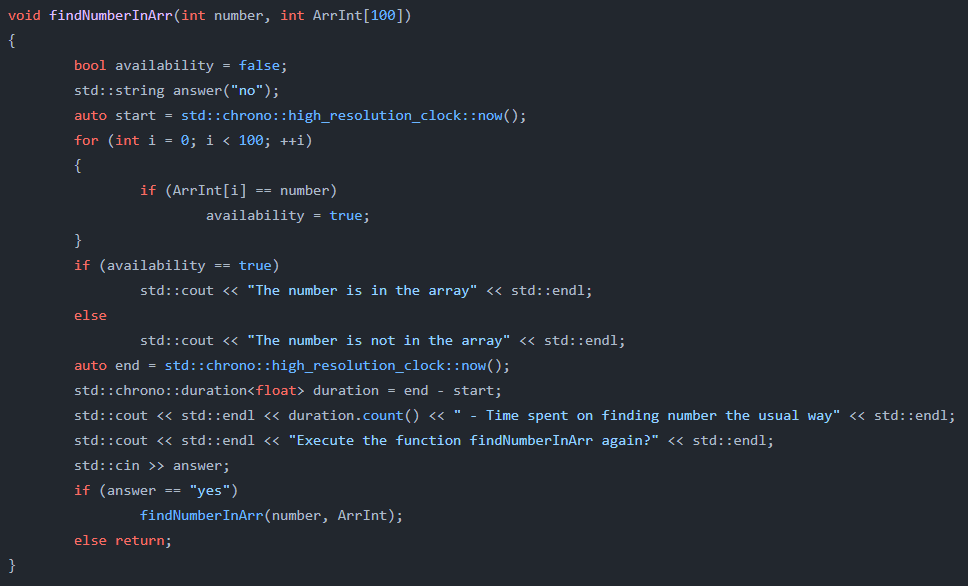


Рисунок 11 (Функция поиска заданного с клавиатуры числа в массиве обычным способом)

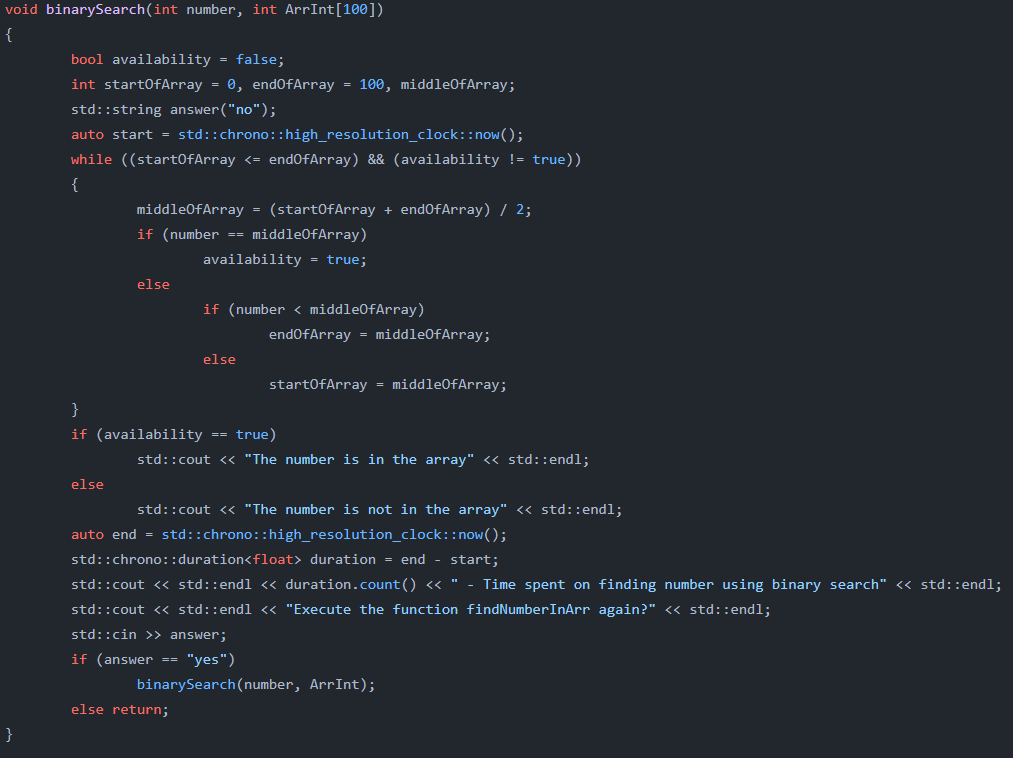


Рисунок 12 (Функция поиска заданного с клавиатуры числа в массиве при помощи бинарного поиска)

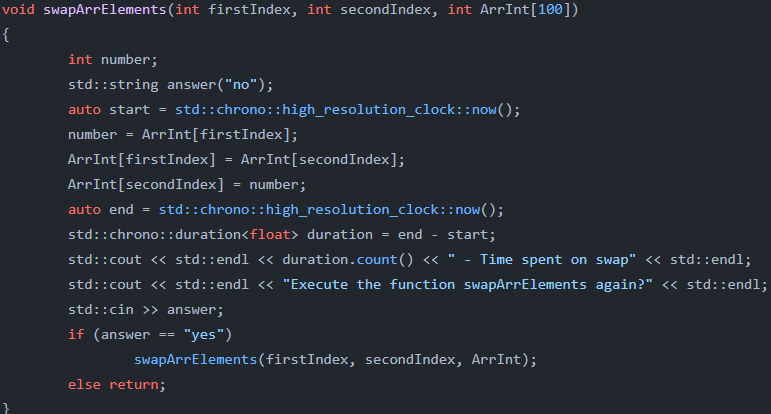


Рисунок 13 (Функция, производящая обмен двух элементов массива местами)



Рисунок 14 (Основная программа)

**Экспериментальные результаты.**



Рисунок 15 (Результат работы функции по созданию массива из случайных чисел)

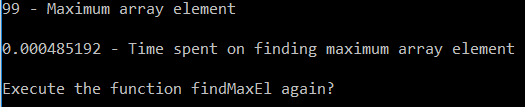


Рисунок 16 (Результат работы функции по поиску максимального элемента в неотсортированном массиве)

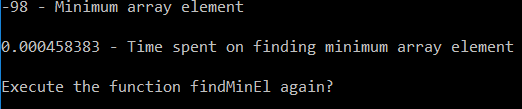


Рисунок 17 (Результат работы функции по поиску минимального элемента в неотсортированном массиве)

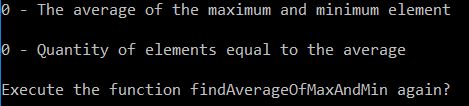


Рисунок 18 (Результат работы функции по поиску среднего значения между минимальным и максимальным элементом и нахождению количества таких же элементов в массиве)

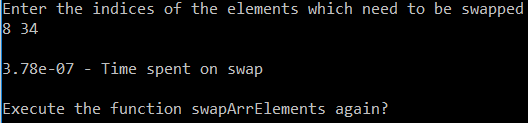


Рисунок 19 (Результат работы функции по обмену двух элементов массива местами)

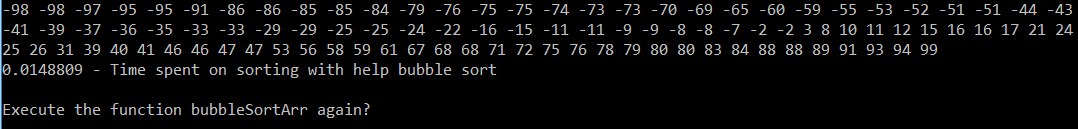


Рисунок 20 (Результат работы функции пузырьковой сортировки)

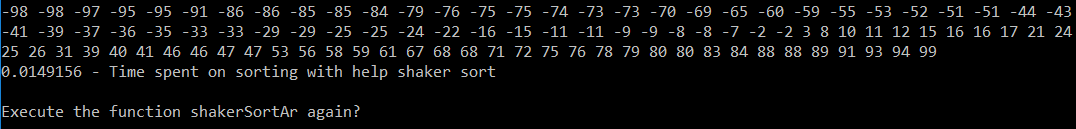


Рисунок 21 (Результат работы функции сортировки shaker)

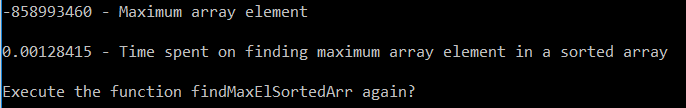


Рисунок 22 (Результат работы функции по поиску максимального элемента в отсортированном массиве)

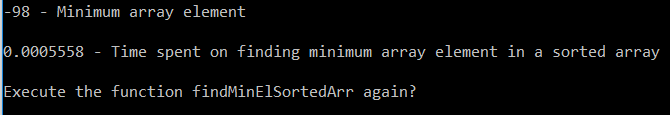


Рисунок 23 (Результат работы функции по поиску минимального элемента в отсортированном массиве)

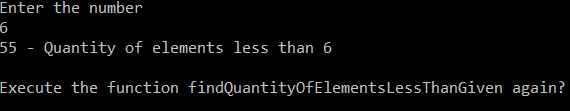


Рисунок 24 (Результат работы функции по нахождению количества элементов меньше введенного с клавиатуры)

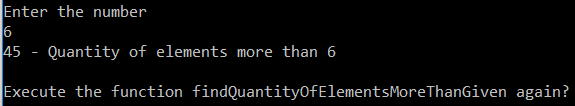


Рисунок 25 (Результат работы функции по нахождению количества элементов больше введенного с клавиатуры)

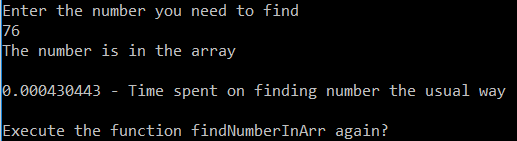


Рисунок 26 (Результат работы функции по поиску числа введенного с клавиатуры в массиве обычным способом)

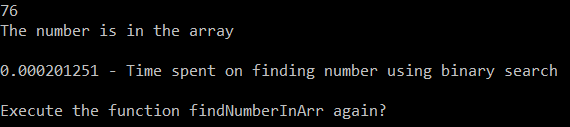


Рисунок 27 (Результат работы функции по поиску числа введенного с клавиатуры в массиве при помощи бинарного поиска)

**Выводы.**

Была написана программа, которая создает целочисленный массив размерности *N* = 100, элементы которого принимают случайное значение в диапазоне от -99 до 99. Массив был отсортирован двумя сортировками пузырьковой и shaker. Было определено время, затраченное на каждую сортировку. Были найдены максимальный и минимальный элементы массива и подсчитано время их поиска в отсортированном и неотсортированном массивах. Далее было выведено среднее значение максимального и минимального элементов, а также подсчитано количество элементов, равных ему. Было выведено количество элементов в отсортированном массиве, которые больше и меньше числа, введенного с клавиатуры. Также был произведен поиск заданного с клавиатуры числа обычным способом и с помощью бинарного поиска. Была возможность вывода каждой части программы многократно.