

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра информационных систем

ОТЧЕТ
по практической работе №2
по дисциплине «Программирование»
Тема: Одномерные статические массивы

Студент гр. 1323

Князев И. А.

Преподаватель

Глущенко А. Г.

Санкт-Петербург

2022

Цель работы.

Изучение понятия массива и их внутреннего представления в памяти, получение практических навыков работы с массивами. Знакомство и методами сортировки массива.

Основные теоретические положения.

При использовании простых переменных каждой области памяти для хранения данных соответствует свое имя. Если с группой величин одинакового типа требуется выполнить однообразные действия, им дают одно имя, а различают по порядковому номеру (индексу). Это дает возможность компактно записать множество операций с использованием циклов.

Массив представляет собой индексированную последовательность однотипных элементов с заранее определенным количеством элементов. Наглядно одномерный массив можно представить, как набор пронумерованных ячеек, в каждой из которых содержится определенное значение.

Элементы массива нумеруются с нуля. При описании массива используются те же модификаторы (класс памяти, `const` и инициализатор), что и для простых переменных.

Аналогом одномерного массива из математики может служить последовательность некоторых элементов с одним индексом: a_i при $i = 0, 1, 2, \dots, n$ — одномерный вектор. Каждый элемент такой последовательности представляет собой некоторое значение определенного типа данных. Наглядно одномерный массив можно представить как набор пронумерованных ячеек, в каждой из которых содержится определенное значение.

Постановка задачи.

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.

Выполнение работы.

Исходный код программы можно найти в репозитории:

<https://github.com/puppymail/lab>

Результаты работы программы.

```
PS C:\Users\acc13\Documents\cpp\lab\02> .\lab02.exe

1. Create array with size 100
Continue? (Y/N): y
Press any key to continue . . .

2. Sort array
1 = Bubble, 2 = Shaker, 3 = Comb, 4 = Insert: 4
Sorting took 33100 nanoseconds
Continue? (Y/N): n
1 = Bubble, 2 = Shaker, 3 = Comb, 4 = Insert: 1
Sorting took 60300 nanoseconds
Continue? (Y/N): y
Press any key to continue . . .

3. Find min and max
Min = -99, Max = 98
Finding min and max in sorted array took 3200 nanoseconds
Finding min and max in unsorted array took 900 nanoseconds
Continue? (Y/N): y
Press any key to continue . . .

4. Find median of min and max and find all elements equal to it
Median = 0
Indexes: 51
Count = 1
Continue? (Y/N): y
Press any key to continue . . .

5. Find elements less than
Enter a value between -99 and 99: 42
There are 75 elements less than 42
Continue? (Y/N): y
Press any key to continue . . .

6. Find elements more than
Enter a value between -99 and 99: 42
There are 25 elements more than 42
Continue? (Y/N): y
Press any key to continue . . .

7. Find a number
Enter a value between -99 and 99: 0
Position of '0' is 51
Normal search took 300 nanoseconds
Binary search took 400 nanoseconds
```

Рисунок 1.1. Результат программы (ч.1).

```
8. Swap elements in an array
Enter first position: 0
Enter second position: 99
Before:
0      -99
1      -95
2      -91
3      -90
4      -87
```

Рисунок 1.2. Результат программы (ч.2).

```
Swapped -99 and 98
After:
0      98
1      -95
2      -91
3      -90
4      -87
```

Рисунок 1.3. Результат программы (ч.3).

Вывод.

Во время выполнения данной работы были изучены массивы и их внутреннее представление в памяти. Также был получен навык работы с массивами и получено представление о работе алгоритмов сортировки. Полученная программа способна выводить на экран массив, сортировать его и производить другие операции над массивом, такие как: поиск, бинарный поиск, нахождение медианы и т.п.