Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Тульский государственный университет»

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

**СОРТИРОВКА ПРЯМЫМ ОБМЕНОМ**

отчет о лабораторной работе №3

по дисциплине

*ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ*

***ВАРИАНТ 6***

|  |  |  |
| --- | --- | --- |
| Выполнила: | ст. гр. 230711 | Павлова В.С. |
| Проверил: | асс. каф. ИБ | Курбаков М.Ю. |

Тула, 2022 г.

**ЦЕЛЬ И ЗАДАЧА РАБОТЫ**

**Цель:** изучить метод сортировки массивов прямым обменом. Написать программу, демонстрирующую изученные принципы.

**Задача:** в данной работе требуется написать программу, демонстрирующую использование изученных принципов.

**ЗАДАНИЕ НА РАБОТУ**

**Задание из варианта 6:** задается массив, упорядоченный по убыванию. В него добавляются пять произвольных элементов. Используя метод шейкерной сортировки, упорядочить массив по возрастанию.

**СХЕМА ПРОГРАММЫ**

Схема алгоритма программы, предназначенной для сортировки массива по возрастанию методом шейкерной сортировки, представлена на рисунке 1.



Рисунок 1 – схема алгоритма программы для упорядочивания массива по возрастанию методом шейкерной сортировки

Схема подпрограммы, непосредственно реализующей метод шейкерной сортировки, представлена на рисунке 2.



Рисунок 2 – схема алгоритма подпрограммы шейкерной сортировки

**ТЕКСТ ПРОГРАММЫ**

Текст программы на языке программирования С++ для сортировки по возрастанию методом шейкерной сортировки массива, отсортированного по убыванию, в который добавлено пять произвольных элементов, представлен в листинге 1.

## **Листинг 1. Текст программы**

#include <iostream>

#include <fstream>

int\* ShakerSort(int\* a, int size)

{

int left = 0;

int right = size - 1;

while (left <= right)

{

for (int i = right; i >= left; i--)

if (a[i - 1] > a[i]) std::swap(a[i], a[i - 1]);

left++;

for (int i = left; i <= right; i++)

if (a[i - 1] > a[i]) std::swap(a[i], a[i - 1]);

right--;

}

return a;

}   
  
int main()

{

setlocale(LC\_ALL, "RUSSIAN");

std::ifstream input("input.txt");

int size;

std::cout << "Введите размер массива: ";

std::cin >> size;

int\* arrayToSort = new int[size];

std::cout << "\nМассив до сортировки:\n";

for (int i = 0; i < size; i++)

{

input >> arrayToSort[i];

std::cout << arrayToSort[i] << "\t";

if (i > 0 && (i+1) % 10 == 0) std::cout << "\n";

}

input.close();

ShakerSort(arrayToSort, size);

std::cout << "\nМассив после сортировки:\n";

for (int i = 0; i < size; i++)

{

std::cout << arrayToSort[i] << "\t";

if (i > 0 && (i+1) % 10 == 0) std::cout << "\n";

}

return 0;

}

**ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ**

Данная программа предназначена для сортировки массива по возрастанию методом шейкерной сортировки. При запуске программы пользователю предлагается ввести длину последовательности. Входную последовательность программа считывает из файла и выводит её в консоль в неизменном виде. После программа сортирует последовательность и выводит в консоль уже отсортированную последовательность.

**ИНСТРУКЦИЯ ПРОГРАММИСТА**

Данная программа предназначена для сортировки массива по возрастанию методом шейкерной сортировки. Структуры данных, используемые в программе, приведены в таблице 1.

Таблица 1 – Структуры данных в программе

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип (класс)** | **Предназначение** |
| size | int | Длина массива |
| arrayToSort | int\* | Сортируемый массив |

В программе имеются следующие функции:

1. int\* ShakerSort(int\* a, int size) - функция, реализующая сортировку методом шейкерной сортировки.

Таблица 2 – Структуры данных, используемые в подпрограмме ShakerSort

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Предназначение** |
| *формальные параметры* | | |
| a | int\* | Ссылка на массив для сортировки |
| size | int | Размер массива |
| left | int | Левый индекс |
| right | int | Правый индекс |

# **ДЕМОНСТРАЦИОННЫЙ ПРИМЕР**

Сперва отсортируем входную последовательность вручную. Согласно заданию варианта, заданный массив должен быть отсортирован по убыванию, а после в него добавляется пять произвольных элементов. Пусть входная последовательность имеет длину n = 55 и вид:

|  |
| --- |
| 958 958 898 895 894 887 879 845 819 804 **2** 783 744 714 710 706 700 666 661 644 601 **8** 565 553 552 550 528 510 463 **4** 448 438 428 421 407 385 **11** 366 365 312 305 293 275 245 207 197 **32** 193 122 101 95 93 75 52 2 |

Тогда после сортировки она должна выглядеть следующим образом:

|  |
| --- |
| 2 **2** **4 8 11** **32** 52 75 93 95 101 122 193 197 207 245 275 293 305 312 365 366 385 407 421 428 438 448 463 510 528 550 552 553 565 601 644 661 666 700 706 710 714 744 783 804 819 845 879 887 894 895 898 958 958 |

Теперь проверим результат работы программы, представленный на рисунке 3, на соответствие полученной последовательности.

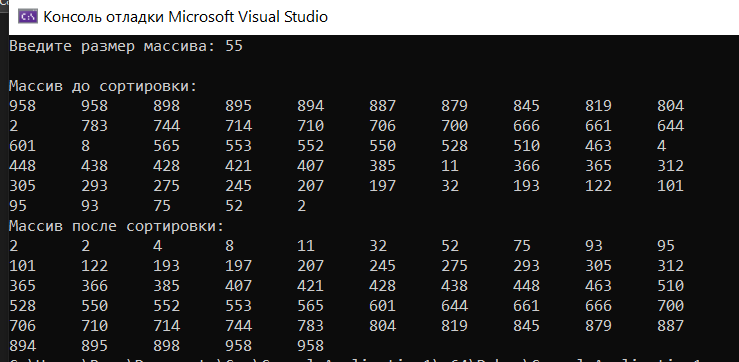


Рисунок 3 – Результат работы программы

Как видно по рисунку, последовательность отсортирована верно, и добавленные элементы тоже встали на свои места.

# **ВЫВОДЫ**

В ходе данной лабораторной работы был изучен принцип метод сортировки массивов прямым обменом, а именно – метод шейкерной сортировки. Для демонстрации полученных знаний была написана программа для сортировки массивов указанным методом, результат работы которой был проверен аналитически. По результатам проверки можно сделать вывод о том, что программа работает корректно.