Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №5**

**«ИСЛЕДОВАНИЕ АЛГОРИТМОВ СОРТИРОВКИ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Криницын Олег Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

1. **Цель домашней контрольной работы:**

Получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами.

1. **Формулировка задания:**

Вариант 11.

1. Реализовать сортировку данных с помощью “пузырькового” алгоритма.
2. Реализовать сортировку данных с помощью алгоритма слияния.
3. В обоих случаях необходимо предусмотреть возможность изменения компаратора (реализация компаратора в виде передаваемой в подпрограмму функции).
4. Считывание и вывод данных необходимо производить из текстового файла.
5. Для демонстрации работы программных реализаций самостоятельно подготовить варианты входных данных (при этом объём текстовых файлов должен позволять оценить скорость работы программ).
6. **Код программы**

**uses**

crt;

**procedure** menu;

**begin**

TextColor(Green);

writeln('Нажмите "1" сортировка "пузырьком".');

TextColor(Red);

writeln('Нажмите "2" сортировка "слиянием".');

TextColor(Yellow);

writeln('Нажмите "3" чтобы выйти из программы.');

**end**;

**var**

m, n, k, i, j: integer;

mass: **Array** [1..10] **of** integer;

mas: **array** [1..10] **of** integer;

inp, outp: text;

key: char;

**procedure** MergeSort(a, c: integer);

**var**

x, j, i, n1, n2: integer;

rez: **array**[1..10] **of** integer;

**begin**

**if** c <= a **then**

**exit**

**else**

**begin**

x := (a + c) **div** 2;

MergeSort(a, x);

MergeSort(x + 1, c);

n1 := a;

n2 := x + 1;

**for** i := a **to** c **do**

**begin**

**if** (n1 < x + 1) **and** ((n2 > c) **or** (mas[n1] < mas[n2])) **then**

**begin**

rez[i] := mas[n1];

inc(n1);

**end**

**else**

**begin**

rez[i] := mas[n2];

inc(n2);

**end**;

**end**;

**for** j := a **to** c **do**

mas[j] := rez[j];

**end**;

**end**;

**begin**

menu;

**repeat**

key := ReadKey;

**case** key **of**

'1':

**begin**

assign(inp, 'sort.in.txt'); // Тесты: sort.in1.txt, sort.in2.txt

assign(outp, 'sort.out.txt');

reset(inp);

rewrite(outp);

readln(inp, m);

**for** i := 1 **to** m **do**

read(inp, mass[i]);

writeln(mass);

**for** i := 1 **to** m - 1 **do**

**for** j := 1 **to** m - i **do**

**if** mass[j] > mass[j + 1] **then**

**begin**

k := mass[j];

mass[j] := mass[j + 1];

mass[j + 1] := k

**end**;

**for** i := 1 **to** m **do**

write(outp, mass[i]:4);

writeln(mass);

close(inp);

close(outp);

menu;

**end**;

'2':

**begin**

assign(inp, 'sort.in.txt'); // Тесты: sort.in1.txt, sort.in2.txt

assign(outp, 'sort.out.txt');

reset(inp);

rewrite(outp);

readln(inp, n);

**for** i := 1 **to** n **do**

read(inp, mas[i]);

writeln(mas);

MergeSort(1, n);

**for** i := 1 **to** n **do**

write(outp, mas[i]:4);

close(inp);

close(outp);

writeln(mas);

menu;

**end**;

**end**;

**until** key = '3'

**end**.

1. **Результат выполнения программы**

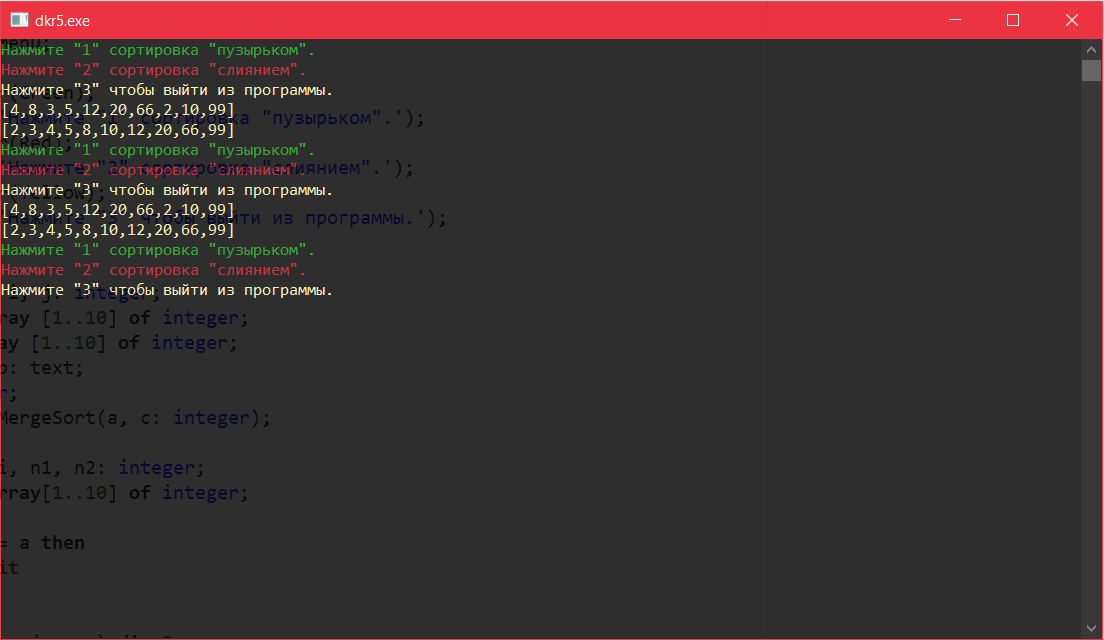
****

Рисунок 1 - Результат выполнения программы

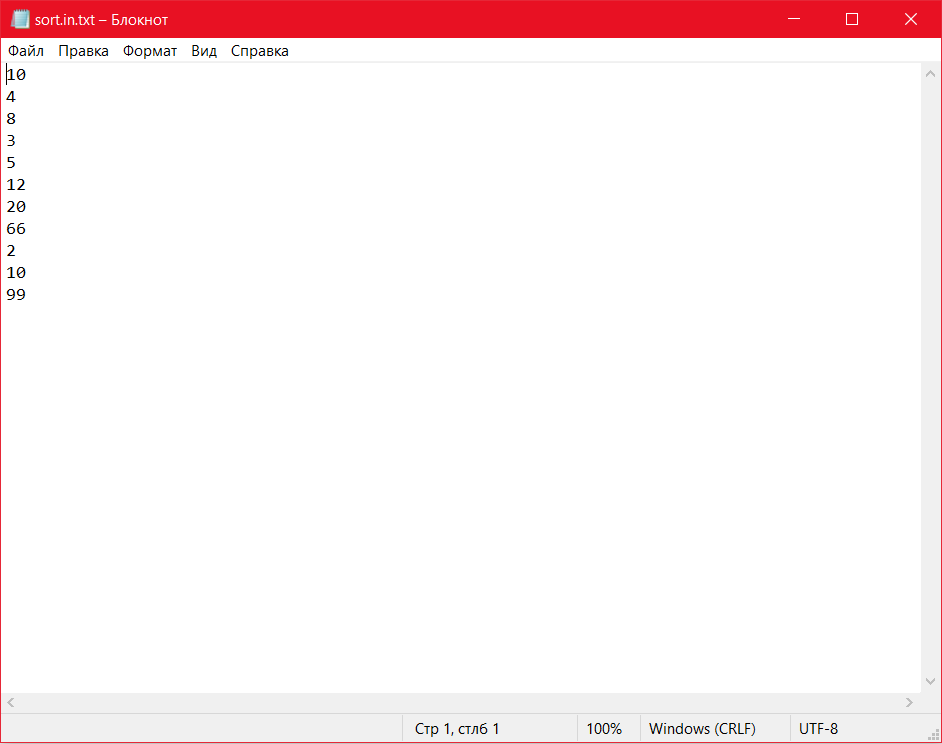


Рисунок 2 - Результат выполнения программы

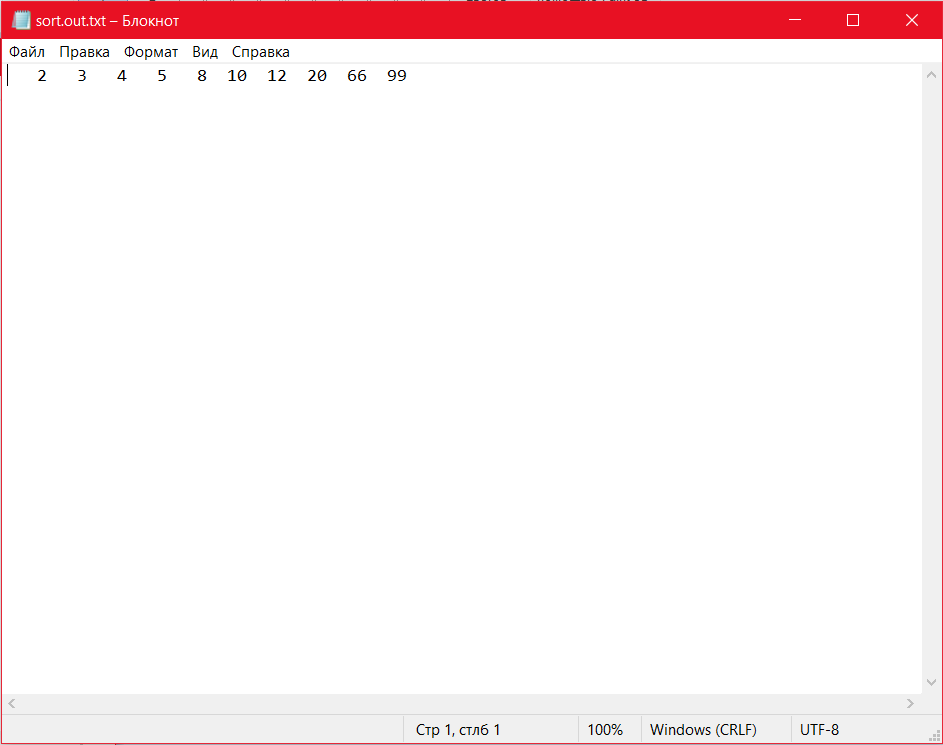


Рисунок 3 - Результат выполнения программы

1. **Вывод**

Нами была поставлена цель работы: Получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами. Работа была разделена на несколько этапов: цель, формулировка задания, написание кода программы, результат выполнения программы, вывод. В коде программы используются модуль crt и процедура MergeSort(a, c: integer). CRT - это модуль, который отвечает за вывод информации и ее внешний вид. В модуле. CRT содержатся команды, которые позволяют изменять цвет шрифта и фона, задерживать выполнение следующей команды, включать и выключать звук, размещать текст в нужном пользователю месте экрана. MergeSort – процедура используется для сортировки методом слияния. В коде программы, в первую очередь, мы вводим процедуру для сортировки слиянием, далее создаем case menu. В case menu, мы используем процедуру и массив для сортировки «пузырьком». Код программы был написан на PascalABC. Итак, делая вывод, хочу сказать, что в ходе работы все цели были достигнуты.