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**

**«Исследование алгоритмов сортировки»**

**ПО ДИСЦИПЛИНЕ «Основы алгоритмизации и программирования»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Островский Тимур Евгеньевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

В отчете должны отображаться:

1. Цель лабораторной работы
2. Формулировка задания (с вариантом)
3. Код программы
4. Результат выполнения программы
5. Вывод

1. Цель домашней контрольной работы:

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

2. Формулировка задания (с вариантом):

Вариант: 17

Задание:

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 объем тестовых файлов должен позволять оценить скорость работы программ).

3. Код программы

**program** DKR5;

**uses** crt, system;

**var**

input: **file of** char;

output: text;

arr: **array** [1..10000] **of** char;

start, start1, stop, stop1, n: integer;

s: string;

**function** vstavsort: integer;

**var**

i, j: integer;

a: char;

**begin**

**for** i := 2 **to** n **do**

**begin**

a := arr[i];

j := i - 1;

**while** (j >= 1) **and** (arr[j] > a) **do**

**begin**

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

j := j - 1;

**end**;

arr[j + 1] := a;

**end**;

**end**;

**function** vstavsortub: integer;

**var**

i, j: integer;

a: char;

**begin**

**for** i := 2 **to** n **do**

**begin**

a := arr[i];

j := i - 1;

**while** (j >= 1) **and** (arr[j] < a) **do**

**begin**

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

j := j - 1;

**end**;

arr[j + 1] := a;

**end**;

**end**;

**procedure** fastsort(l, r: integer);

**var**

i, j, m: integer;

arr1, y1: char;

**begin**

i := l;

j := r;

m := round((l + r) / 2);

arr1 := arr[m];

**repeat**

**while** arr[i] < arr1 **do** inc(i);

**while** arr[j] > arr1 **do** dec(j);

**if** i <= j **then**

**begin**

y1 := arr[i];

arr[i] := arr[j];

arr[j] := y1;

inc(i);

dec(j);

**end**;

**until** i > j;

**if** l < j **then** fastsort(l, j);

**if** i < r **then** fastsort(i, r);

**end**;

**procedure** fastsortub(first, last: integer);

**var**

f, l: integer; mid, count: char;

**begin**

f := first;

l := last;

mid := arr[(f + l) **div** 2];

**repeat**

**while** arr[f] > mid **do** inc(f);

**while** arr[l] < mid **do** dec(l);

**if** f <= l **then**

**begin**

count := arr[f];

arr[f] := arr[l];

arr[l] := count;

inc(f);

dec(l);

**end**;

**until** f > l;

**if** first < l **then** fastsortub(first, l);

**if** f < last **then** fastsortub(f, last);

**end**;

**begin**

assign(input, 'e:input.txt'); assign(output, 'e:output.txt');

rewrite(input); rewrite(output);

writeln('Введите длинну массива');

readln(n);

writeln('Введите массив');

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

**begin**

readln(arr[i]);

write(input, arr[i]);

**end**;

**var** st: integer;

st := 1;

**while** st <> 0 **do**

**begin**

clrscr;

writeln('Выберите сортировку');

writeln('Сортировка вставкой по возрастанию - 1');

writeln('Сортировка быстрым алгоритимом по возрастанию - 2');

writeln('Сортировка вставкой по убыванию - 3');

writeln('Сортировка быстрым алгоритимом по убыванию - 4');

writeln('Выход из программы - 0');

readln(st);

start := DateTime.Now.Second;

start1 := DateTime.Now.MilliSecond;

**case** st **of**

1: vstavsort;

2: fastsort(1, n);

3: vstavsortub;

4: fastsortub(1, n);

0: **exit**;

**end**;

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

print(output, arr[i]);

stop := DateTime.Now.Second;

stop1 := DateTime.Now.MilliSecond;

close(output);

reset(output);

read(output, s);

close(output);

rewrite(output);

write(s);

writeln;

writeln('Время выполнения: ', stop - start, ' секнуд, ', abs(stop1 - start1), ' милисекунд');

writeln('Для продолжения дважды нажмите enter');

readln;

readln;

**end**;

**end**.

4. Результат выполнения программы

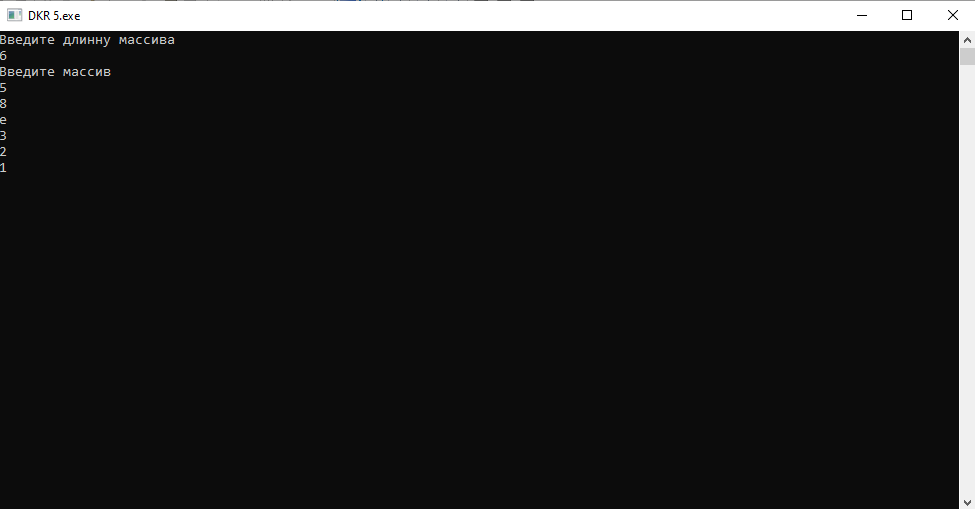


Рисунок 1 – результат вывода программы

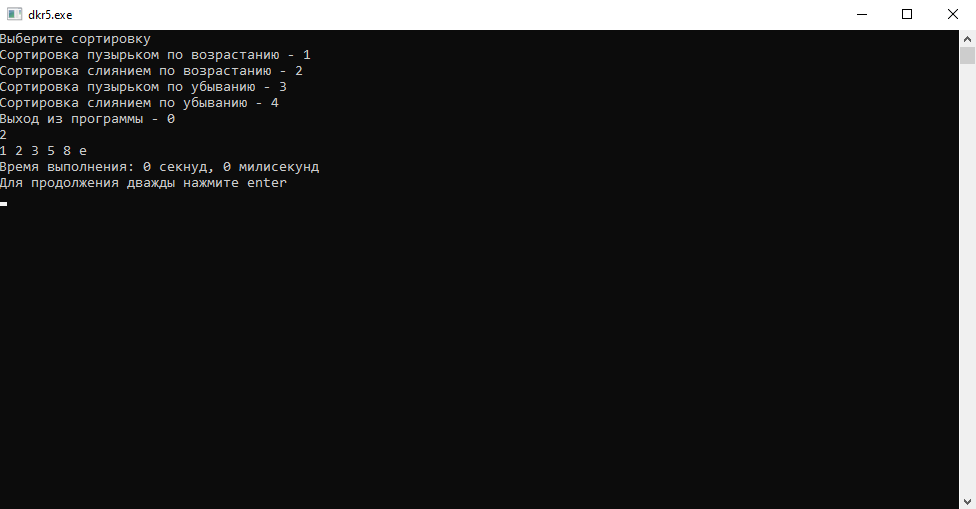


Рисунок 2 – результат вывода программы

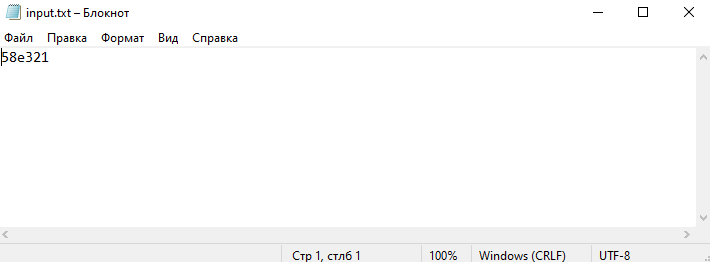


Рисунок 1 – результат вывода программы

6. Вывод

В результате проделанной работы были освоены алгоритмы сортировки, а именно пузырьковый алгоритм и алгоритм слияния. Также была изучена библиотека System, позволяющая вычислить длительность выполнения программы. Написанная программа запрашивает у пользователя массив, который нужно отсортировать, сортирует и выводит его на экран. В программе присутствует case-меню для взаимодействия с пользователем.