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-204-52-00

Краснянская Евгения Сергеевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

1. Цель контрольной работы: получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами.

2. Задание:

Вариант: 1.

1. Реализовать сортировку данных с помощью вставок.
2. Реализовать сортировку данных с помощью поразрядного алгоритма.
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 реализация самостоятельно подготовить варианты входных данных (при этом объем тестовых файлов должен позволять оценить скорость работы программ).

3. Код программы

**uses** Crt, System;

**var**

mas, mas1: **array** [1..10000] **of** integer;

n, t, j, i: integer;

f: **file of** integer;

fin: text;

star, stop, star1, stop1: integer;

**var**

l: byte;

**function** vibor(**var** v: byte): byte;

**begin**

writeln('Сотрировку вставок - 1');

writeln('Сотрировку поразрядам - 0');

Write('Выберите вариант: ');

readln(v);

vibor := v;

**end**;

**var**

l1: byte;

**function** viborl(**var** v1: byte): byte;

**begin**

writeln('Сотрировать по возростанию - 1');

Write('Сотрировать по убыванию - 0 ');

Write('Выберите вариант: ');

readln(v1);

viborl := v1;

**end**;

**function** sortV: integer

**var**

i, j: integer;t: integer;

**begin**

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

**begin**

t := mas[i];

j := i - 1;

**while** (j >= 1) **and** (mas[j] > t) **do**

**begin**

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

j := j - 1;

**end**;

mas[j + 1] := t;

**end**;

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

**begin**

reWrite(fin);

write(fin, mas[i]);

write(fin, ' ');

write(mas[i], ' ');

**end**;

close(fin);

**end**;

**function** sortVM: integer;

**var**

i, j: integer;t: integer;

**begin**

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

**begin**

t := mas[i];

j := i - 1;

**while** (j >= 1) **and** (mas[j] < t) **do**

**begin**

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

j := j - 1;

**end**;

mas[j + 1] := t;

**end**;

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

**begin**

reWrite(fin);

write(fin, mas[i]);

write(fin, ' ');

write(mas[i], ' ');

**end**;

close(fin);

**end**;

**function** SortP: integer;

**var**

C: **array**[0..1000] **of** integer;

i, j: integer;{t:char;}

**begin**

**for** j := 0 **to** n **do**

C[j] := 0;

**for** j := 1 **to** N **do**

C[(mas[j] **mod** (t \* 10)) **div** t] := C[(mas[j] **mod** (t \* 10)) **div** t] + 1;

**for** j := 1 **to** 9 **do**

C[j] := C[j - 1] + C[j];

**for** j := N **downto** 1 **do**

**begin**

mas1[C[(mas[j] **mod** (t \* 10)) **div** t]] := mas[j];

C[(mas[j] **mod** (t \* 10)) **div** t] := C[(mas[j] **mod** (t \* 10)) **div** t] - 1;

**end**;

**end**;

**begin**

Assign(f, 'F:\ДКР 5\Тест.txt');

Assign(fin, 'F:\ДКР 5\Тест1.txt');

reWrite(f);

reWrite(fin);

writeln('Введите количество переменных');

readln(n);

writeln('Введите переменные');

t := 1;

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

**begin**

readln(mas[i]);

write(f, mas[i]);

**end**;

writeln('Какую сортировку испльзовать?');

vibor(l);

**case** l **of**

1:

**begin**

viborl(l1);

**case** l1 **of**

1:

**begin**

star1 := DateTime.Now.Second;

star := DateTime.Now.MilliSecond;

sortV();

stop1 := DateTime.Now.Second;

stop := DateTime.Now.MilliSecond;

writeln('Программа выполнялась ', stop1 - star1, ' секунд ', abs(stop - star), ' милисекунду');

readln;

**end**;

0:

**begin**

star1 := DateTime.Now.Second;

star := DateTime.Now.MilliSecond;

sortVM();

stop1 := DateTime.Now.Second;

stop := DateTime.Now.MilliSecond;

writeln('Программа выполнялась ', stop1 - star1, ' секунд ', abs(stop - star), ' милисекунду');

readln;

**end**;

**end**;

**end**;

0:

**begin**

viborl(l1);

**case** l1 **of**

1:

**begin**

star1 := DateTime.Now.Second;

star := DateTime.Now.MilliSecond;

t := 1;

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

**begin**

SortP;

mas := mas1;

t := t \* 10;

rewrite(fin);

write(fin, mas[i]);

write(fin, ' ');

write(mas[i], ' ');

**end**;

close(fin);

stop1 := DateTime.Now.Second;

stop := DateTime.Now.MilliSecond;

writeln('Программа выполнялась ', stop1 - star1, ' секунд ', abs(stop - star), ' милисекунду');

readln;

**end**;

0:

**begin**

t := 1;

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

**begin**

SortP;

mas1 := mas;

t := t \* 10;

rewrite(fin);

write(fin, mas[i]);

write(fin, ' ');

write(mas[i], ' ');

**end**;

close(fin);

stop1 := DateTime.Now.Second;

stop := DateTime.Now.MilliSecond;

writeln('Программа выполнялась ', stop1 - star1, ' секунд ', abs(stop - star), ' милисекунду');

readln;

**end**;

**end**;

**end**;

**end**;

**end**.

4.Результат выполнения программы

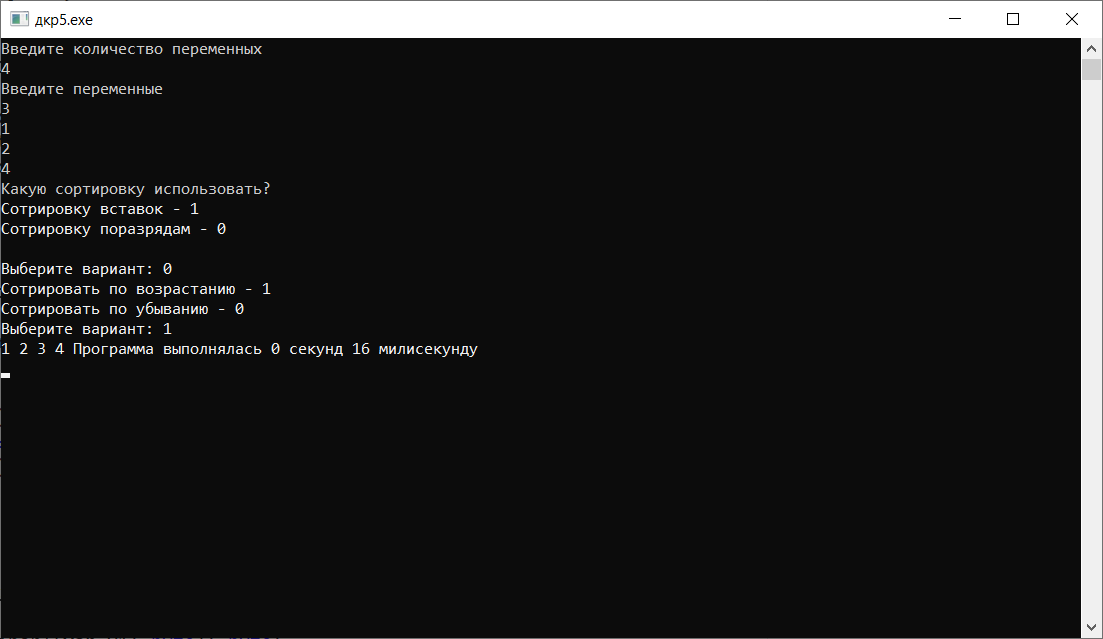


Рисунок 1 – Результат выполнения программы (1/2)

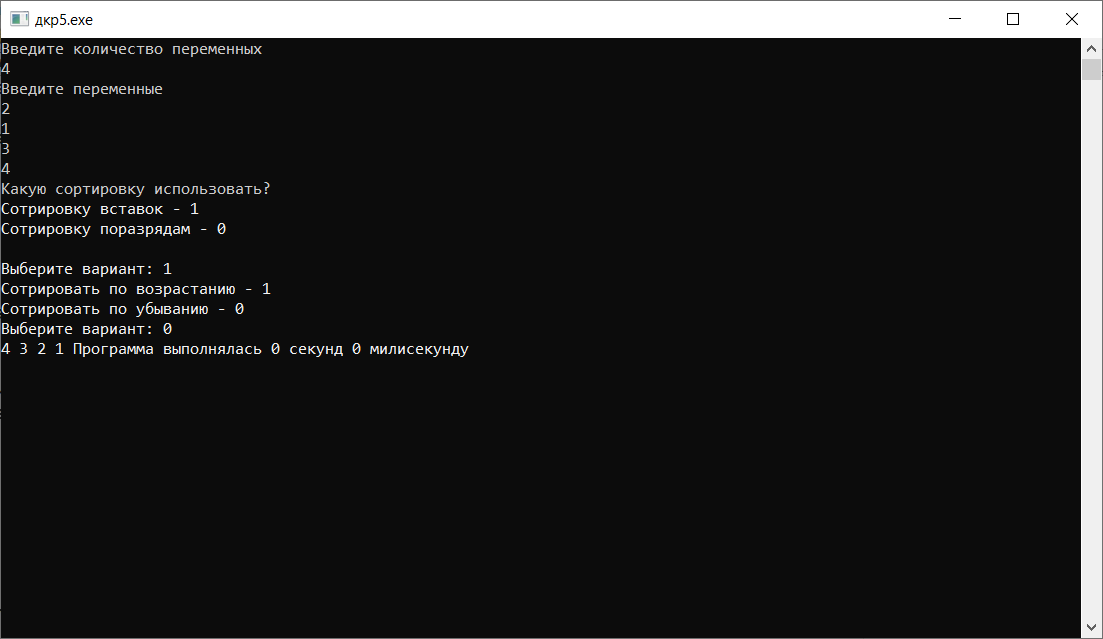


Рисунок 2 – Результат выполнения программы (2/2)

5.Вывод

Для выполнения задания были использованы функции библиотек CRT и System языка Pascal.

Для сортировки данных были изучены и использованы методы сортировки вставки и поразрядного алгоритма.

В программе реализована возможность изменения компаратора для выбора метода сортировки (по возрастанию либо по убыванию).

Для вычисления времени выполнения программы были использованы функции DateTime.Now.MilliSecond библиотеки System.

Ко всему можно добавить использование операторов if, циклов for, внутри блочное описание переменных, инициализации переменных при описании, расширенное присваивание.