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»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Конышев Иван Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

Цель работы

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

Формулировка задания

1. Реализовать сортировку данных с помощью алгоритма подсчётом.

2. Реализовать сортировку данных с помощью быстрого алгоритма.

3. В обоих случаях необходимо предусмотреть возможность изменения компаратора (реализация компаратора в виду 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 тестовых файлов должен позволять оценить скорость работы программ).

Описание алгоритма

Данный алгоритм позволяет выбрать 3 уже заданных массива из файлов или задать свой массив самостоятельно. Далее выбирается способ сортировки массива и производится сама сортировка.

Схема алгоритма с комментариями

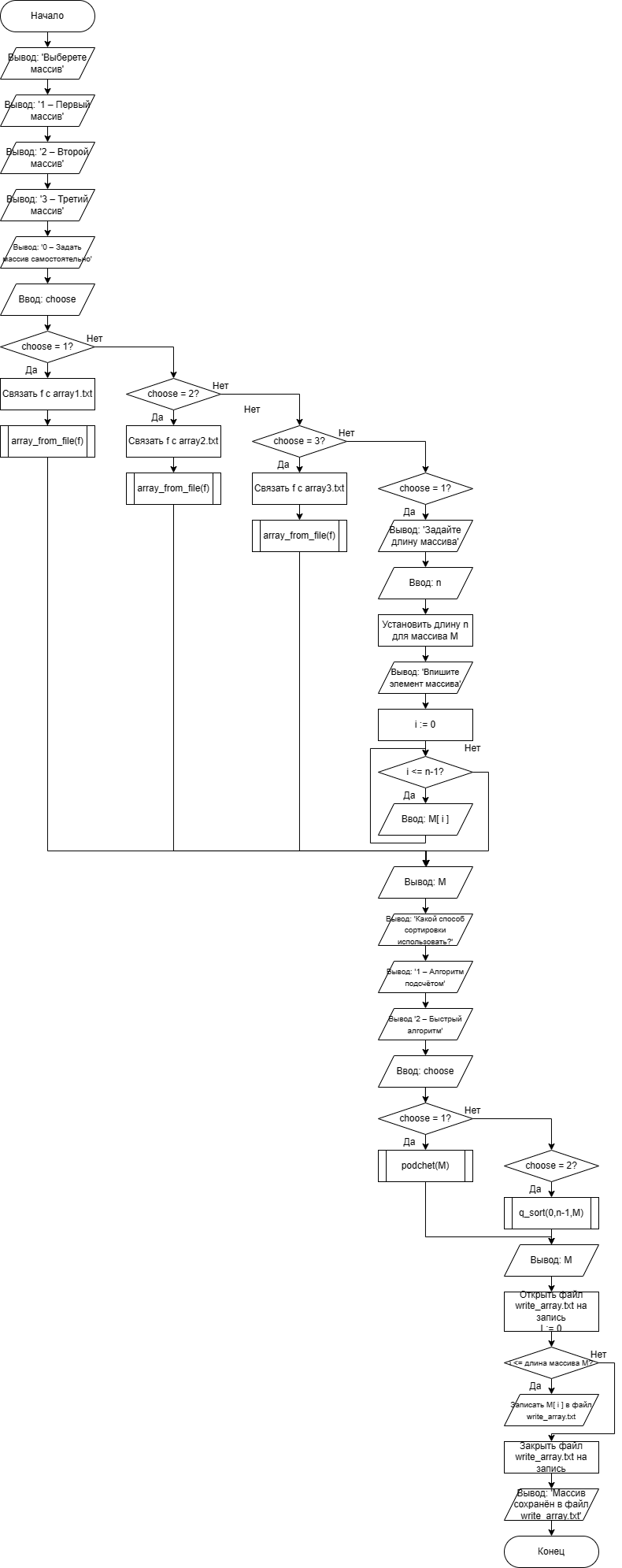


Рисунок 1 – Схема главного алгоритма



Рисунок 2 – Схема алгоритма процедуры array\_from\_file

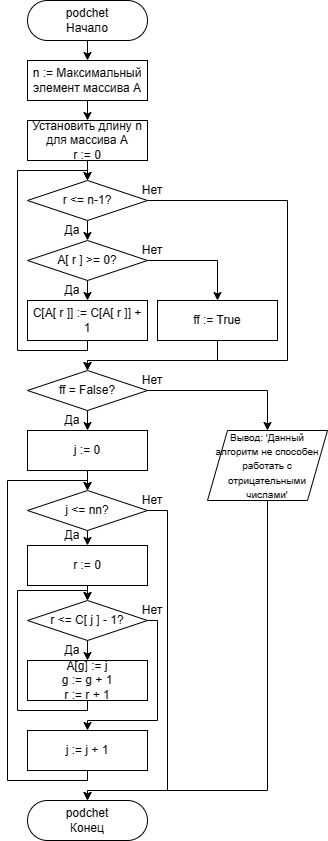


Рисунок 3 – Схема алгоритма процедуры podchet

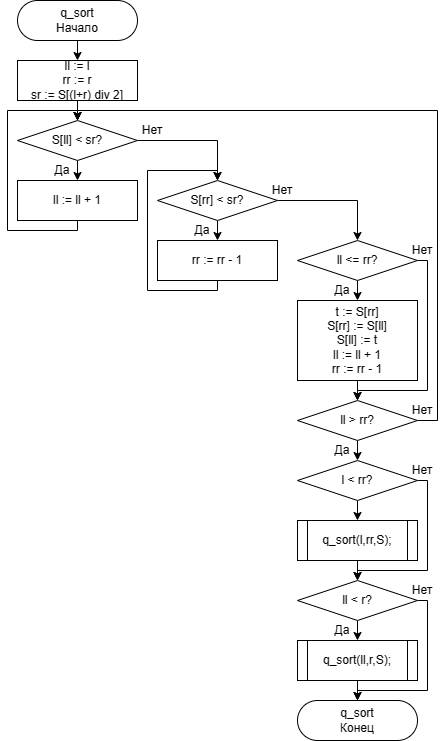


Рисунок 4 – Схема алгоритма процедуры q\_sort

Код программы

**var** choose,n,i,x : integer;

M : **array of** integer;

//C : array of integer;

f : text;

**procedure** array\_from\_file(ff : text);

**begin**

reset(ff);

**while not** Eof(ff) **do begin**

n := n + 1;

readln(ff,x);

Setlength(M, n);

M[n-1] := x;

**end**;

//n := length(M);

CloseFile(ff);

**end**;

**procedure** podchet(A : **array of** integer);

**var** C : **array of** integer;

g,nn,j,r : integer;

ff : boolean;

**begin**

//nn := A[0];

//for r := 0 to n-1 do if nn < A[r] then nn := A[r];

nn := A.Max;

//print(nn);

setlength(C,nn+1);

**for** r := 0 **to** n-1 **do if** A[r] >= 0 **then** C[A[r]] := C[A[r]] + 1 **else begin** ff := True; **break**; **end**;

//writeln(C);

**if** ff = False **then**

**for** j := 0 **to** nn **do begin**

**for** r := 0 **to** C[j] - 1 **do begin**

A[g] := j;

g := g + 1;

**end**;

**end else** writeln('Данный алгоритм не способен работать с отрицательными числами');

//write(A);

**end**;

**procedure** q\_sort(l, r : integer; S : **array of** integer);

**var** ll,rr,t,sr : integer;

**begin**

ll := l;

rr := r;

sr := S[(l+r) **div** 2];

**repeat**

**while** S[ll] < sr **do** ll := ll + 1;

**while** S[rr] > sr **do** rr := rr - 1;

**if** ll <= rr **then begin**

t := S[rr];

S[rr] := S[ll];

S[ll] := t;

ll := ll + 1;

rr := rr - 1;

**end**;

**until** ll > rr;

**if** l < rr **then** q\_sort(l,rr,S);

**if** ll < r **then** q\_sort(ll,r,S);

**end**;

**begin**

//writeln('Введите длину массива');

//readln(n);

//setlength(M,n);

Writeln('Какой массив использовать?');

Writeln('1 – Первый массив');

Writeln('2 – Второй массив');

Writeln('3 – Третий массив');

Writeln('0 – Задать массив самостоятельно');

readln(choose);

**case** choose **of**

1 : **begin** Assign(f, 'array1.txt'); array\_from\_file(f); **end**;

2 : **begin** Assign(f, 'array2.txt'); array\_from\_file(f); **end**;

3 : **begin** Assign(f, 'array3.txt'); array\_from\_file(f); **end**;

0 : **begin** writeln('Задайте длину массива'); readln(n); setlength(M,n); writeln('Впишите элементы массива'); **for** i := 0 **to** n-1 **do** readln(M[i]); **end**;

**end**;

writeln(M);

writeln('Какой способ сортировки использовать?');

writeln('1 – Алгоритм подсчётом');

writeln('2 – Быстрый алгоритм');

readln(choose);

**case** choose **of**

1 : podchet(M);

2 : q\_sort(0,n-1,M);

//0 : exit

**end**;

write(M);

**end**.

Результат выполнения программы

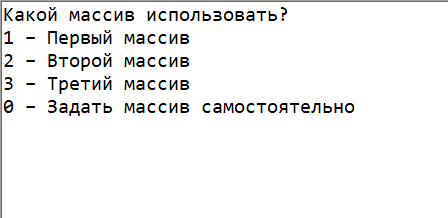


Рисунок 5 - Результат выполнения программы

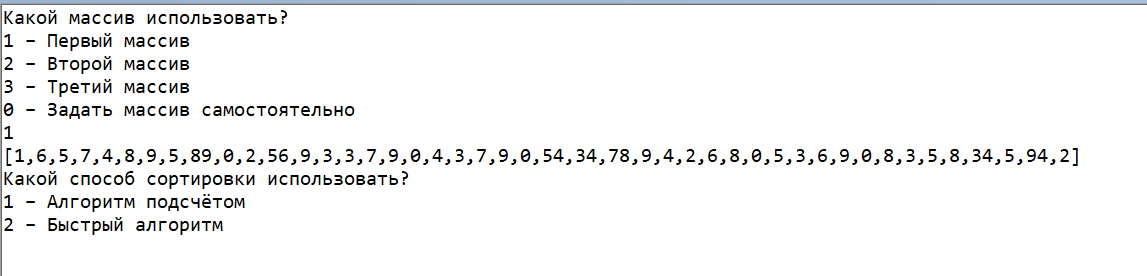


Рисунок 6 – Результат выполнения процедуры sumint

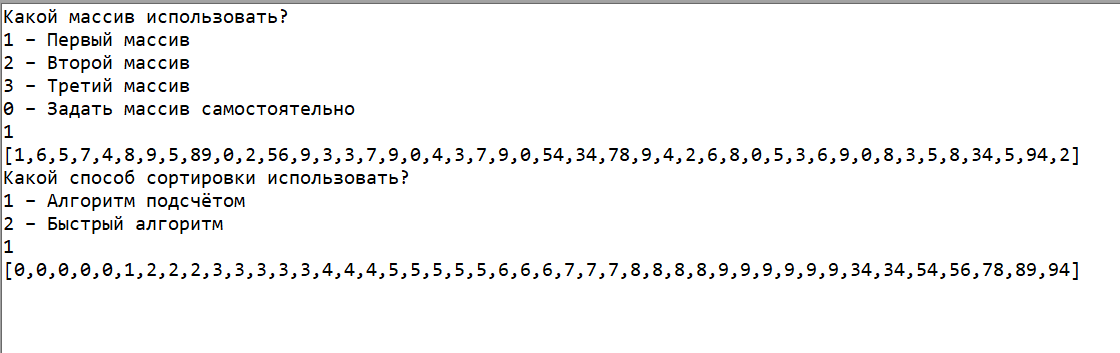


Рисунок 11 – Результат выполнения процедуры podchet

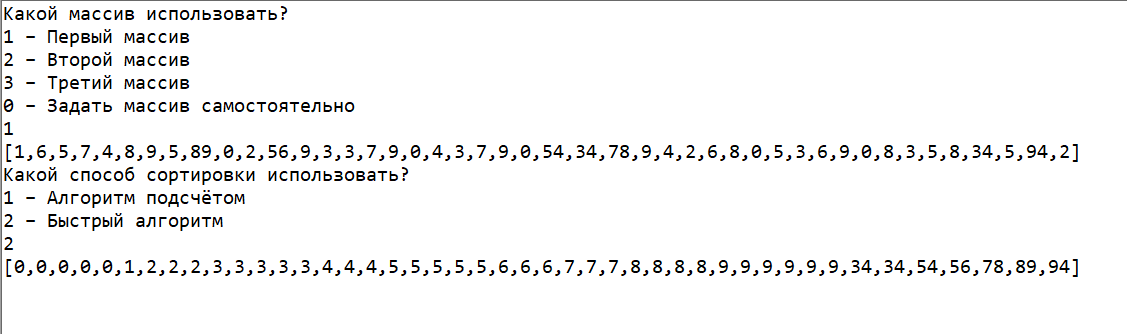


Рисунок 7 – Результат выполнения процедуры q\_sort

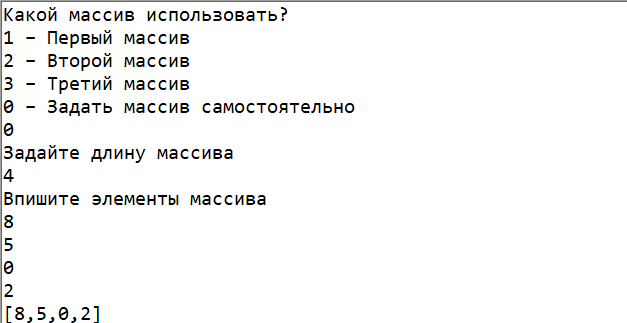


Рисунок 13 – Результат выполнения choose = 0

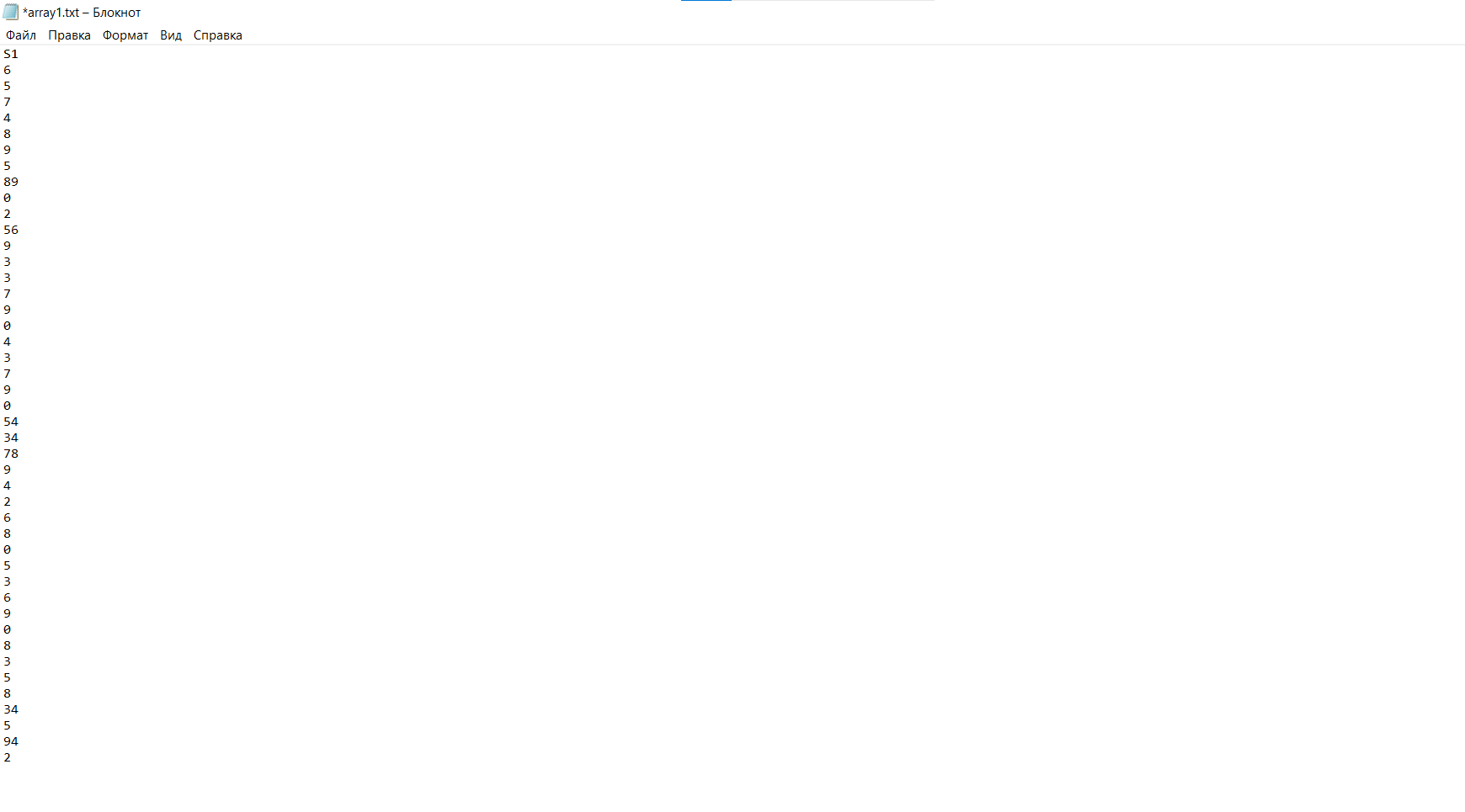


Рисунок 14 – Файл с изначальным массивом

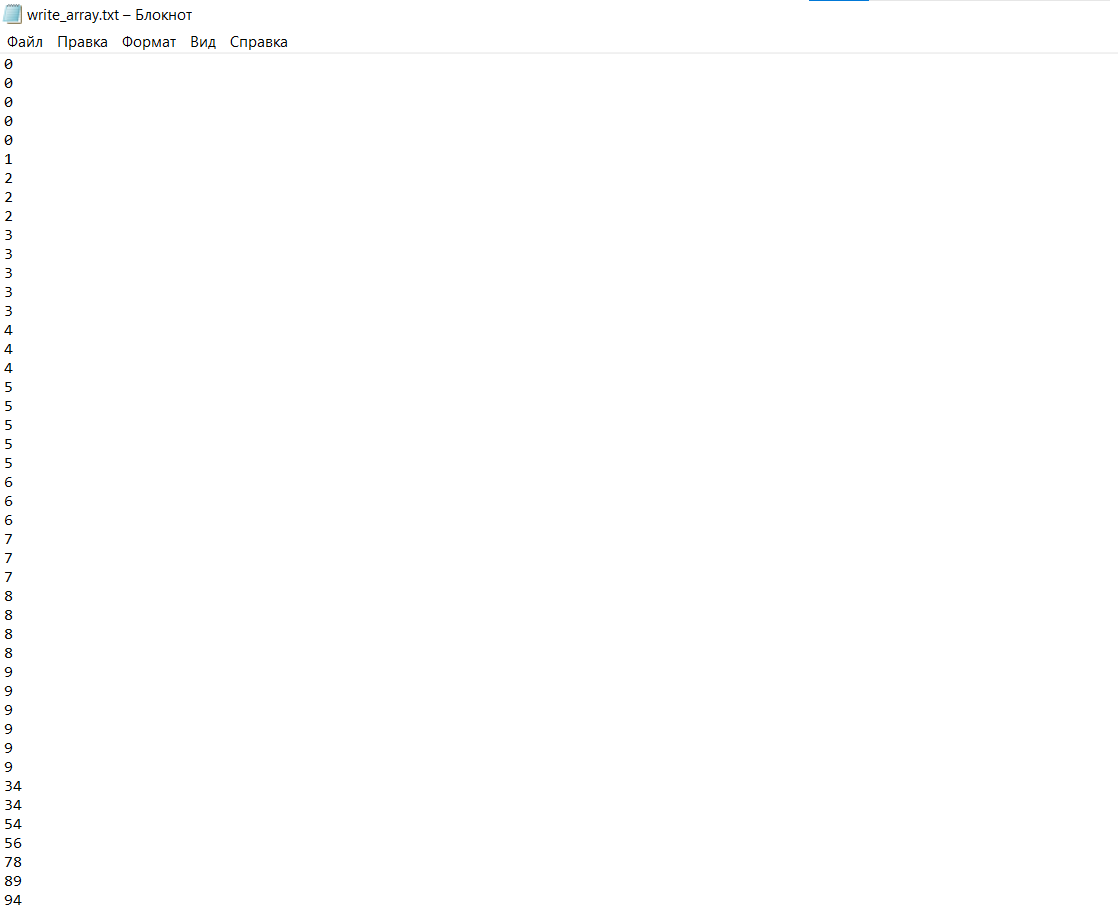


Рисунок 15 – Отсортированный массив, записанный в файл write\_array.txt

Вывод

В ходе выполнения данной работы мы смогли применить изученный материал лабораторных занятий на практике, процедуры, условный оператор, цикл while, repeat, for, создание переменных и операторов ввода и вывода, а также мы построили схему алгоритма с помощью бесплатного онлайн-сервиса app.diagrams.net.

В результате выполнения данной работы получили базовые сведения о наиболее известных алгоритмах сортировки, изучили принципы работы с текстовыми файлами, а также выполнили данное нам практическое задание, использовав изученный материал на практике.

Таким образом, выполнение домашней контрольной работы №5 принесло нам ценный опыт и знания, которые будут полезны в нашем дальнейшем образовании. Мы научились применять определённые базовые конструкции языка программирования, а также разрабатывать схему алгоритма для решения задачи. Этот опыт будет полезен нам будущем для решений новых задач и работой над реальными проектами.