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 ПРОГРАММИРИВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Цветкова Кристина Андреевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

Содержание

1. Цель лабораторной работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод

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

**2.Формулировка задания (Вариант:22)**

**Вариант 22**

1. Реализовать сортировку данных с помощью алгоритма выбором.
2. Реализовать сортировку данных с помощью быстрого алгоритма.
3. В обоих случаях необходимо предусматривать 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 скорость работы программ).

**3. Описание алгоритма**

**Задание № 1**

В процедуре sortselect происходит сортировка данных. В цикле 1 происходит присвоение максимальному индексу, значение i. Дальше запускается второй цикл, где происходит условие, при котором индекс массива больше максимального индекса, то к нему присваивается значение j. Иначе проверяется условье, где максимальный индекс неравен i. После завершается второй и первый циклы.

В части кода, где происходит открытие файла, изначально открывается файл с входными данными, далее следует цикл, в котором происходит перебор данных из файла, после данный файл закрывается, и происходит сортировка. Создается второй файл, который получает выходные данные. Далее завершается первый цикл и начинается второй цикл, который служит для записывания элементов во втором файле, после для сохранения всех данных второй файл закрывается и заканчивается второй цикл.

**Задание № 2**

В процедуре quickSort происходит сортировка данных. В части кода, где происходит открытие файла input.txt, изначально открывается файл с входными данными, далее следует цикл, в котором происходит перебор данных из файла, после данный файл закрывается, и происходит сортировка. Создается второй файл output.txt, который получает выходные данные. Далее завершается первый цикл и начинается второй цикл, который служит для записывания элементов во втором файле, после для сохранения всех данных второй файл закрывается и заканчивается второй цикл.

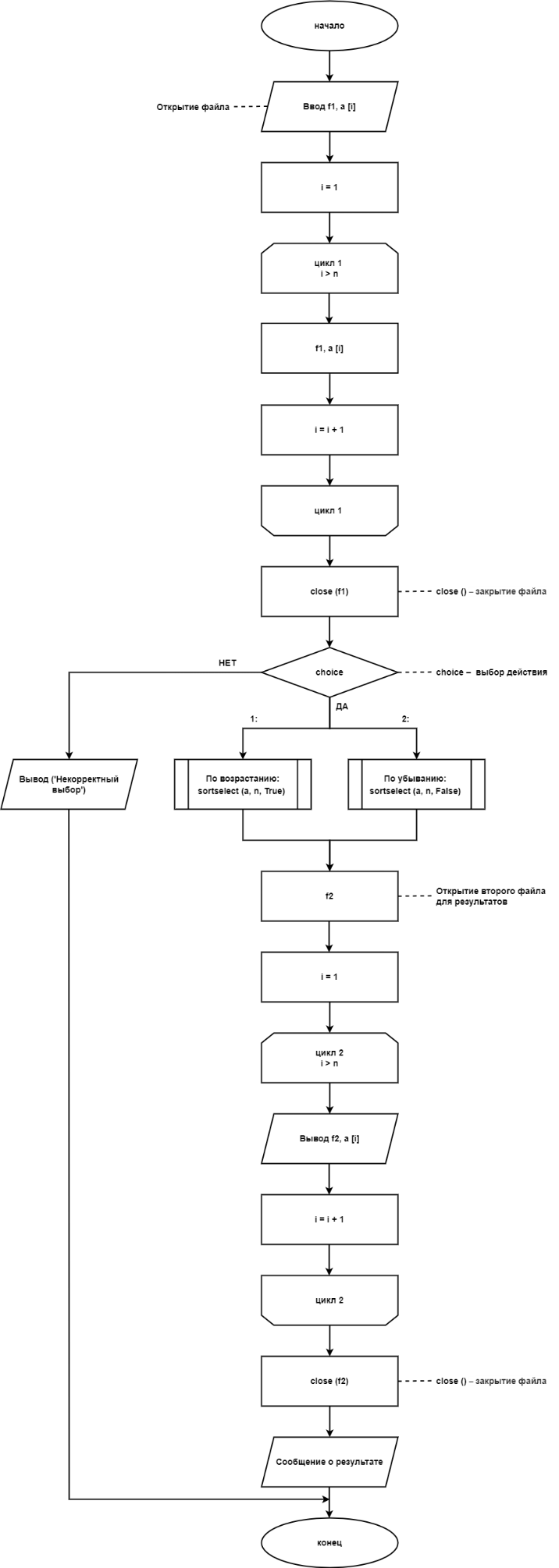
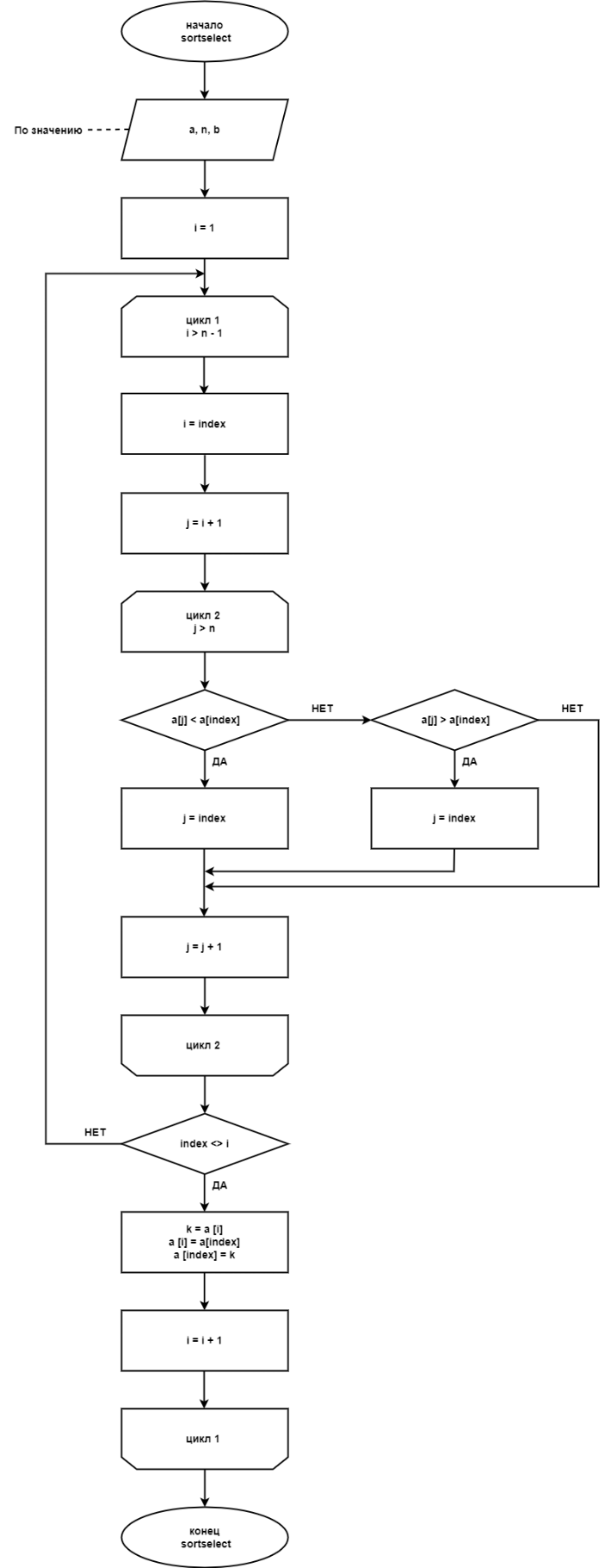
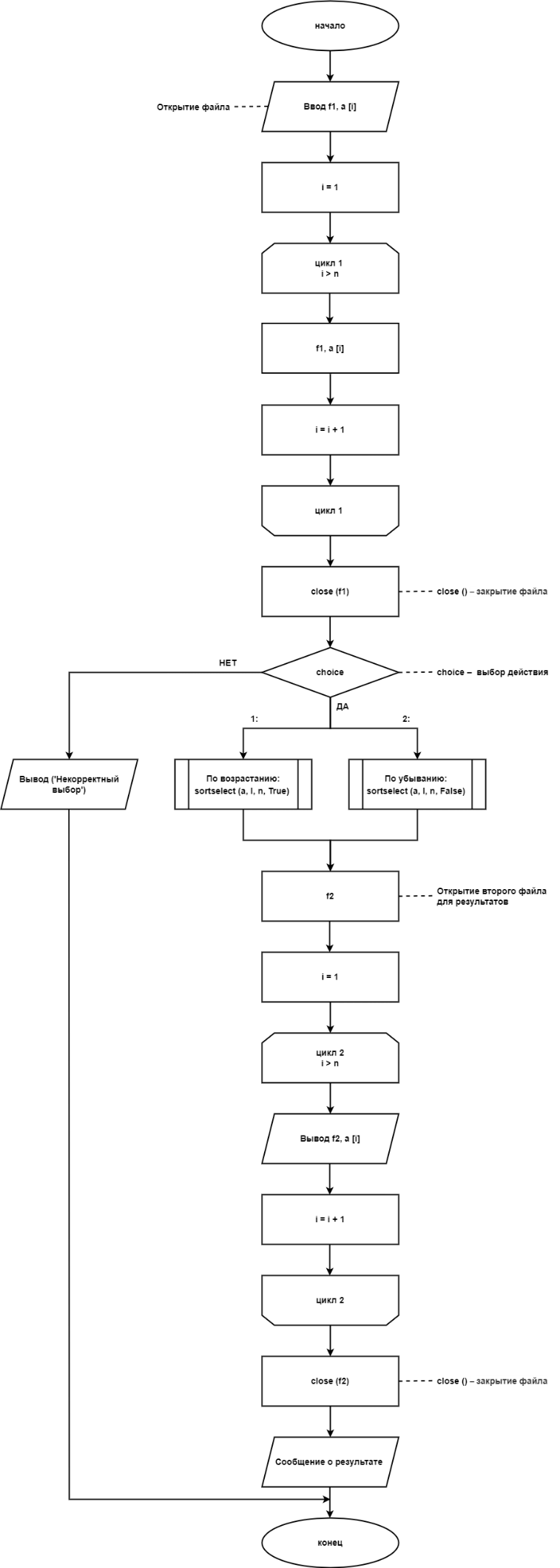
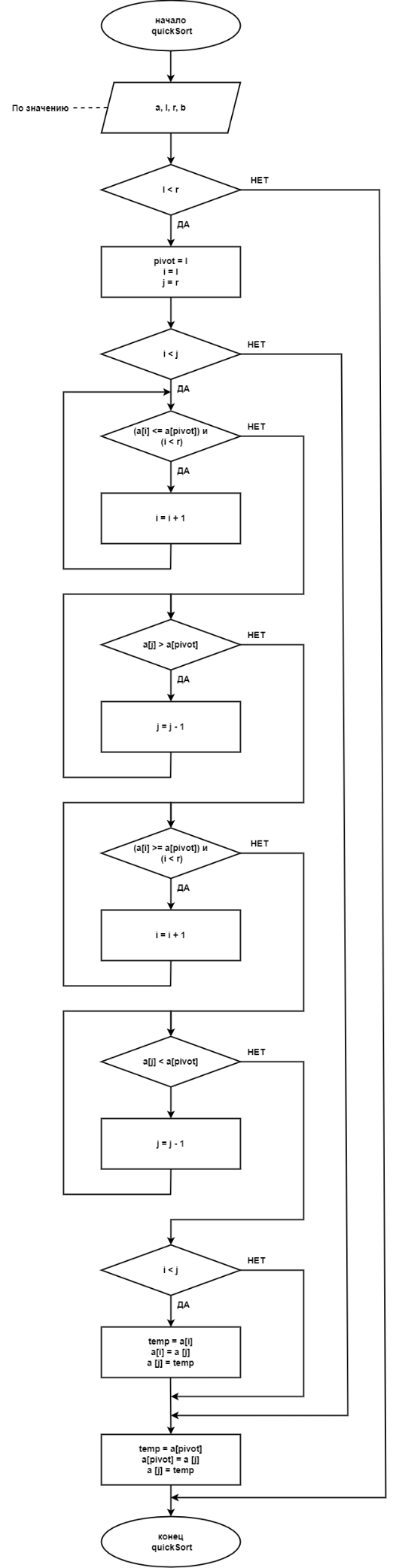
******4. Схема алгоритма с комментариями**

Рис 1. Схема алгоритма (задание 1) с комментариями

Рис 2. Схема алгоритма (задание 2) с комментариями.

**4. Код программы**

**Задача №1**

**program** ДКР5\_1;

**type**

arr = **array**[1..100] **of** Integer;

**var**

f1, f2: text;

i, n: Integer;

a: arr;

**procedure** sortselect(**var** a: arr; n: Integer; b: Boolean);

**var**

i, j, index, k: Integer;

**begin**

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

**begin**

index := i;

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

**begin**

**if** b **then**

**begin**

**if** a[j] < a[index] **then**

index := j;

**end**

**else**

**begin**

**if** a[j] > a[index] **then**

index := j;

**end**;

**end**;

**if** index <> i **then**

**begin**

k := a[i];

a[i] := a[index];

a[index] := k;

**end**;

**end**;

**end**;

**begin**

assign(f1, 'pop.txt');

reset(f1);

readln(f1, n);

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

read(f1, a[i]);

close(f1);

writeln('Выберите порядок сортировки:');

writeln('1. По возрастанию');

writeln('2. По убыванию');

write('Введите номер: ');

**var** choice: Integer;

readln(choice);

**case** choice **of**

1: sortselect(a, n, True); // сортировка по возрастанию

2: sortselect(a, n, False); // сортировка по убыванию

**else**

**begin**

writeln('Некорректный выбор.');

Halt;

**end**;

**end**;

assign(f2, 'outpop.txt');

rewrite(f2);

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

write(f2, a[i], ' ');

close(f2);

writeln('Сортировка прошла успешно! Результат записан в файл outpop.txt!');

**end**.

**Задача №2**

**program** DKR5\_2;

**type**

arr = **array**[1..100] **of** integer;

**var**

f1, f2: text;

i, n: integer;

a: arr;

**procedure** quickSort(**var** a: arr; l, r: integer; b: boolean);

**var**

i, j, pivot, temp: integer;

**begin**

**if** l < r **then**

**begin**

pivot := l;

i := l;

j := r;

**while** i < j **do**

**begin**

**if** b **then**

**begin**

**while** (a[i] <= a[pivot]) **and** (i < r) **do**

i := i + 1;

**while** (a[j] > a[pivot]) **do**

j := j - 1;

**end**

**else**

**begin**

**while** (a[i] >= a[pivot]) **and** (i < r) **do**

i := i + 1;

**while** (a[j] < a[pivot]) **do**

j := j - 1;

**end**;

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

**begin**

temp := a[i];

a[i] := a[j];

a[j] := temp;

**end**;

**end**;

temp := a[pivot];

a[pivot] := a[j];

a[j] := temp;

{quickSort(a, l, j - 1, b);

quickSort(a, j + 1, r, b);}

**end**;

**end**;

**begin**

assign(f1, 'input.txt');

reset(f1);

readln(f1, n);

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

read(f1, a[i]);

close(f1);

writeln('выберите порядок быстрой сортировки: ');

writeln('1. по возрастанию');

writeln('2. по убыванию');

write('введите номер: ');

**var** choice: integer;

readln(choice);

**case** choice **of**

1: quickSort(a, 1, n, true); // сортировка по возрастанию

2: quickSort(a, 1, n, false); // сортировка по убыванию

**else**

**begin**

writeln('некорректный выбор.');

halt;

**end**;

**end**;

assign(f2, 'output.txt');

rewrite(f2);

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

write(f2, a[i], ' ');

close(f2);

writeln('сортировка прошла успешно! результат записан в файл output.txt!');

**end**.

**5. Результат выполнения программы**

**Задача №1**

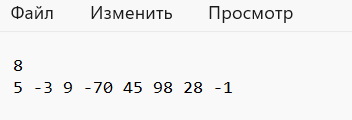
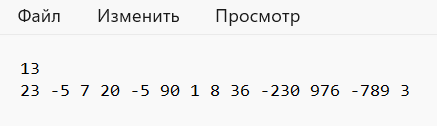
***(Исходный файл)***

Рис.3. Результат выполнения программы (задание 1)

***(Отсортированный файл***

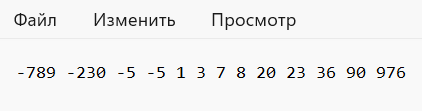
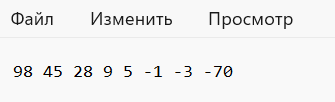
***1 по возрастанию, 2 по убыванию)***

Рис.4. Результат выполнения программы (задание 1)

**Задача № 2**

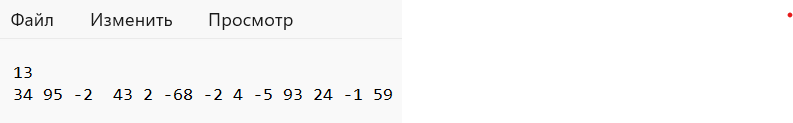
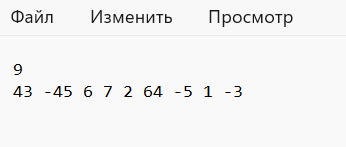
***(Исходный файл)***

Рис.5. Результат выполнения программы (задание 2)

***(Отсортированный файл***

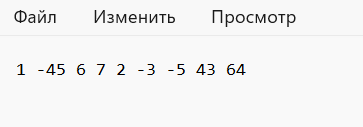
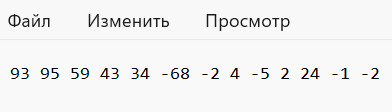
***1 по возрастанию, 2 по убыванию)***

Рис.6. Результат выполнения программы (задание 2)**6. Вывод**

В ходе этой лабораторной работы были изучены базовые структуры организации программы и основные конструкции языка программирования Python. Была создана программа, сортирующая числа, которая берёт числа из файла. Во время сортировки чисел были изучены некоторые алгоритмы сортировки, такие как выбором и быстрый.

Алгоритм выбора — это простой алгоритм сортировки, который сортирует массив элементов в порядке возрастания или убывания. Этодостигается путем обхода массива для поиска минимального элемента и замены его элементом с индексом 0 в отсортированном подмассиве. Затем несортированный подмассив снова просматривается, и минимальный элемент находится и переставляется в правильное положение.

Быстрый алгоритм также относится к алгоритмам «разделяй и властвуй». Его используют чаще других алгоритмов, описанных в этой статье. При правильной конфигурации он чрезвычайно эффективен и не требует дополнительной памяти, в отличие от сортировки слиянием. Массив разделяется на две части по разные стороны от опорного элемента. В процессе сортировки элементы меньше опорного помещаются перед ним, а равные или большие — позади.

В процессе выполнения работы, для создания блок схемы мы познакомились с программой Draw io. Draw.io — инструмент для создания диаграмм, схем алгоритмов, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя.