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

Гущин Тимофей Михайлович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель работы:**

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

1. **Формулировка задания:**

Вариант: 8

1. Реализовать сортировку данных с помощью алгоритма выбором.
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 реализаций самостоятельно подготовить варианты входных данных (при этом объем текстовых файлов должен позволять оценить скорость работы программ).
6. **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 условие, 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 Radix Sort происходит сортировка значение по методу поразрядного алгоритма. Считывается минимальный индекс массива и с помощью цикла перебирается до максимального индекса. Входные данные берутся из текстового файл, соответственно результат выводится так же в другой текстовый файл. Далее следует вычислению с помощью метода.

Во второй части кода происходит записывание результата в текстовый файл. Открывается текстовый файл, с помощью цикла данные из файла перебираются и считываются программой. После для сохранности первый файл закрывается и отрывается второй файл куда заносится результат. После для сохранения данных файл так же закрывается и результат сохраняется. Программа выводит сообщение о успешной работе программы.

1. **Схемы алгоритмов:**

**Задание 1.**

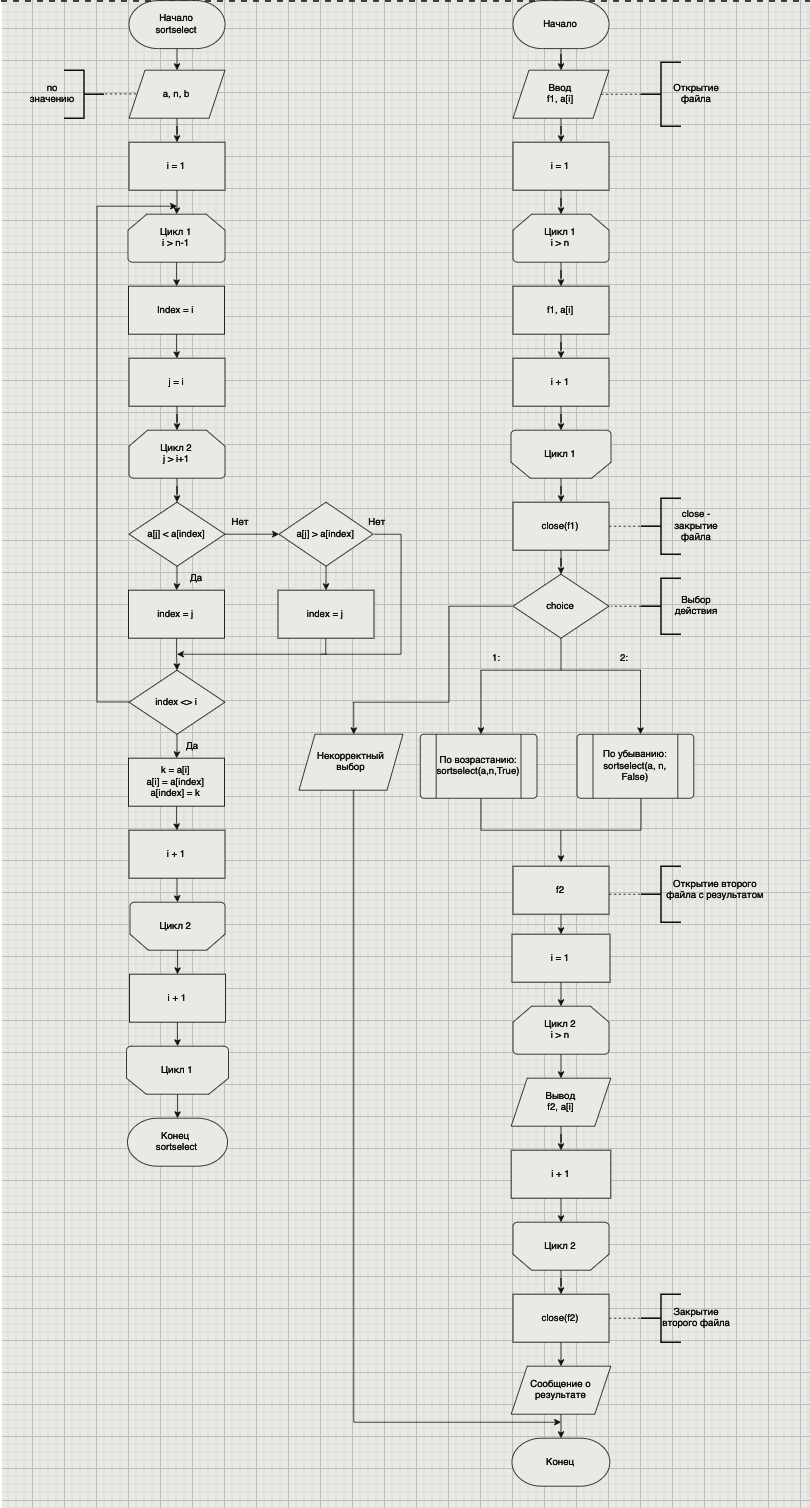
****

Рисунок 1 – Схема Алгоритма.

**Задание 2.**

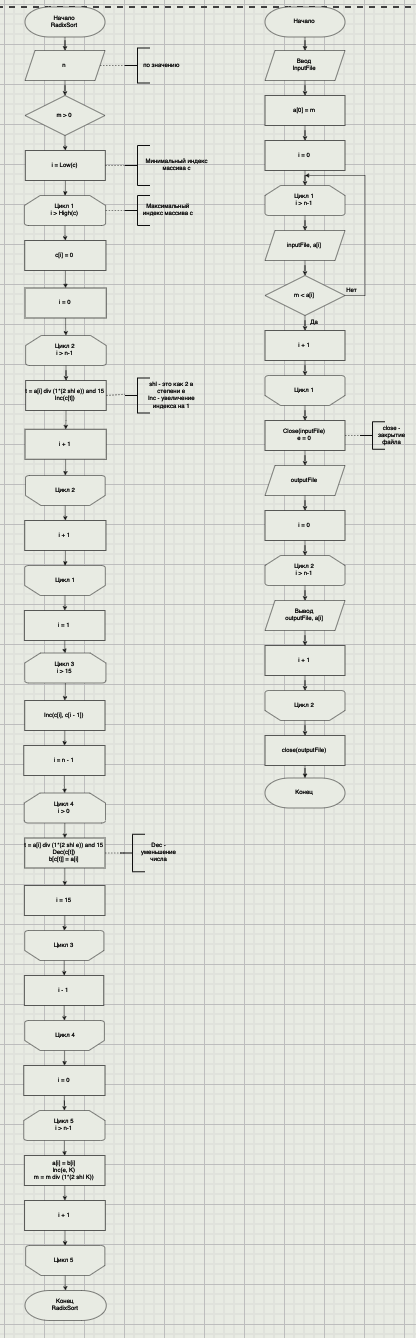
****

Рисунок 2 – Схема Алгоритма.

1. **Код программы:**

**Задание 1.**

**program** DKR5;

**type**

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

**var**

f1, f2: text; // файловые переменные

i, n: Integer;

a: arr;

**procedure** sortselect(**var** a: arr; n: Integer; b: Boolean); // упорядочивание элементов массива

**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, 'popit.txt'); // открытие файла с данными

reset(f1);

readln(f1, n);

**for** i := 1 **to** n **do** // счетчик элементов в файле

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, 'outpopit.txt'); // открытие файла с результатом

rewrite(f2);

**for** i := 1 **to** n **do** // запись отсортированных в файл

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

close(f2);

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

**end**.

**Задание 2.**

**Program** DKR5;

**const**

Sz = 100;

K = 4;

**var**

a, b: **array**[0..Sz] **of** Integer;

c: **array**[0..15] **of** Integer;

m, n, i, t: Integer;

e: Integer;

inputFile, outputFile: Text; // тектовые файлы

**procedure** RadixSort(i: integer);

**begin**

**while** m > 0 **do**

**begin**

**for** i := Low(c) **to** High(c) **do** // обнуление массива c

c[i] := 0;

**for** i := 0 **to** n - 1 **do** // для подсчета элементов в каждом разряде

**begin**

t := a[i] **div** (1 \* (2 **shl** e)) **and** 15;

Inc(c[t]);

**end**;

**for** i := 1 **to** 15 **do** // вычисление префиксных сумм для массива с

Inc(c[i], c[i - 1]);

**for** i := n - 1 **downto** 0 **do** // перемешивание элементов с учетом разряда

**begin**

t := a[i] **div** (1 \* (2 **shl** e)) **and** 15;

Dec(c[t]);

b[c[t]] := a[i];

**end**;

**for** i := 0 **to** n - 1 **do** // копирование массива b обратно в массив a

a[i] := b[i];

Inc(e, K);

m := m **div** (1 \* (2 **shl** K)); // обновление переменной m со сдвигом вправо

**end**;

**end**; // конец процедуры

**begin**

Assign(inputFile, 'rawr.txt'); // привязка файла к переменной

Reset(inputFile);

Read(inputFile, n);

Read(inputFile, m);

a[0] := m; // присвоение элемента m

**for** i := 1 **to** n - 1 **do** // для оставшихся элементов массива а из файла

**begin**

Read(inputFile, a[i]);

**if** m < a[i] **then**

m := a[i];

**end**;

Close(inputFile); // закрытие файла с входными данными

e := 0;

RadixSort(i);

Assign(outputFile, 'outrawr.txt'); // запись отсортированных значений в файл

Rewrite(outputFile);

**for** i := 0 **to** n - 1 **do**

Write(outputFile, a[i], ' ');

WriteLn(outputFile);

Close(outputFile); // закрытие файла с выходными данными

Writeln('Результат успешно сохранен в файл outrawr.txt');

**end**.

1. **Результат выполнения программы:**

**Задание 1.**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5  23 12 45 8 34 | 45 34 23 12 8 |
| 8  5 17 29 8 12 3 41 20 | 41 29 20 17 12 8 5 3 |
| 3  100 50 75 | 100 75 50 |
| 1  99 | 99 |
| 6  10 20 30 40 50 60 | 60 50 40 30 20 10 |

**Задание 2.**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 8  52 64 12 76 87 98 13 65 | 12 13 52 64 65 76 87 98 |
| 5  13 56 76 24 66 | 13 24 56 66 76 |
| 9  34 65 76 23 56 78 98 11 34 | 11 23 34 34 56 65 76 78 98 |
| 3  53 75 24 | 24 53 75 |
| 10  34 65 76 35 98 67 46 22 54 55 | 22 34 35 46 54 55 65 67 76 98 |

1. **Вывод.**

Данная работа прошла успешно. В работе были рассмотрены операции с сортировками чисел на языке программирования Pascal. А также работа с текстовыми файлами, а также взятие из них входных данных и вывода готовых данных в файл.

В работе были написаны программы по сортировке с алгоритмом выбора, а также с поразрядным алгоритмом сортировки.

Трудности в работе возникли в построении схемы алгоритма и соединении связей в схеме алгоритма.

Таким образом, в данной работе были операции исследованием алгоритмов сортировки, построение схемы алгоритма, был описан каждой алгоритм, и проверка данной программы на работоспособность и анализ результата по ранее написанному коду.