**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

Алембаев Артём Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

1. **Цель работы**

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

1. **Формулировка задания (с вариантом)**

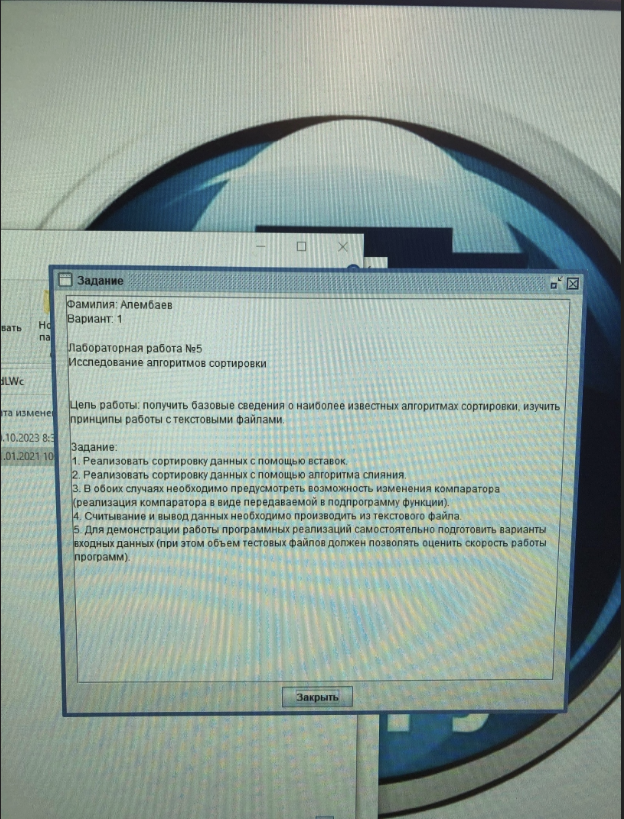
****

Рисунок 1 – Вариант

1. **Описание алгоритма**

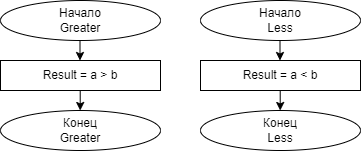
Данный алгоритм выполняет сортировку массива arr методом слияния (MergeSort) или методом вставки (InsertionSort), в зависимости от выбора пользователя.

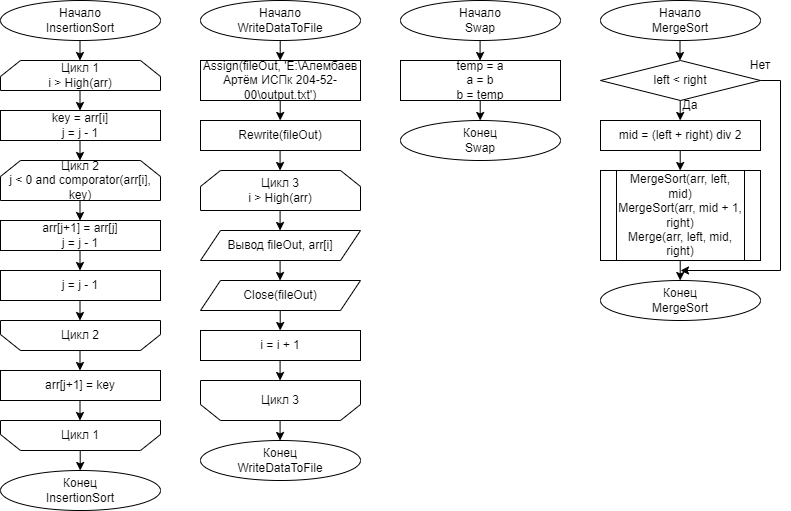
Алгоритм MergeSort разбивает исходный массив на две половины, сортирует каждую из половин рекурсивно с помощью вызова функции MergeSort, а затем объединяет эти половины с помощью функции Merge. Функция Merge соединяет два отсортированных массива L и R в один отсортированный массив arr.

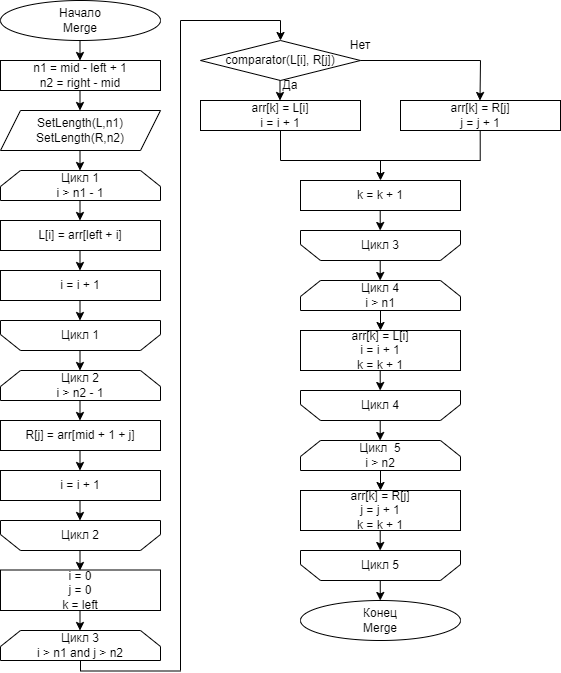
Алгоритм InsertionSort проходит по элементам массива arr и на каждой итерации вставляет текущий элемент на правильное место в уже отсортированной части массива. Этот процесс продолжается до тех пор, пока весь массив не будет отсортирован.

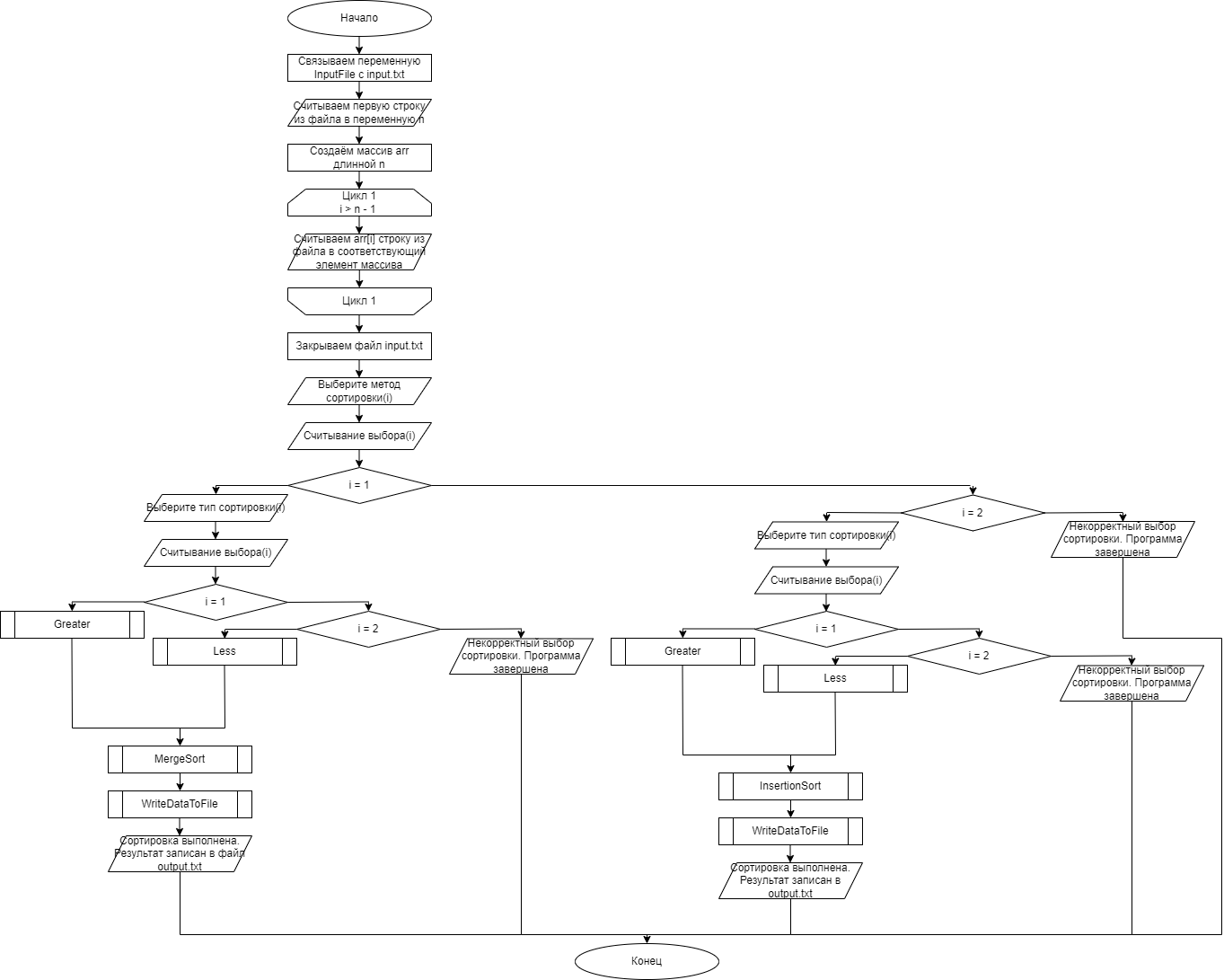
После сортировки массива arr выбранным методом, результат записывается в файл output.txt с помощью функции WriteDataToFile.

1. **Схема алгоритма с комментариями**









1. **Код программы**

type

TComparator = function(a, b: Integer): Boolean;

var

comparator: TComparator;

function Greater(a, b: Integer): Boolean;

begin

Result := a > b;

end;

function Less(a, b: Integer): Boolean;

begin

Result := a < b;

end;

procedure Swap(var a, b: Integer);

var

temp: Integer;

begin

temp := a;

a := b;

b := temp;

end;

procedure Merge(var arr: array of Integer; left, mid, right: Integer);

var

i, j, k: Integer;

n1, n2: Integer;

L, R: array of Integer;

begin

n1 := mid - left + 1;

n2 := right - mid;

SetLength(L, n1);

SetLength(R, n2);

for i := 0 to n1 - 1 do

L[i] := arr[left + i];

for j := 0 to n2 - 1 do

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

i := 0;

j := 0;

k := left;

while (i < n1) and (j < n2) do

begin

if comparator(L[i], R[j]) then

begin

arr[k] := L[i];

Inc(i);

end

else

begin

arr[k] := R[j];

Inc(j);

end;

Inc(k);

end;

while i < n1 do

begin

arr[k] := L[i];

Inc(i);

Inc(k);

end;

while j < n2 do

begin

arr[k] := R[j];

Inc(j);

Inc(k);

end;

end;

procedure MergeSort(var arr: array of Integer; left, right: Integer);

var

mid: Integer;

begin

if left < right then

begin

mid := (left + right) div 2;

MergeSort(arr, left, mid);

MergeSort(arr, mid + 1, right);

Merge(arr, left, mid, right);

end;

end;

procedure InsertionSort(var arr: array of Integer);

var

i, j, key: Integer;

begin

for i := 1 to High(arr) do

begin

key := arr[i];

j := i - 1;

while (j >= 0) and comparator(arr[j], key) do

begin

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

j := j - 1;

end;

arr[j + 1] := key;

end;

end;

procedure WriteDataToFile(arr: array of Integer);

var

fileOut: Text;

i: Integer;

begin

Assign(fileOut, 'E:\Алембаев Артём ИСПк 204-52-00\output.txt');

Rewrite(fileOut);

for i := 0 to High(arr) do

Writeln(fileOut, arr[i]);

Close(fileOut);

end;

var

arr: array of Integer;

n, i: Integer;

inputFile: Text;

begin

Assign(inputFile, 'E:\Алембаев Артём ИСПк 204-52-00\input.txt');

Reset(inputFile);

ReadLn(inputFile, n);

SetLength(arr, n);

for i := 0 to n - 1 do

begin

ReadLn(inputFile, arr[i]);

end;

Close(inputFile);

Write('Выберите метод сортировки (1 - сортировка слиянием, 2 - сортировка вставками): ');

ReadLn(i);

if i = 1 then

begin

Write('Выберите тип сортировки (1 - по убыванию, 2 - по возрастанию): ');

ReadLn(i);

if i = 1 then

comparator := Greater

else if i = 2 then

comparator := Less

else

begin

Writeln('Некорректный выбор сортировки. Программа завершена.');

Exit;

end;

MergeSort(arr, 0, High(arr));

end

else if i = 2 then

begin

Write('Выберите тип сортировки (1 - по возрастанию, 2 - по убыванию): ');

ReadLn(i);

if i = 1 then

comparator := Greater

else if i = 2 then

comparator := Less

else

begin

Writeln('Некорректный выбор сортировки. Программа завершена.');

Exit;

end;

InsertionSort(arr);

end

else

begin

Writeln('Некорректный выбор сортировки. Программа завершена.');

Exit;

end;

WriteDataToFile(arr);

Writeln('Сортировка выполнена. Результат записан в файл output.txt');

end.

1. **Результат выполнения программы**

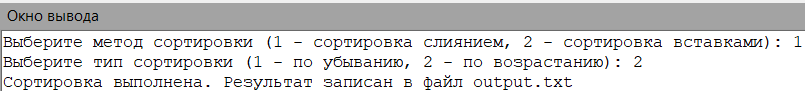


Рисунок 2 – Сортировка слиянием

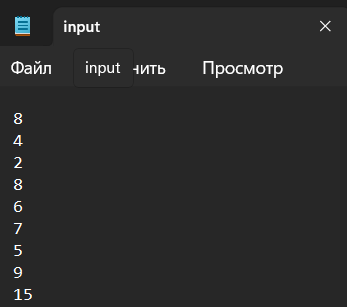


Рисунок 3 – Вот так выглядит input

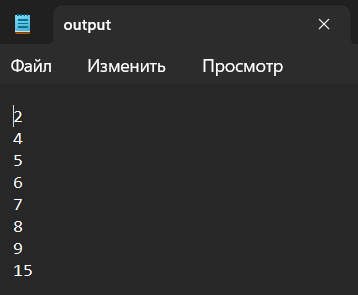


Рисунок 4 – Вот так выглядит сортировка слиянием

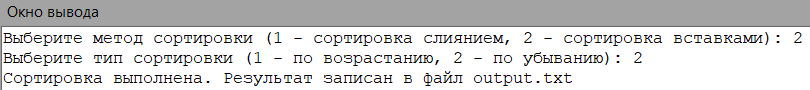


Рисунок 5 – Сортировка вставками

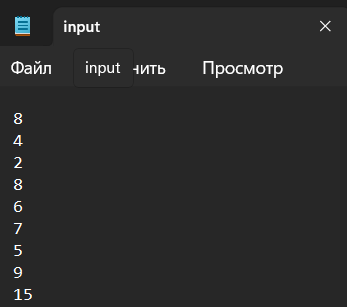


Рисунок 6 – Вот так выглядит input

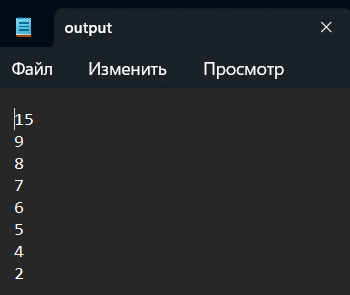


Рисунок 4 – Вот так выглядит сортировка вставками

1. **Вывод**

Были изучены способы сортировки данных, которых можно представить в любом виде чисел. Изучены сортировки алгоритмом выбора и поразрядным алгоритмом. Изучены способы работы с файлами, запись в файл через разделительный символ, запись в файл через строку, чтение файла через строку. Изучен способ разбиения сортировочной части на множество частей как в поразрядном алгоритме сортировки. Изучена теория сортировочных алгоритмов. Изучены способы значительного увеличения быстродействия алгоритмов. Изучены влияния типов данных, так как в переменные типа byte иногда не мог вместиться индекс числа пришлось использовать довольно требовательный тип integer. Все задачи были выполнены, а цель достигнута.