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-205-52-00 Головёшкин Андрей Евгеньевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Вариант 2**

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

**Задание:**

1. Реализовать сортировку данных с помощью "пузырькового" алгоритма.

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 тестовых файлов должен позволять оценить скорость работы программ).

**Описание алгоритма:**

1. **Определение компараторов:**

- Ascending: возвращает True, если первый элемент больше второго (для сортировки по возрастанию).

- Descending: возвращает True, если первый элемент меньше второго (для сортировки по убыванию).

1. **Пузырьковая сортировка (`BubbleSort`):**

- Для каждого элемента массива:

- Проходим по всем парам соседних элементов.

- Если они нарушают порядок (в зависимости от компаратора), меняем их местами.

1. **Поразрядная сортировка (`RadixSort`):**

- Для каждого разряда (начиная с младшего):

- Сортируем элементы по текущему разряду с помощью подсчётной сортировки.

- После обработки всех разрядов, если компаратор указывает на убывание, переворачиваем массив.

1. **Чтение данных из файла (`ReadDataFromFile`):**

- Открываем файл, считываем все числа и сохраняем их в массив.

1. **Запись данных в файл (`WriteDataToFile`):**

- Записываем отсортированные числа в указанный файл.

1. **Основная программа:**

- Читаем исходные данные из файла.

- Сортируем массив пузырьковым методом по убыванию и

записываем результат в один файл.

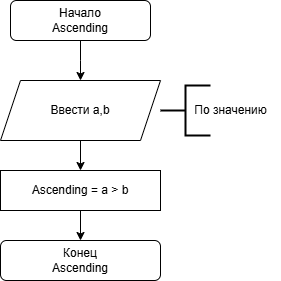
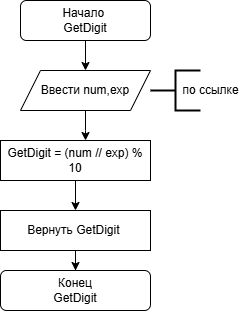
- Сортируем массив поразрядным методом по убыванию и

записываем результат в другой файл.

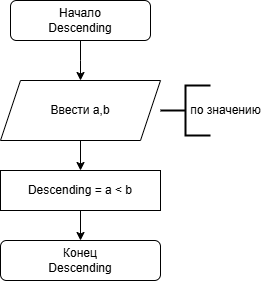
- Выводим сообщение о сохранении файлов.

**Схема алгоритма**

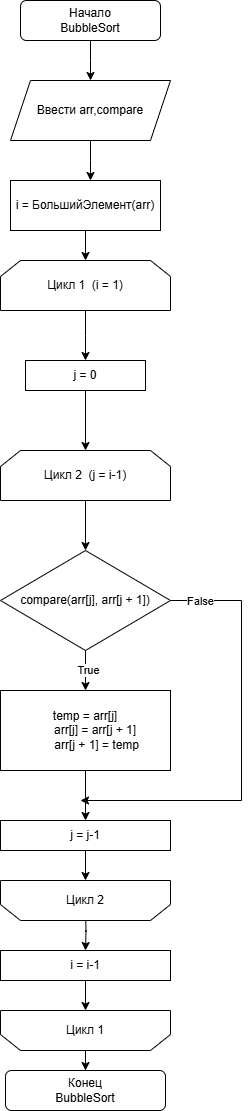
**Процедуры и функции:**

****  

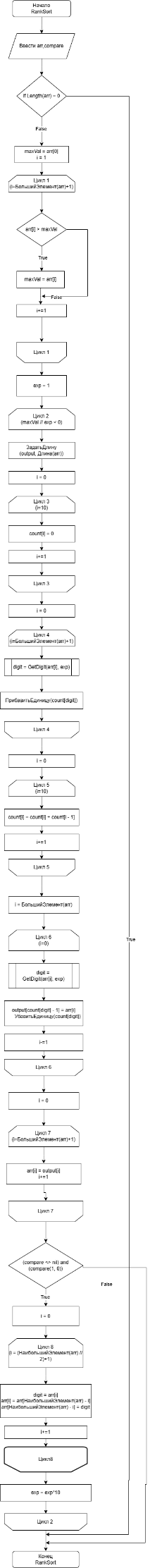
Процедура 1 Функция 1

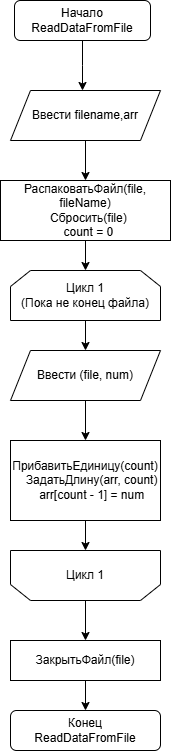
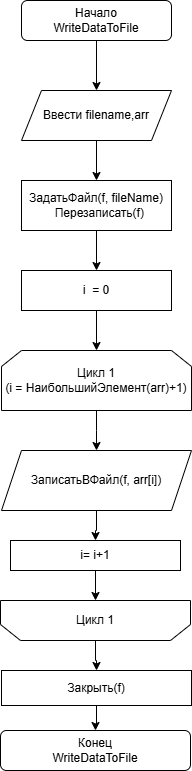


*Процедура 1*

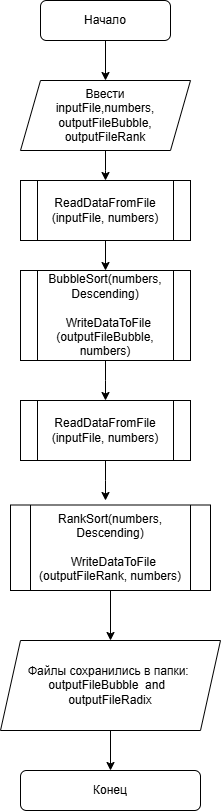


Процедура 3



Процедура 5 Процедура 6



Основная программа

**Код программы**

program my52;

type

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

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

begin

Ascending := a > b;

end;

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

begin

Descending := a < b;

end;

procedure BubbleSort(var arr: array of Integer; compare: Comparator);

var

i, j, temp: Integer;

begin

for i := High(arr) downto 1 do

for j := 0 to i - 1 do

if compare(arr[j], arr[j + 1]) then

begin

temp := arr[j];

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

arr[j + 1] := temp;

end;

end;

procedure ranksort(var arr: array of Integer; compare: Comparator);

var

i, digit, maxVal, exp: Integer;

output: array of Integer;

count: array[0..9] of Integer;

function GetDigit(num, exp: Integer): Integer;

begin

GetDigit := (num div exp) mod 10;

end;

begin

if Length(arr) = 0 then

Exit;

maxVal := arr[0];

for i := 1 to High(arr) do

if arr[i] > maxVal then

maxVal := arr[i];

exp := 1;

while maxVal div exp > 0 do

begin

SetLength(output, Length(arr));

for i := 0 to 9 do

count[i] := 0;

for i := 0 to High(arr) do

begin

digit := GetDigit(arr[i], exp);

Inc(count[digit]);

end;

for i := 1 to 9 do

count[i] := count[i] + count[i - 1];

for i := High(arr) downto 0 do

begin

digit := GetDigit(arr[i], exp);

output[count[digit] - 1] := arr[i];

Dec(count[digit]);

end;

for i := 0 to High(arr) do

arr[i] := output[i];

exp := exp \* 10;

end;

if (compare <> nil) and (compare(1, 0)) then

begin

for i := 0 to (High(arr) div 2) do

begin

digit := arr[i];

arr[i] := arr[High(arr) - i];

arr[High(arr) - i] := digit;

end;

end;

end;

procedure ReadDataFromFile(const fileName: string; var arr: array of Integer);

var

f: TextFile;

num: Integer;

count: Integer;

begin

AssignFile(f, fileName);

Reset(f);

count := 0;

while not Eof(f) do

begin

ReadLn(f, num);

Inc(count);

SetLength(arr, count);

arr[count - 1] := num;

end;

CloseFile(f);

end;

procedure WriteDataToFile(const fileName: string; const arr: array of Integer);

var

f: TextFile;

i: Integer;

begin

AssignFile(f, fileName);

Rewrite(f);

for i := 0 to High(arr) do

WriteLn(f, arr[i]);

CloseFile(f);

end;

var

numbers: array of Integer;

inputFile, outputFileBubble, outputFileRadix: string;

begin

inputFile := 'D:\train vscode\python\train\random\_numbers.txt';

outputFileBubble := 'D:\test\bubble\_sorted.txt';

outputFileRadix := 'D:\test\radix\_sorted.txt';

ReadDataFromFile(inputFile, numbers);

BubbleSort(numbers, Descending);

WriteDataToFile(outputFileBubble, numbers);

ReadDataFromFile(inputFile, numbers);

ranksort(numbers, Descending);

WriteDataToFile(outputFileRadix, numbers);

WriteLn('Файлы сохранились в папки: ', outputFileBubble, ' and ', outputFileRadix);

end.

**Вывод:**

Программа реализует два алгоритма сортировки — пузырьковую и поразрядную. Она считывает данные из файла, сортирует их с использованием компараторов для выбора порядка сортировки и записывает результаты в два разных файла. Программа демонстрирует основы работы с файлами и сортировками на языке Pascal.