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-202-52-00

Савватеев Аркадий Сергеевич

Преподаватель:

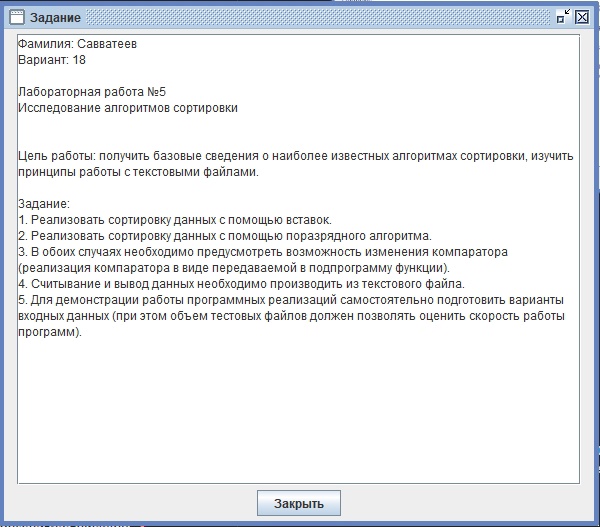
Сергеева Елизавета Григорьевна

Киров

2024

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

1. **Формулировка задания**



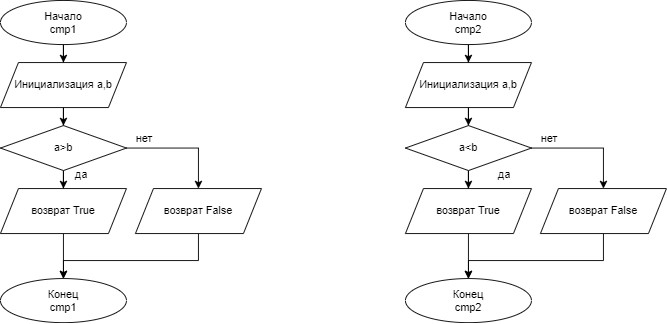
1. **Описание алгоритма**
2. Сортировка вставками. Это алгоритм, который упорядочивает элементы массива, вставляя каждый новый элемент на своё место среди уже отсортированных элементов. Алгоритм работает по принципу «расширяющегося отсортированного подмассива». Сначала первый элемент массива считается отсортированным. Затем каждый последующий элемент сравнивается с элементами отсортированного подмассива и помещается в нужную позицию.
3. алгоритм поразрядной сортировки гениален в том, что сортирует не числа целиком, а значения разрядов. Получается, что он как бы разбирается с числами на уровне единиц, десятков, сотен и т. д. и только потом он делает общую сортировку. Это позволяет ему не бегать по всем сравниваемым числам и не делать миллион сравнений. Отсюда и экономия времени.
4. **Схема алгоритма с комментариями**

Рисунок 1 – схема алгоритма

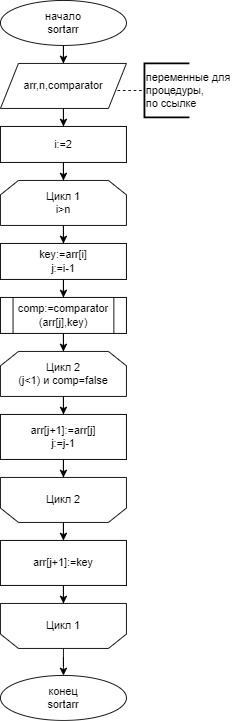


Рисунок 2 – схема алгоритма

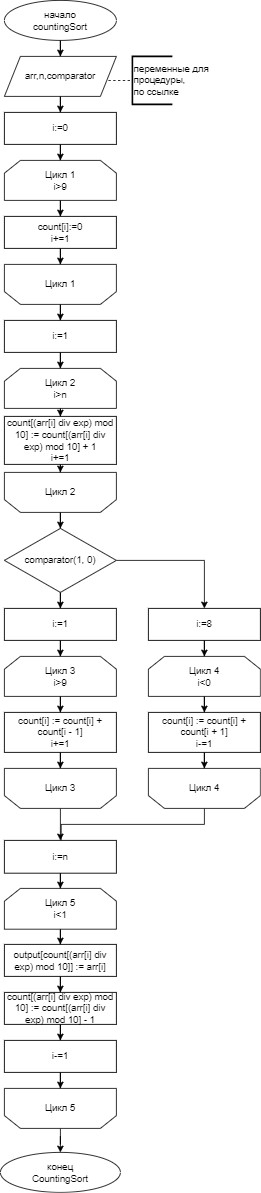


Рисунок 3 – схема алгоритма

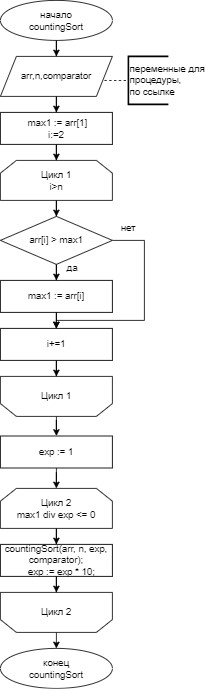


Рисунок 4 – схема алгоритма

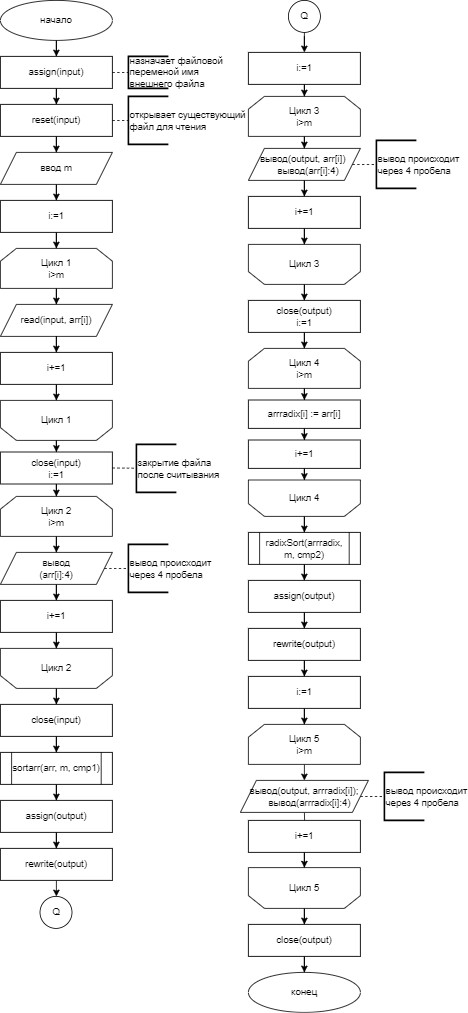


Рисунок 5 – схема алгоритма

**Код программы**

**program** qq;

**type**

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

TComparator = **function**(a, b: integer): boolean;

**var**

arr: TArray;

arrRadix: TArray;

i, m: integer;

input, output: TextFile;

//сортировка по возрастанию

**function** cmp1(a,b:integer):boolean;

**begin**

result:= a>b;

**end**;

//сортировка по убыванию

**function** cmp2(a,b:integer):boolean;

**begin**

result:= a<b;

**end**;

//сортировка вставками

**procedure** sortarr(**var** arr: TArray; n: Integer; comparator: TComparator);

**var**

i, j, key: Integer;

**begin**

**for** i := 2 **to** n **do**

**begin**

key := arr[i];

j := i - 1;

**while** (j >= 1) **and** comparator(arr[j],key) **do**

**begin**

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

j := j - 1;

**end**;

arr[j + 1] := key;

**end**;

**end**;

//вспомагательная функция для сортировки поразрядным алгоритмом

**procedure** countingSort(**var** arr: TArray; n, exp: Integer; comparator: TComparator);

**var**

output: TArray;

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

i: Integer;

**begin**

**for** i := 0 **to** 9 **do**

count[i] := 0;

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

count[(arr[i] **div** exp) **mod** 10] := count[(arr[i] **div** exp) **mod** 10] + 1;

**if** comparator(1, 0) **then**

**begin**

**for** i := 1 **to** 9 **do**

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

**end**

**else**

**begin**

**for** i := 8 **downto** 0 **do**

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

**end**;

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

**begin**

output[count[(arr[i] **div** exp) **mod** 10]] := arr[i];

count[(arr[i] **div** exp) **mod** 10] := count[(arr[i] **div** exp) **mod** 10] - 1;

**end**;

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

arr[i] := output[i];

**end**;

// сортировка поразрядным алгоритмом

**procedure** radixSort(**var** arr: TArray; n: Integer; comparator: TComparator);

**var**

max1, exp, i: Integer;

**begin**

max1 := arr[1];

**for** i := 2 **to** n **do**

**if** arr[i] > max1 **then**

max1 := arr[i];

exp := 1;

**while** max1 **div** exp > 0 **do**

**begin**

countingSort(arr, n, exp, comparator);

exp := exp \* 10;

**end**;

**end**;

**begin**

assign(input, 'd:\дкр 5\input.txt');

reset(input);

writeln('Введите количество чисел в массиве');

readln(m);

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

read(input, arr[i]);

close(input);

writeln('Исходный массив: ');

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

write(arr[i]:4);

writeln;

sortarr(arr, m, cmp1);

assign(output, 'd:\дкр 5\output1.txt');

rewrite(output);

writeln('Отсортированный массив (методом вставки):');

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

**begin**

write(output, arr[i]);

write(arr[i]:4);

**end**;

close(output);

writeln();

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

arrradix[i] := arr[i];

radixSort(arrradix, m, cmp2);

assign(output, 'd:\дкр 5\output2.txt');

rewrite(output);

writeln('Отсортированный массив (методом поразрядного алгоритма):');

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

**begin**

writeln(output, arrradix[i]);

write(arrradix[i]:4);

**end**;

close(output);

**end**.

1. **Результат выполнения программы**

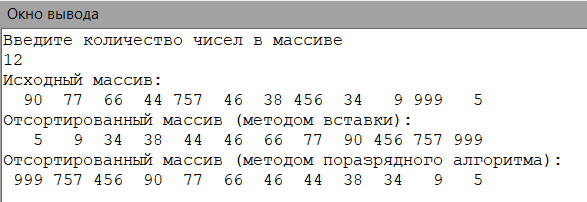
****

Рисунок 7 – Результаты тестирования задач

**Вывод**

В ходе работы над домашней контрольной работе мы освоили синтаксис построения функций, изучили алгоритмы сортировки.

Также были получены знания о том, что такое: функция, сортировка вставками, сортировка слиянием.

Для создания алгоритмических схем было изучено правильное построение письменного алгоритма для достижения верного решения задачи, а также и само построение схем в программе draw.io, бесплатном онлайн-сервисе, который позволяет создавать и совместно редактировать диаграммы, различные схемы и прочие визуальные представления данных.

Однако в ходе работы мы столкнулись и с проблемами. Например, при выполнении программы вместо ожидаемого нам ответа выводился массив в неправильной последовательности. Решением стало изменение условия сортировки в процедурах.

В конечном итоге при помощи полученных знаний и исправленных ошибок у нас получилось добиться поставленной цели – решить задачу при помощи кода с алгоритмами сортировки вставками и слияния при помощи компаратора.