МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Липецкий Государственный Технический Университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Лабораторная работа

по технологиям программирования №1

“Методы сортировки и поиска в информационных массивах”

Студент \_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

Группа АС-21-1

Руководитель

Доцент, кандидат наук \_\_\_\_\_\_\_\_\_\_\_\_ Хромов А. М.

(подпись, дата)

Липецк 2022 г.

**Содержание:**

1. Титульный лист.

2. Задание кафедры, соответствующее варианту, номер варианта.

3. Цель работы.

4. Краткие теоретические сведения

5. Блок-схему алгоритма разработанной программы

6. Текст программы

7. Контрольный пример

8. Выводы по работе.

**2. Задание кафедры, соответствующее варианту, номер варианта.**Осуществить программную реализацию сортировки информации заданного вида сбалансированным N-ленточным слиянием (в оперативной памяти),

используя выбранные из табл. 1, в соответствии с вариантом, алгоритм внутренней сортировки и формат исходных данных.  
Тип ключевого поля записи: int  
Типы информационных полей записи: char  
Метод внутренней сортировки: метод подсчёта

**3. Цель работы:** приобретение навыков реализации базовых алгоритмов сортировки массивов данных.

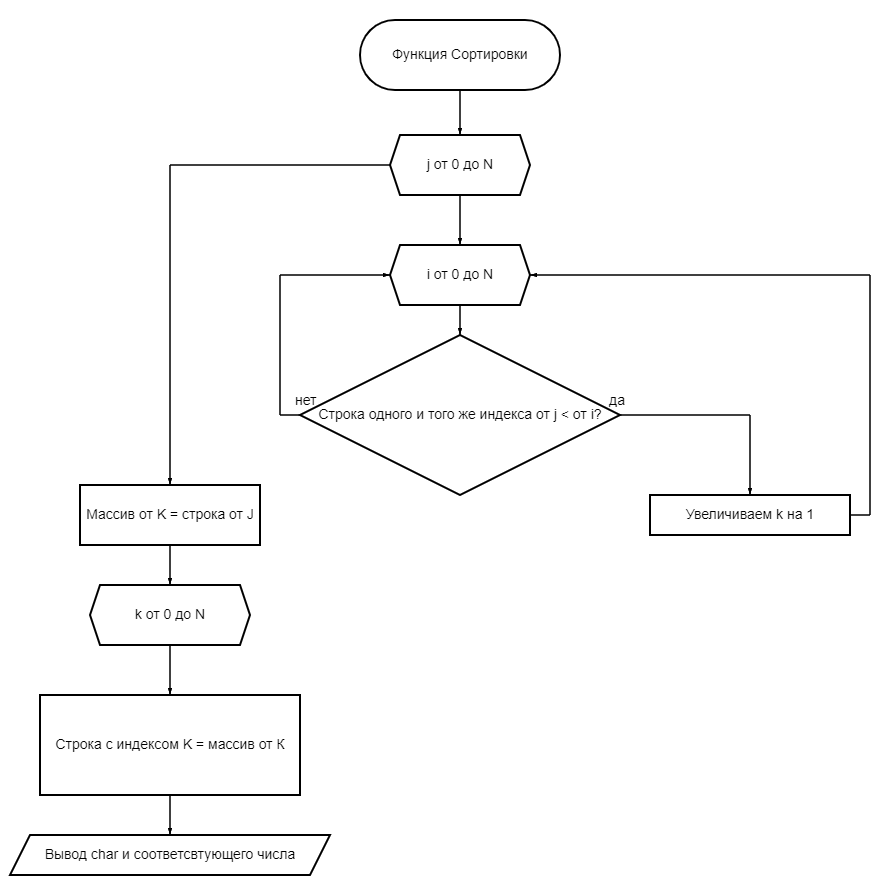
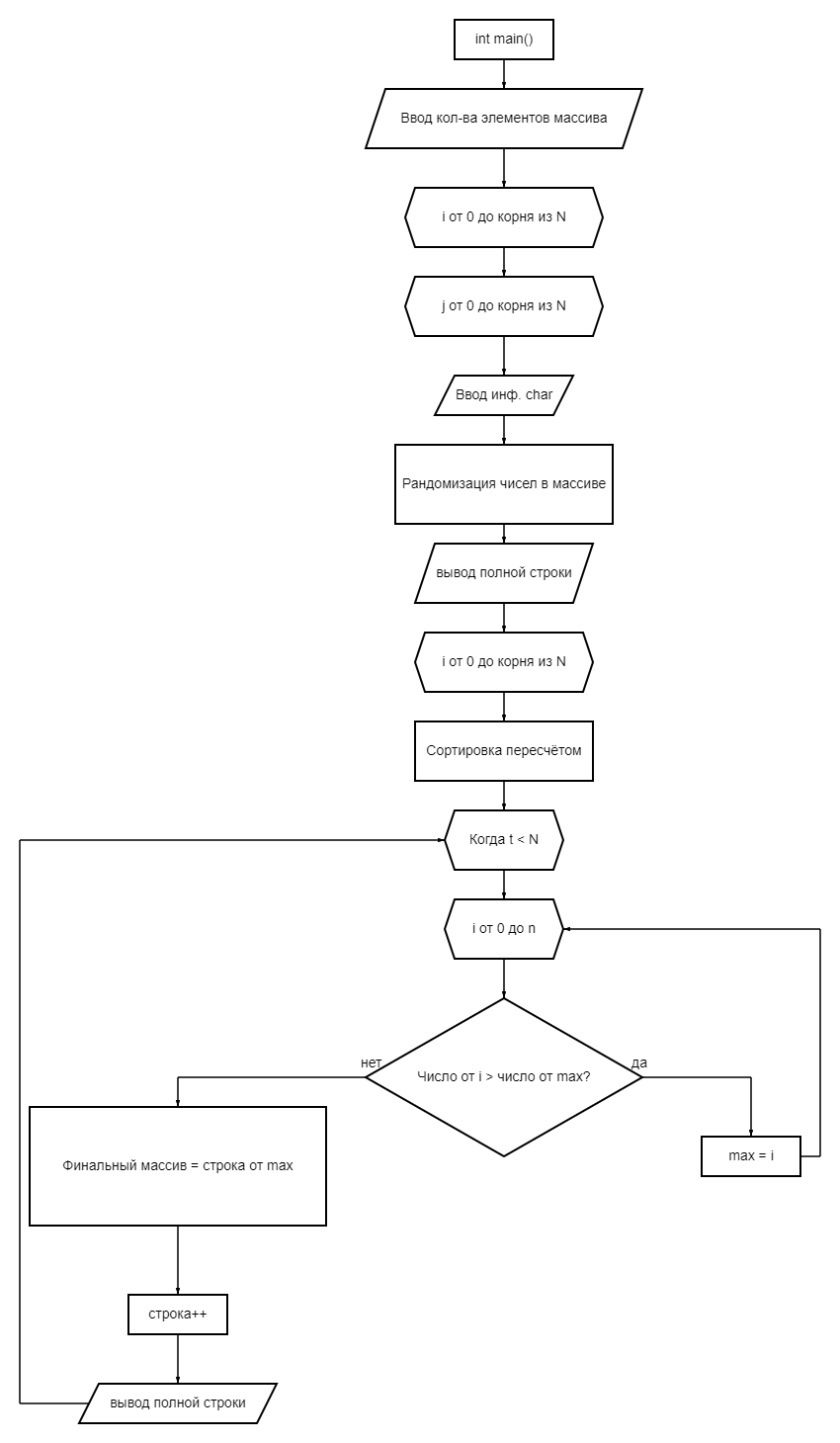
**4. Краткие теоретические сведения**

При использовании этого метода сортировки упорядоченная последовательность B создается на свободном участке памяти. Метод основан на том, что (К + 1)-й элемент упорядоченной последовательности превышает ровно К элементов и, следовательно, занимает (К + 1)-ю позицию. В процессе сортировки на каждом i-м проходе i-й элемент исходной последовательности A попарно сравнивается со всеми остальными элементами. Если в результате сравнения установлено, что Ai > Аj, то значение счетчика К увеличивается на единицу.

Первоначально счетчик K инициализируется нулем. По окончании прохода счетчик К показывает число элементов, меньших чем Аi. Соответственно элемент Аi в последовательности В занимает позицию К+1.

Для данного метода максимальное, минимальное и среднее число сравнений равно 2N.

Рассмотренный алгоритм сортировки методом подсчета можно использовать лишь в тех случаях, когда в исходной последовательности отсутствуют одинаковые элементы, иными словами, когда в упорядочиваемом массиве нет записей с одинаковыми значениями ключей. Для сортировки массивов, имеющих записи с одинаковыми значениями ключей, алгоритм нужно модифицировать.

**5. Блок-схема алгоритма разработанной программы  
**

**6. Текст программы.**#include<string.h>

#include<math.h>

#include<locale.h>

#include<stdlib.h>

#include<stdio.h>

#include<cstring>

struct S { int number; char stroka[20]; };

S resul[8000]; //выделение памяти с запасом))))

void sort(S str[20][400 + 1], int index, int N)

{

int i, j, k;

//сортировка методом подсчета

for (j = 0; j < N; j++)

{

k = 0;

for (i = 0; i < N; i++)

{

if (str[index][j].number < str[index][i].number)

k++;

}

resul[k] = str[index][j];

}

for (k = 0; k < N; k++)

{

str[index][k] = resul[k];

printf("%s: %4d\n", str[index][k].stroka, str[index][k].number);

}

}

S str[20][400 + 1], result[20 \* 400];

int main()

{

int N;

scanf\_s("%d", &N);

setlocale(LC\_ALL, "");

int i, j;

printf(" ----------------Первоначальные данные----------------\n");

for (i = 0; i < sqrt(N); i++)

{

for (j = 0; j < sqrt(N); j++)

{

char key[20];

scanf\_s("%s", key, N);

str[i][j].number = rand() % 10000;

strcpy\_s(str[i][j].stroka, key);

printf("%s: %4d\n", str[i][j].stroka, str[i][j].number);

}

}

printf("\n\n");

printf(" ----------------Отсортированные ленты----------------\n");

for (i = 0; i < sqrt(N); i++)

{

sort(str, i, sqrt(N));

printf("\n");

}

printf(" ----------------Конечные данные----------------\n");

char stroka1[9] = { 0,0,0,0,0,0,0,0,0 }, t = 0;

while (t < N)

{

int max = 0;

for (i = 0; i < N; i++)

{

if (str[i][stroka1[i]].number > str[max][stroka1[max]].number) //проблема с сортировкой

max = i;

}

result[t] = str[max][stroka1[max]];

stroka1[max]++;

printf("\n%s: %4d", result[t].stroka, result[t].number);

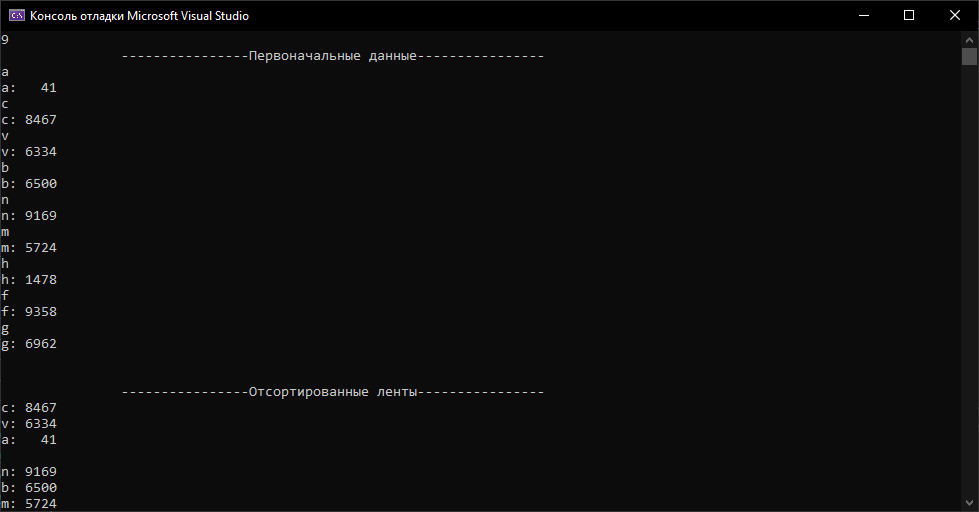
t++;

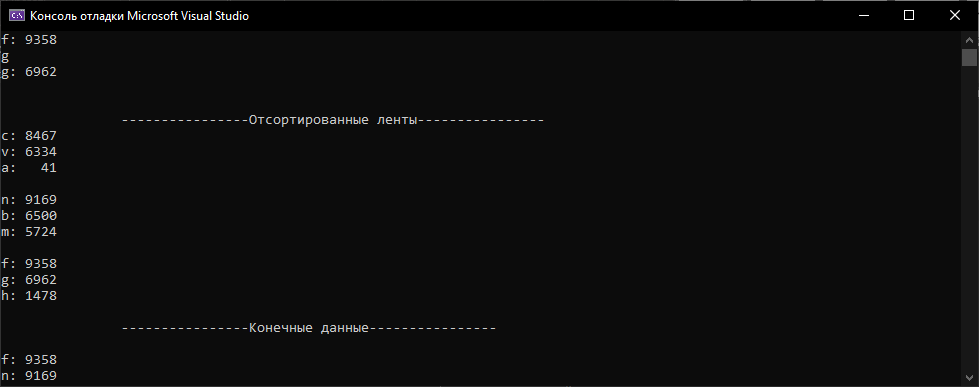
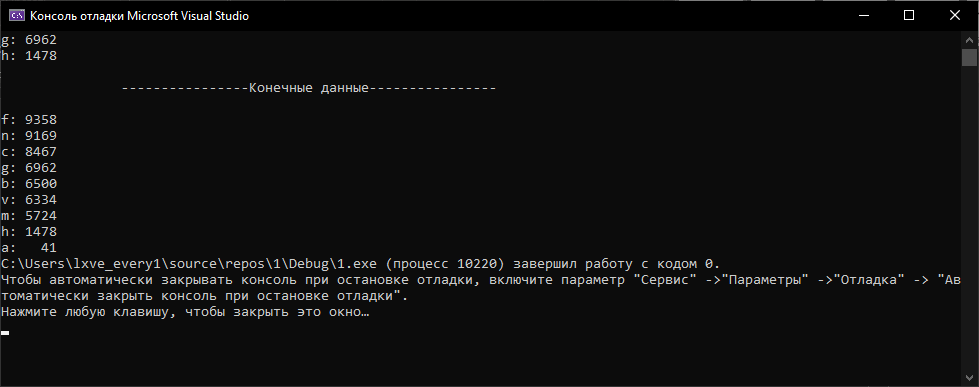
}

return 0;

}

**7. Контрольный пример**Вводим следующие данные

  
  
Промежуточные результаты:

  
Конечные результаты:  


Программа успешно выполнилась и отсортировала массив по ключевому значению key.

**8. Выводы по работе.**Приобрёл навыки реализации базовых алгоритмов сортировки массивов данных.