

Практическая работа № 10

Сроковый тип данных (по желанию)

Цель работы: Получить практический навык работы со строковыми переменными

Теоретический материал:

1. Понятие строка

Строки — это последовательность символов, заключенная в кавычки.

В конце каждой строки компилятор добавляет нулевой символ, представляемый управляющей последовательностью '\0'.

Строка описывается как массив символов. Число элементов массива равно числу элементов в строке плюс символ конца строки (\0). Символьная строка в программе может располагаться на нескольких строках. Для переноса используется *обратная дробная черта* с последующим *нажатием клавиши «ввод»*. Обратная дробная черта игнорируется компилятором, и следующая строка считается продолжением предыдущей.

В классическом языке C строк как таковых нет, есть лишь массивы символов char.

В языке C++ для удобной работы со строками **есть класс string**, для использования которого необходимо подключить заголовочный файл string.

Строки можно объявлять и одновременно присваивать им значения:

```
string S1, S2 = "Hello";
```

Строка S1 будет пустой, строка S2 будет состоять из 5 символов.

К отдельным символам строки можно обращаться по индексу, как к элементам массива или C-строк. Например S[0] - это первый символ строки.

Для того, чтобы узнать длину строки можно использовать метод size() строки. Например, последний символ строки S это S[S.size() - 1].

Для начала работы с классом string требуется подключить следующий заголовочный файл: #include <string>

Для работы со строками также нужно подключить стандартное пространство имён: using namespace std;

Вывод текста только большими буквами

```
#include <iostream>
#include <stdio.h>
#include <locale.h>
```

```

#include <string.h>
using namespace std;

int main() {
    int len;
    char st[250];
    setlocale(LC_ALL, "Russian");
    cout << "Введите строку";
    //gets_s(st, 250); или
    cin >> st;
    len = 0;
    while (st[len] != 0)
    {
        st[len] = toupper(st[len]);
        cout << st[len]; //вывод посимвольно
        len++;
    }
    cout << endl<<st;
    return 0;
}

#include <iostream>
using namespace std;
#include<stdio.h>
#include <string>

int main()
{
    int top, bot;
    char temp;
    string st("qwertyuiop");
    bot = st.size()-1;
    cout << bot<< endl;
    cout << st[bot] << endl;
    for (top = 0, bot; top < bot; top++, bot--)
    {
        temp = st[top];
        st[top] = st[bot];

```

```

        st[bot] = temp;
    }
    cout << st;
    return 0;
}

```

Подсчитать длину введенной строки

```

#include <iostream>
#include <stdio.h>
#include <locale.h>
#include <string.h>
using namespace std;
int strlen(const char* str) {
    int len = 0;
    while (str[len] != 0) len++;
    return len;
}
char myString[250];
int main() {
    setlocale(LC_ALL, "Russian");
    cout << "напишите любую строку";
    gets_s(myString, 250);
    printf("Длина введенной строки - %d\n",
        strlen(myString));
    return 0;
}

```

Определить число пробелов с строке

```

#include <iostream>
#include <stdio.h>
#include <locale.h>
#include <string.h>

using namespace std;
int strlen(const char* str) {
    int sps = 0; int len = 0;
    while (str[len] != 0)
    {
        len++;
        if (str[len] == ' ') sps++;
    }
}

```

```

    }
    return sps;
}
char myString[250];
int main() {
    setlocale(LC_ALL, "Russian");
    cout << "Напишите текст с пробелами";
    gets_s(myString, 250);
    printf("Число пробелов в строке - %d\n", strlen(myString));
    cout << "Число пробелов в строке =" << strlen(myString);
    return 0;
}

```

Основные возможности, которыми обладает класс string:

- инициализация массивом символов (строкой встроенного типа) или другим объектом типа string.
- копирование одной строки в другую, для встроенного типа приходится использовать функцию strcpy;
- доступ к отдельным символам строки для чтения и записи. Во встроенном массиве для этого применяется операция взятия индекса или косвенная адресация с помощью указателя;
- сравнение двух строк на равенство., для встроенного типа используются функции семейства strcmp;
- конкатенация (сцепление) двух строк. Для встроенного типа применяется функция strcat, однако, чтобы получить результат в новой строке, необходимо последовательно задействовать функции strcpy и strcat, а также позаботиться о выделении памяти;
- встроенные средства определения длины строки (функции-члены класса size и length). Узнать длину строки встроенного типа можно только вычислением с помощью функции strlen;
- узнать, пуста ли строка можно выражением !st.size() или empty()

Рассмотрим эти базовые возможности более подробно. Инициализация строк при описании и вычисление длины строки (не включая завершающий нуль-терминатор) выполняется следующим образом:

```

string st("Моя строка\n");
cout << "Длина " << st << ": " << st.size() << " символов, включая \\n";

```

Строка может быть задана пустой:

```
string st2;
```

Для проверки того, пуста ли строка, можно сравнить её длину с нулём:

```
if (!st.size()) // пустая или
if (st.empty()) // пустая
```

Для сцепления строк используется операция сложения (+) или операция сложения с присваиванием (+=).

Пусть даны две строки: `string s1("hello, "); string s2("world\n");`

Мы можем получить третью строку, состоящую из первых двух, таким образом: `string s3 = s1 + s2;` Если же мы хотим добавить `s2` в конец `s1`, мы должны написать: `s1 += s2;`

К отдельным символам объекта типа `string`, как и встроенного типа, можно обращаться с помощью операции взятия индекса.

Вот, например, фрагмент кода, заменяющего все точки символами подчёркивания:

```
string str ("www.disney.com"); int size = str.size();
for (int i = 0; i < size; i++) if (str[i] == '.') str[ i ] = '_'; cout << str;
```

Конструкторы строк

Строки можно создавать с использованием следующих конструкторов:

`string()` - конструктор по умолчанию (без параметров) создает пустую строку.

`string(string & S)` - копия строки `S`

`string(size_t n, char c)` - повторение символа `c` заданное число `n` раз.

`string(size_t c)` - строка из одного символа `c`.

`string(string & S, size_t start, size_t len)` - строка, содержащая не более, чем `len` символов данной строки `S`, начиная с символа номер `start`.

Конструкторы можно вызывать явно, например, так:

```
S += string(10, 'z');
```

В этом примере явно вызывается конструктор `string` для создания строки, состоящей из 10 символов `'z'`.

Неявно конструктор вызывается при объявлении строки с указанием дополнительных параметров. Например, так:

```
string S(10, 'z');
```

В табл. 1 приводится краткое описание основных операторов и функций класса `string`.

Основные методы и операторы класса `string`

Задание символов в строке	
<code>operator=</code>	Присваивает значения строке
<code>assign</code>	Назначает символы строке
Доступ к отдельным символам	
<code>at</code>	Получает указанный символ и проверяет выход индекса за границы
<code>operator[]</code>	Получает указанный символ
<code>front</code>	Получает первый символ
<code>back</code>	Получает последний символ
<code>data</code>	Возвращает указатель на первый символ строки
<code>c_str</code>	Возвращает немодифицируемый массив символов C, содержащий символы строки
Проверка на вместимость строки	
<code>empty</code>	Проверяет, является ли строка пустой
<code>size</code> или <code>length</code>	Возвращает количество символов в строке
<code>max_size</code>	Возвращает максимальное количество символов
<code>reserve</code>	Резервирует место под хранение

Операции над строкой	
<code>clear</code>	Очищает содержимое строки
<code>insert</code>	Вставляет символы
<code>erase</code>	Удаляет символы
<code>push_back</code>	Добавляет символ в конец строки
<code>pop_back</code>	Удаляет последний символ
<code>append</code>	Добавляет символы в конец строки
<code>operator+=</code>	
<code>compare</code>	Сравнивает две строки
<code>replace</code>	Заменяет каждое вхождение указанного символа
<code>substr</code>	Возвращает подстроку
<code>copy</code>	Копирует символы
<code>resize</code>	Изменяет количество хранимых символов
<code>swap</code>	Обменивает содержимое
Поиск в строке	
<code>find</code>	Находит символы в строке
<code>rfind</code>	Находит последнее вхождение подстроки
<code>find_first_of</code>	Находит первое вхождение символов
<code>find_first_not_of</code>	Находит первое вхождение отсутствия символов
<code>find_last_of</code>	Находит последнее вхождение символов
<code>find_last_not_of</code>	Находит последнее вхождение отсутствия символов

Альтернативный путь решения подобных задач – встроенные в современный стандарт C++ готовые алгоритмы.

Например, предыдущее действие мы могли бы выполнить вызовом одной-единственной функции `replace()`:

Здесь использован не метод `replace` класса `string`, а одноимённый алгоритм: Поскольку объект `string` ведёт себя как контейнер, к нему могут применяться и другие алгоритмы. Это позволяет решать задачи, не решаемые напрямую функциями класса `string`.

Практическое задание:

Разработать алгоритм и написать программу на C++ для решения следующих задач

1. Напишите программу, которая вводит строку и выводит ее, сокращая каждый раз на символ до тех пор, пока в строке не останется 1 символ.
2. Напишите программу, определяющую число слов в строке. Одно слово от другого отделяется 1 пробелом.
3. Составьте программу, определяющую, является ли введенное слово числом.

4. Введите 2 целых числа. Преобразуйте числа в две строки, объедините их в одну строку и выведите на экран результат.
5. Напишите программу, которая удаляет из введенной строки любой требуемый введенный с клавиатуры символ.
6. Выясните, какая из букв первая или последняя встречается в заданном слове чаще.
7. Сколько букв "у" в слове стоит на четных местах?
8. Замените в заданном слове все буквы "о" пробелами.
- 9 В тексте, состоящем из латинских букв и заканчивающемся точкой, подсчитайте количество гласных букв.
10. Заданы фамилия, имя и отчество учащегося, разделенные пробелом. Напечатайте его фамилию и инициалы.
11. Дан текст, в котором слова разделены пробелами. Сколько слов в тексте? Найдите самое длинное слово текста (длина текста 100 символов).
12. Составьте программу, которая запрашивает 2 строки, из четырех цифр. Если в строке введены не цифры, то вывести сообщение об ошибке и прервать программу.
13. Вычислите длину самого короткого слова в предложении из трех слов, разделенное пробелами.

Практическая работа № 11

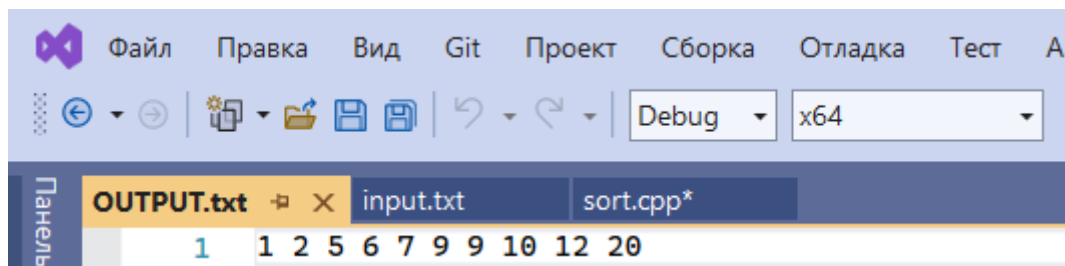
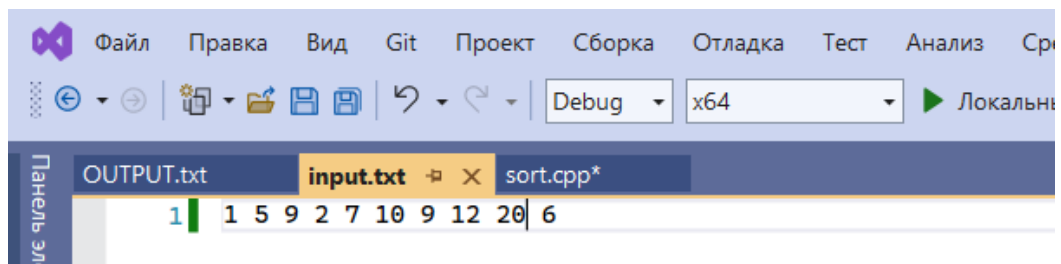
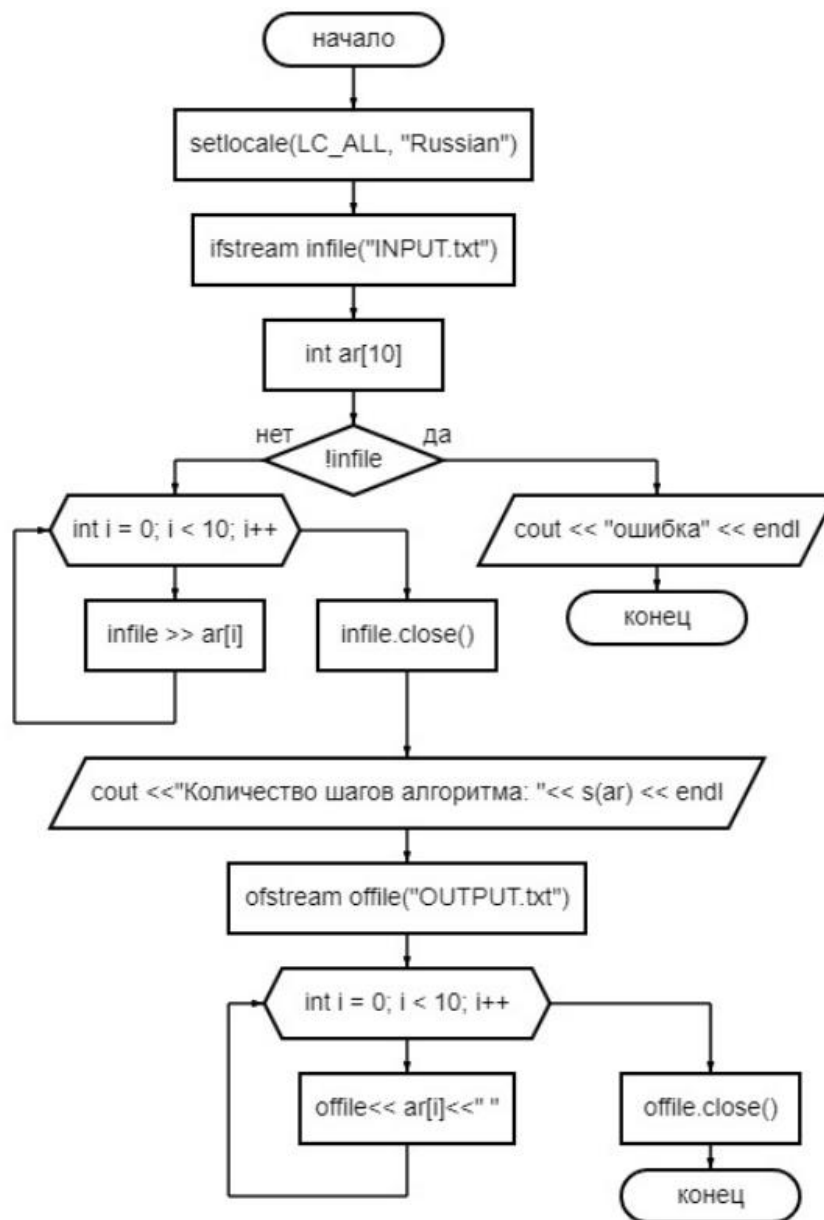
Преобразование одномерных массивов. Сортировка. Работа с файлами

Цель работы: Получить практический навык работы с текстовыми файлами, сортировки одномерных массивов.

В соответствии с вариантом задания разработать графический алгоритм решения задачи. Создать текстовый файл, заполнить его элементами массива. Написать программу на C++ для решения задачи.

Пример

Прочитать из файла одномерный массив из 10 элементов целого типа, произвести быструю сортировку по убыванию элементов. Результат вывести в отдельный файл. На консоль вывести количество шагов алгоритма.



```

#include <iostream>
#include <fstream>
using namespace std;
int s(int* ar) {
    int b = 0;

```

```

int j = 0;
for (int i = 1; i < 10; i++) {
    if (ar[i - 1] > ar[i]) {
        //b = ar[i - 1];
        //ar[i - 1] = ar[i];
        //ar[i] = b;
        swap(ar[i - 1], ar[i]); // поменять местами
        i = 0;
        j++;
    }
}
return j;
}
int main() {
    setlocale(LC_ALL, "Russian");
    ifstream infile("input.txt");

    int ar[10];
    if (!infile) {
        cout << "ошибка открытия файла" << endl;
        return 0;
    }
    for (int i = 0; i < 10; i++) {
        infile >> ar[i];
    }
    infile.close();
    cout << "Количество шагов алгоритма: " << s(ar) << endl;
    ofstream outfile("OUTPUT.txt");
    for (int i = 0; i < 10; i++) {
        outfile << ar[i] << " ";
    }
    outfile.close();
    return 0;
}

```

Варианты задания:

1. Прочитать из файла одномерный массив из 10 элементов вещественного типа, произвести быструю сортировку по возрастанию элементов. Результат вывести в отдельный файл. На консоль вывести количество шагов алгоритма
2. Прочитать из файла одномерный массив из 10 элементов целого типа, произвести быструю сортировку по убыванию элементов. Результат вывести в отдельный файл. На консоль вывести количество шагов алгоритма
3. Прочитать из файла одномерный массив из 20 элементов целого типа, произвести сортировку по возрастанию пузырьковым методом по убыванию элементов. Результат вывести в отдельный файл. На консоль вывести количество шагов алгоритма
4. Прочитать из файла одномерный массив из 20 элементов целого типа, произвести сортировку по убыванию пузырьковым методом. Результат вывести в отдельный файл. На консоль вывести количество шагов алгоритма

5. Прочитать из файла одномерный массив из 10 элементов целого типа, произвести сортировку Шелла по возрастанию элементов. Результат вывести в отдельный файл. На консоль вывести количество шагов алгоритма

Пример быстрой сортировки:

```
quickSort(numbers, 0, 9);

fstream file;

file.open("output.txt", fstream::in | fstream::out);

for (int i = 0; i < 10; i++) {

file << numbers[i] << endl;

}
```

Пример сортировки Шелла

```
void Shell(int A[], int n) //сортировка Шелла
{
d=n;
d=d/2;
while (d>0)
{
for (i=0; i<n-d; i++)
{
j=i;
while (j>=0 && A[j]>A[j+d])
{
count=A[j];
A[j]=A[j+d];
A[j+d]=count;
j--;
}}
d=d/2;
}
for (i=0; i<n; i++) cout<<A[i]<<" "; //вывод массива
}

Вызов функции сортировки в main:

int *A= new int[n]; //объявление динамического массива
Shell(A, n);
delete []
```

Практическая работа № 13 -14

Структуры данных. Работа с файлами.

Задание: Описать структуру с именем, содержащим следующие поля:

Название пункта назначения (строковый тип или массив)

Номер рейса (строковый или целый)

Время (строковый или массив)

Считать данные из файла в массив из 5 элементов.

Организовать вывод в консоль и в файл типа :

Пункт назначения:

Москва
 Рейс: 578
 Тип: 12:48

Пункт назначения: Ростов
 Рейс: 678
 Тип: 13:30

9) Создаем структуру с именем AEROFLOT

```

1  #include <iostream>
2  #include <fstream>
3  #include <stdlib.h>
4  #include <windows.h> //Вывод кириллицы в консоли
5  #include <string> //Работа со строками
6
7  using namespace std;
8
9  struct AEROFLOT
10 {
11     char Punkt[100];
12     string Reis;
13     char Type[100];
14 };
15
16 void IzFila(AEROFLOT* m, int n) //Считывает данные из файла
17 {
18     ifstream in("tik.txt");
19     if (!in)
20     {
21         cout << "Не могу открыть файл..." << endl;
22         system("pause");
23         exit(1);
24     }
25     for (int i = 0; i < n; i++)
26     {
27         in >> m[i].Punkt >> m[i].Reis >> m[i].Type;
28     }
29 }

```

16-29) Создаем функцию, которая будет считывать данные из файла tik.

21) Если файл не найден, выводим на экран «Не могу открыть файл»

25) Считываем данные n-количество раз в цикле. (по заданию 5 раз)

```

31 void Vivod(AEROFLOT* m, int n)//Выводит в консоль,заношит в файл
32 {
33     ofstream o("tok.txt");
34     cout << "Данные занесены в tok.txt" << endl;
35     for (int i = 0; i < n; i++)
36     {
37         cout << "Пункт назначения : " << m[i].Punkt << endl;
38         cout << "Рейс: " << m[i].Reis << endl;
39         cout << "Тип: " << m[i].Type << endl;
40         cout << "-----" << endl;
41         o << "Пункт назначения: " << m[i].Punkt << endl;
42         o << "Рейс: " << m[i].Reis << endl;
43         o << "Тип: " << m[i].Type << endl;
44         o << "-----" << endl;
45     }
46 }

```

- 31) Создаем функцию, которая будет вводить в консоль данные и вносить их в файл tok.
 33) Заносим данные в файл tok
 34) Выводим на экран, что данные занесены в файл
 35) Цикл для вывода данных. (заполняем структуру данными)

```

48 int main()
49 {
50     SetConsoleCP(1251);
51     SetConsoleOutputCP(1251);
52     int n;
53     cout << "Ввод N(по условию N=5) " << endl;
54     cin >> n;
55     AEROFLOT* m = new AEROFLOT[n];
56
57     IzFaila(m, n);
58     Vivod(m, n);
59
60     delete[] m;
61     system("pause");
62     return 0;
63 }

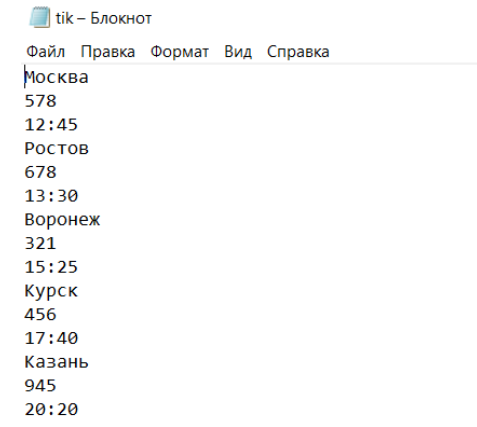
```

- 50-51) Для русского языка
 52) Объявляем переменную
 53) Вывод на экран
 54) Вводим n с клавиатуры
 55) Выделяем память под массив структур из n элементов
 57) Вызываем функцию IzFaila

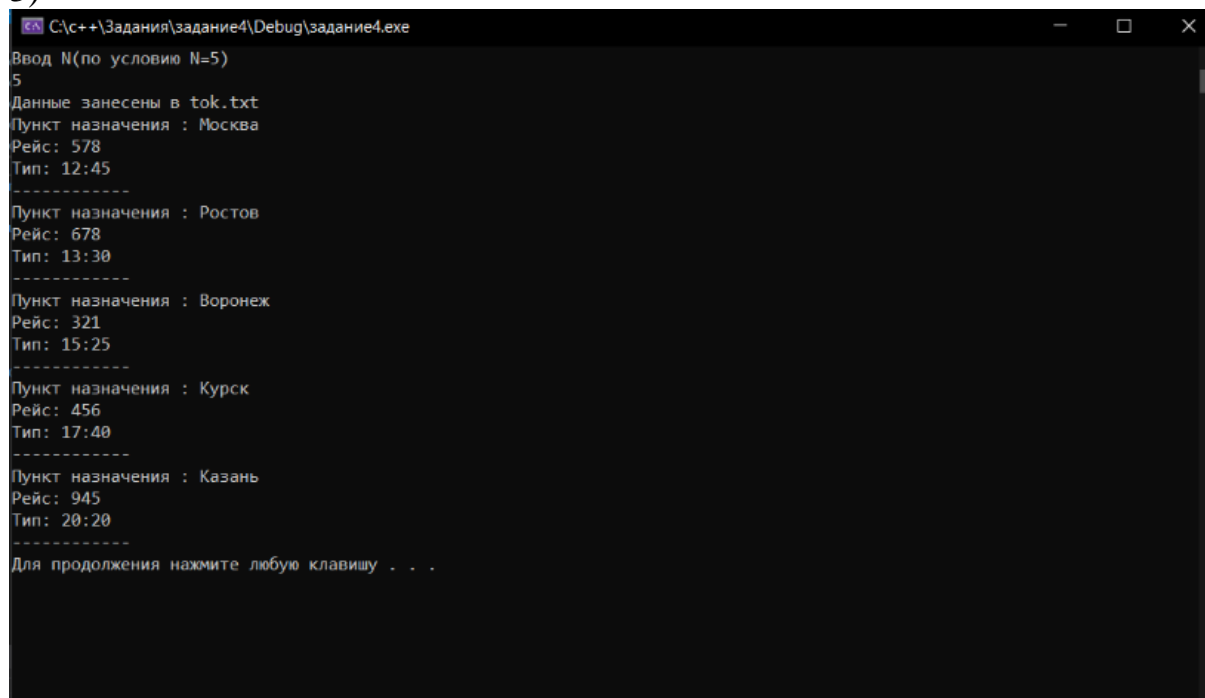
58) Вызываем функцию Vivod

60) очищаем память.

1) Файл с данными



3)



4) Заносим результат в файл.

tok – Блокнот

Файл Правка Формат Вид Справка

Пункт назначения: Москва
 Рейс: 578
 Тип: 12:45

 Пункт назначения: Ростов
 Рейс: 678
 Тип: 13:30

 Пункт назначения: Воронеж
 Рейс: 321
 Тип: 15:25

 Пункт назначения: Курск
 Рейс: 456
 Тип: 17:40

 Пункт назначения: Казань
 Рейс: 945
 Тип: 20:20

2. Практическое задание

Разработать алгоритм и программу по теме структуры и файлы

Варианты заданий:

<p>1. Опишите запись с именем типа <i>Karta</i>, содержащую следующие поля:</p> <ul style="list-style-type: none"> номер измерения (тип <i>int</i>); значение (тип <i>float</i>). <p>Переменную, определяющую запись, назовите <i>Z</i>.</p> <p>Считать данные из файла в массив из <i>n</i> элементов.</p> <p>Организовать вывод в консоль и в файл, указав названия полей.</p>	<p>5. Опишите запись с именем типа <i>Systema</i>, содержащую информацию о планетах солнечной системы:</p> <ul style="list-style-type: none"> номер планеты по удалению от Солнца (тип <i>int</i>); название планеты (тип <i>string</i>); диаметр (тип <i>float</i>); <p>Переменную определяющую запись назовите <i>Planeta</i>.</p> <p>Считать данные из файла в массив из <i>n</i> элементов.</p> <p>Организовать вывод в консоль и в файл, указав названия полей</p>
<p>2. Опишите запись с именем типа <i>Doc</i>, содержащую следующие поля:</p> <ul style="list-style-type: none"> номер строки документа (тип <i>int</i>); текст строки (тип <i>string</i>). <p>Переменную, определяющую запись, назовите <i>S</i>.</p> <p>Считать данные из файла в массив из <i>n</i> элементов.</p> <p>Организовать вывод в консоль и в файл, указав названия полей</p>	<p>7. Опишите запись с именем типа <i>Tovar</i>, содержащую информацию о хранящемся на складе товаре:</p> <ul style="list-style-type: none"> код товара (тип <i>int</i>); наименование товара (тип <i>string</i>); цену (тип <i>float</i>). <p>Переменную, определяющую запись, назовите <i>Tov</i>.</p> <p>Считать данные из файла в массив из <i>n</i> элементов.</p> <p>Организовать вывод в консоль и в файл, указав названия полей</p>

<p>3. Опишите запись с именем типа <i>Tovar</i>, содержащую информацию о хранящемся на складе товаре:</p> <ul style="list-style-type: none"> • код товара (тип <i>int</i>); • дата поступления (тип <i>string</i>); • наименование товара (тип <i>string</i>); • цену (тип <i>float</i>). <p>Переменную, определяющую запись, назовите <i>Tov</i>.</p> <p>Считать данные из файла в массив из <i>n</i> элементов.</p> <p>Организовать вывод в консоль и в файл, указав названия полей</p>	<p>8. Опишите запись с именем типа <i>Boln</i>, содержащую информацию больничной базы данных о стационарных больных.</p> <ul style="list-style-type: none"> • ФИО (тип <i>string</i>); • возраст (тип <i>int</i>); • адрес (тип <i>string</i>); • дата поступления (тип <i>string</i>); • диагноз (тип <i>string</i>); • ФИО лечащего врача (тип <i>string</i>). <p>Переменную определяющую запись назовите <i>B</i></p> <p>Считать данные из файла в массив из <i>n</i> элементов.</p> <p>Организовать вывод в консоль и в файл, указав названия полей</p>
<p>4. Опишите запись с именем типа <i>Baza</i>, содержащую информацию для студенческой базы данных:</p> <ul style="list-style-type: none"> • номер студенческого билета (тип <i>int</i>); • ФИО (тип <i>string</i>); • год рождения (тип <i>int</i>); • адрес (тип <i>string</i>). <p>Считать данные из файла в массив из <i>n</i> элементов.</p> <p>Организовать вывод в консоль и в файл, указав названия полей Поиск элементов по заданию преподавателя</p>	<p>9. Опишите запись с именем типа <i>Tovar</i>, содержащую информацию о хранящемся на складе товаре:</p> <ul style="list-style-type: none"> • код товара (тип <i>int</i>); • наименование товара (тип <i>string</i>); • цену (тип <i>float</i>). <p>Считать данные из файла в массив из <i>n</i> элементов.</p> <p>Организовать вывод в консоль и в файл, указав названия полей Поиск элементов по заданию преподавателя</p>
<p>5. Опишите запись с именем типа <i>Post</i>, содержащую информацию в почтовой базе данных о подписчиках на газеты и журналы:</p> <ul style="list-style-type: none"> • ФИО (тип <i>string</i>); • адрес (тип <i>string</i>); • название журнала (тип <i>string</i>). <p>Введите информацию для трех записей.</p> <p>Считать данные из файла в массив из <i>n</i> элементов.</p> <p>Организовать вывод в консоль и в файл, указав названия полей Поиск элементов по заданию преподавателя</p>	<p>10. Составьте программу , которая описывает массив записей жильцов дома, отображая в нем следующую информацию о каждом:</p> <ul style="list-style-type: none"> • номер квартиры; • ФИО; • Возраст. <p>Программа должна обеспечить ввод данных поиск самого молодого жильца.</p> <p>Создайте массив из записей (не менее 5 элементов), осуществите ввод, каждого элемента массива. Поиск элементов по заданию преподавателя</p>

