Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №8**

Дисциплина: основы алгоритмизации и программирования

Тема: “Блоковый ввод-вывод” в Си

Выполнил работу

студент группы РИС-20-1б

Сухина В.Д.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Цель работы**

Цель – работа с двоичными файлами, организация ввода-вывода структурированной информации и ее хранение на внешних носителях.

Постановка задачи

Сформировать двоичный файл из элементов, заданной в варианте структуры, распечатать его содержимое, выполнить удаление и добавление элементов в соответствии со своим вариантом, используя для поиска удаляемых или добавляемых элементов функцию. Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

1. Структура "Государство":

* название;
* столица;
* численность населения;
* занимаемая площадь.

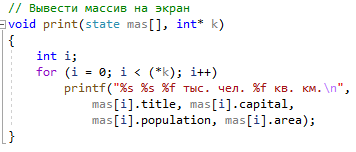
Удалить все элементы, у которых численность меньше заданной, добавить элемент после элемента с указанным номером.

Для выполнения данной работы использован язык программирования Си в программной среде Visual Studio 2019.

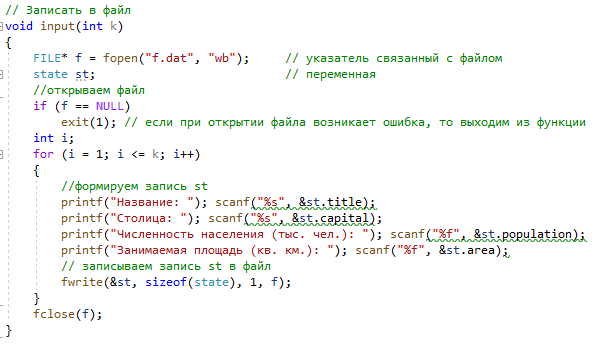
Анализ задачи

1. Определить какие операции должны быть выполнены по заданию:

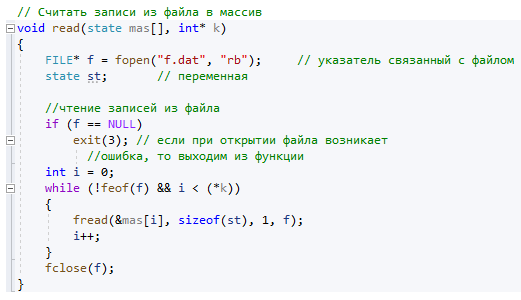
* Вывод данных на экран.



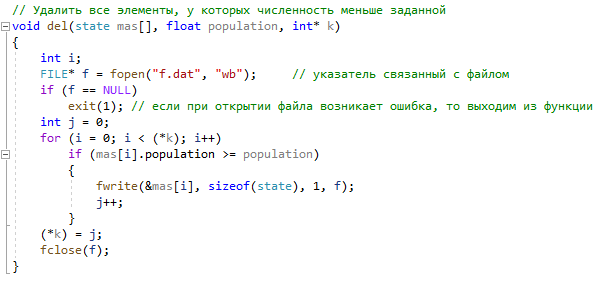
* Запись структур в двоичный файл.



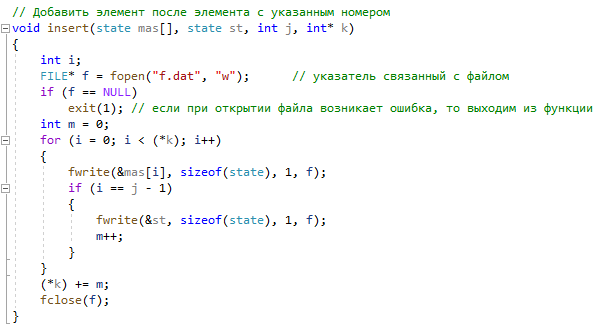
* Чтение структур из двоичного файла в массив.



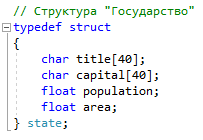
* Запись структур из массива в файл по условию (элементы, у которых численность больше либо равна заданной).



* Запись структур из массива в файл, а также запись заданной структуры после элемента с заданным номером.



1. С какого типа данными это можно сделать:



1. В каком виде эти данные будут представлены для решения задачи:

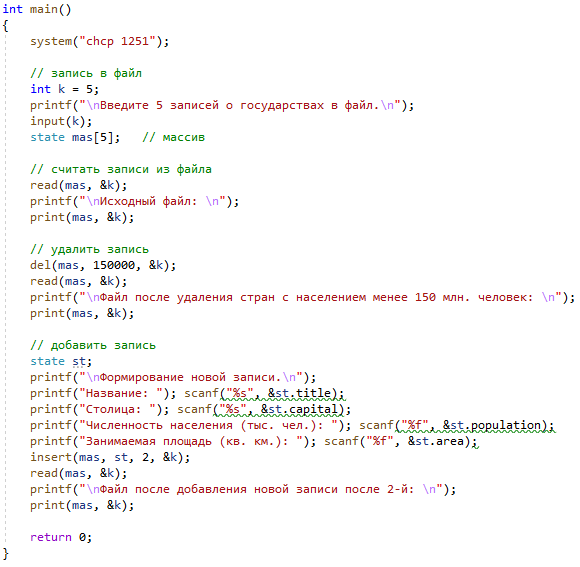
Из файла структуры считываются в массив state mas[].

Для чтения и записи в файл используется файловая переменная FILE\*.

1. Какими операторами будет реализован ввод и вывод данных:

Ввод-вывод на экран осуществляется при помощи операторов scanf и printf соответственно. Для записи в файл используется оператор fwrite, для чтения из файла – fread.

Выполнение действий (вызов соответствующих подпрограмм) осуществляется в главной программе.



Код

#pragma warning(disable:4996)

#include <stdlib.h>

#include <string.h>

#include <locale.h>

#include <stdio.h>

// Структура "Государство"

typedef struct

{

char title[40];

char capital[40];

float population;

float area;

} state;

// Записать в файл

void input(int k)

{

FILE\* f = fopen("f.dat", "wb"); // указатель связанный с файлом

state st; // переменная

//открываем файл

if (f == NULL)

exit(1); // если при открытии файла возникает ошибка, то выходим из функции

int i;

for (i = 1; i <= k; i++)

{

//формируем запись st

printf("Название: "); scanf("%s", &st.title);

printf("Столица: "); scanf("%s", &st.capital);

printf("Численность населения (тыс. чел.): "); scanf("%f", &st.population);

printf("Занимаемая площадь (кв. км.): "); scanf("%f", &st.area);

// записываем запись st в файл

fwrite(&st, sizeof(state), 1, f);

}

fclose(f);

}

// Считать записи из файла в массив

void read(state mas[], int\* k)

{

FILE\* f = fopen("f.dat", "rb"); // указатель связанный с файлом

state st; // переменная

//чтение записей из файла

if (f == NULL)

exit(3); // если при открытии файла возникает

//ошибка, то выходим из функции

int i = 0;

while (!feof(f) && i < (\*k))

{

fread(&mas[i], sizeof(st), 1, f);

i++;

}

fclose(f);

}

// Удалить все элементы, у которых численность меньше заданной

void del(state mas[], float population, int\* k)

{

int i;

FILE\* f = fopen("f.dat", "wb"); // указатель связанный с файлом

if (f == NULL)

exit(1); // если при открытии файла возникает ошибка, то выходим из функции

int j = 0;

for (i = 0; i < (\*k); i++)

if (mas[i].population >= population)

{

fwrite(&mas[i], sizeof(state), 1, f);

j++;

}

(\*k) = j;

fclose(f);

}

// Добавить элемент после элемента с указанным номером

void insert(state mas[], state st, int j, int\* k)

{

int i;

FILE\* f = fopen("f.dat", "w"); // указатель связанный с файлом

if (f == NULL)

exit(1); // если при открытии файла возникает ошибка, то выходим из функции

int m = 0;

for (i = 0; i < (\*k); i++)

{

fwrite(&mas[i], sizeof(state), 1, f);

if (i == j - 1)

{

fwrite(&st, sizeof(state), 1, f);

m++;

}

}

(\*k) += m;

fclose(f);

}

// Вывести массив на экран

void print(state mas[], int\* k)

{

int i;

for (i = 0; i < (\*k); i++)

printf("%s %s %f тыс. чел. %f кв. км.\n",

mas[i].title, mas[i].capital,

mas[i].population, mas[i].area);

}

int main()

{

system("chcp 1251");

// запись в файл

int k = 5;

printf("\nВведите 5 записей о государствах в файл.\n");

input(k);

state mas[5]; // массив

// считать записи из файла

read(mas, &k);

printf("\nИсходный файл: \n");

print(mas, &k);

// удалить запись

del(mas, 150000, &k);

read(mas, &k);

printf("\nФайл после удаления стран с населением менее 150 млн. человек: \n");

print(mas, &k);

// добавить запись

state st;

printf("\nФормирование новой записи.\n");

printf("Название: "); scanf("%s", &st.title);

printf("Столица: "); scanf("%s", &st.capital);

printf("Численность населения (тыс. чел.): "); scanf("%f", &st.population);

printf("Занимаемая площадь (кв. км.): "); scanf("%f", &st.area);

insert(mas, st, 2, &k);

read(mas, &k);

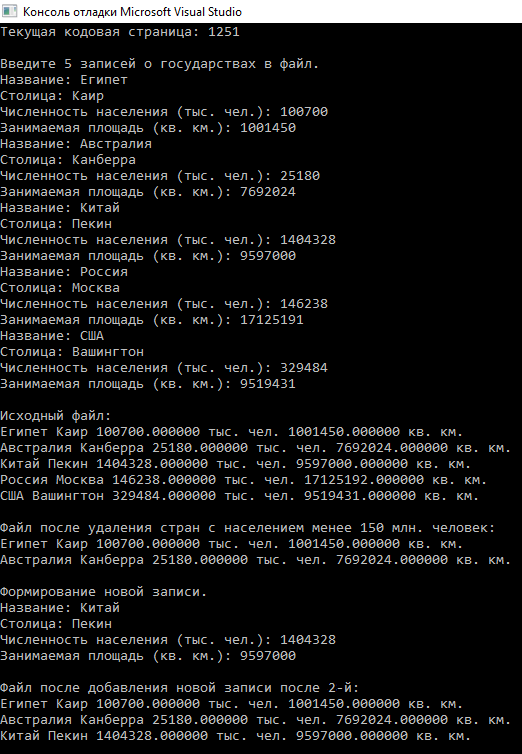
printf("\nФайл после добавления новой записи после 2-й: \n");

print(mas, &k);

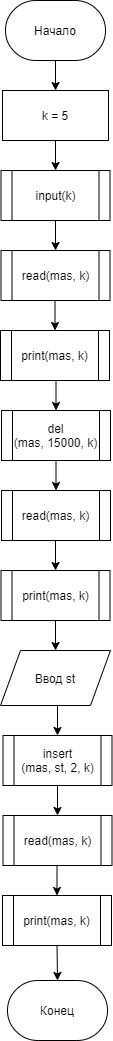
return 0;

}

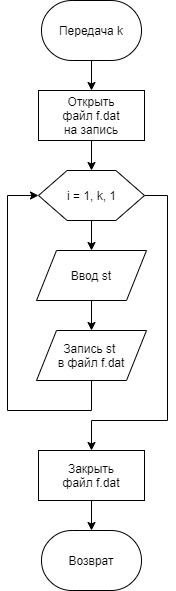
Скриншот выполнения программы



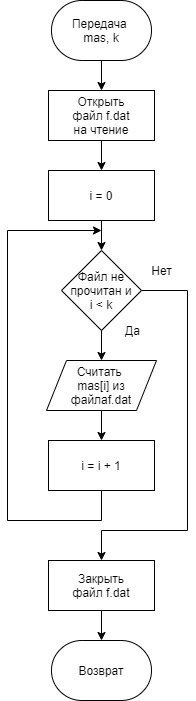
Блок-схема главной программы



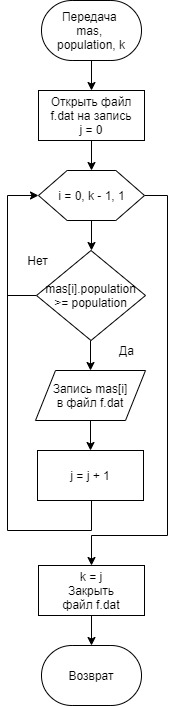
Блок-схема функции input



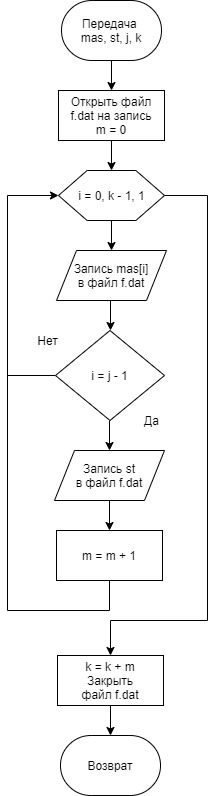
Блок-схема функции read



Блок-схема функции del



Блок-схема функции insert



Блок-схема функции print

