МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСІТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТІТУТ»

КАФЕДРА МЕДИЧНОЇ КІБЕРНЕТИКИ ТА ТЕЛЕМЕДИЦИНИ

**Лабораторна робота № 14**

з дисципліни «Алгоритмізація та програмування»

на тему: “ Бінарні файли даних ”

Варіант 16

**Виконав:**

Студент групи БС-32

Хоміцевич Микита

**Перевірила:**

викладач

Матвійчук А.О.

Київ-2014

**Задание**



**□ Лабораторна робота виконана без зауважень**

**□ Лабораторна робота має зауваження:**

**□ присутні зауваження до блок-схем:**

**□ виконані не за стандартом**

**□ не відповідають коду**

**□ присутній код**

**□ інші зауваження:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**□присутні зауваження до коду:**

**□задача завдання вирішена хибно**

**□ код програми не компілюється**

**□ використано глобальні змінні**

**□ недостатня декомпозиція на функціонуванні користувача**

**□ статичні змінні при роботі з масивами**

**□ оформлення коду**

**□присутні зайві символи «{»та «}»**

**□інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**□невірні відповіді на запитання:**

**□№1 □№2 □№3**

**□маються інші зауваження: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Программный код**

#include <iostream>

#include <cstddef>

#include <conio.h>

#include <fstream>

#include <cstdio>

#include <cstdlib>

#include <ctime>

using namespace std;

//////////////////////MAIN FUNCTIONS PROTOTYPES////////////////////

void MAIN();

void bool\_mas\_input\_zeros(int \*, int n);

void rand\_input\_in\_file(FILE \*, int \*, int);

void write\_same\_in\_file(FILE \*, FILE \*, int \*, int, int \*);

void output\_same\_from\_file(FILE \*, int \*, int);

//////////////////////HELP FUNCTIONS PROTOTYPES////////////////////

void output\_file(FILE \*, int \*, int);

void output\_arr(int \*X, int);

///////////////////////////PROGRAMS BODY/////////////////////////

int main() {

MAIN();

\_getch();

}

void MAIN() {

setlocale(LC\_ALL, "RUS");

srand(time(NULL));

FILE \*F, \*F1;

fopen\_s(&F, "D:\\F.dat", "wb");

int n, k=0;

cout << "¬ведите кол-во элементов в файле F: ";

cin >> n;

int \*bool\_mas = new int[n];

// input zeros in arr

bool\_mas\_input\_zeros(bool\_mas, n);

// random input in file

rand\_input\_in\_file(F, bool\_mas, n);

fopen\_s(&F1, "D:\\F1.dat", "wb");

// запись одинаковых эл-ов в файл

write\_same\_in\_file(F, F1, bool\_mas, n, &k);

// закрытие файлов ///

fclose(F);

fclose(F1);

fopen\_s(&F1, "D:\\F1.dat", "rb");

int \*a = new int[k];

output\_same\_from\_file(F1, a, k);

fclose(F);

fclose(F1);

}

////////////////////===================FUNCTIONAL=====================////////////

void bool\_mas\_input\_zeros(int \*bool\_mas, int n) {

for (int i = 0; i < n; ++i)

bool\_mas[i] = 0;

}

void rand\_input\_in\_file(FILE \*F, int \*bool\_mas, int n) {

int x;

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

x = rand() % n;

fwrite(&x, sizeof(int), 1, F);

bool\_mas[x]++;

}

}

void write\_same\_in\_file(FILE \*F, FILE \*F1, int \*bool\_mas, int n, int \*k) {

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

if (bool\_mas[i] > 1) {

fwrite(&i, sizeof(int), 1, F1);

(\*k)++;

}

}

cout << endl;

}

void output\_same\_from\_file(FILE \*F1, int \*a, int k) {

fread(a, sizeof(int), k, F1);

cout << "Ёлементы в файле, которые повтор¤ютс¤: " << endl;

for (int i = 0; i < k; ++i)

cout << a[i] << " ";

cout << endl;

}

////////////////////==================DEBUGGING===================///////////////

void output\_arr(int \*X, int n) {

for (int i = 0; i < n; ++i)

cout << X[i] << " ";

cout << endl;

}

void output\_file(FILE \*F, int\* q, int n) {

fread(q, sizeof(int), n, F);

cout << "Ёлементы файла F: " << endl;

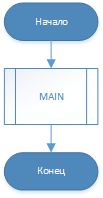
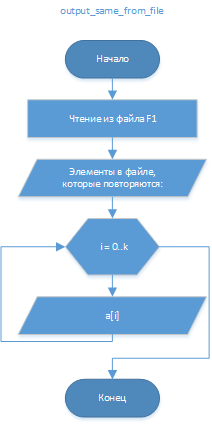
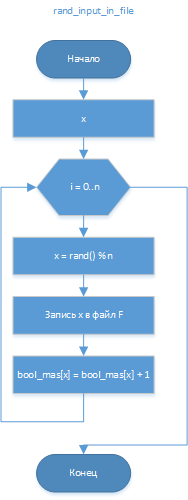
for (int i = 0; i < n; ++i)

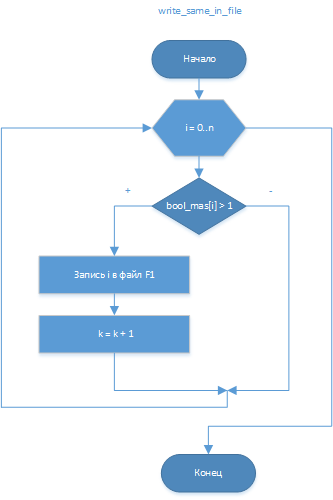
cout << q[i] << " ";

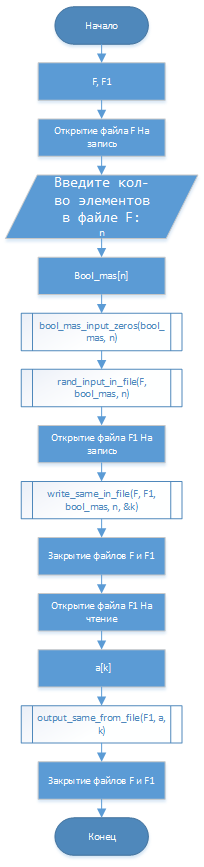
cout << endl;

}

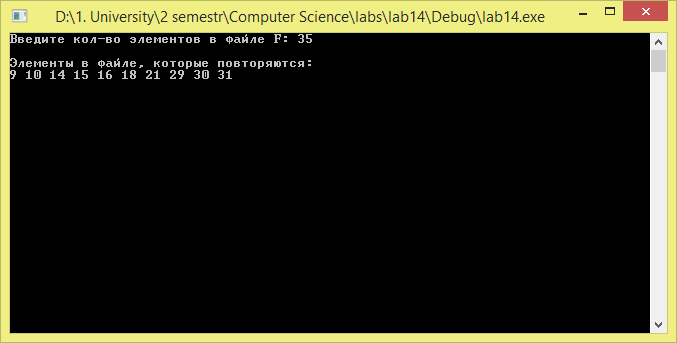
**Блок-схема**

****

****

****

**Скриншот работы программы**

****

**Ответы на контрольные вопросы**

1. Что такое файл?

Файл – это упорядоченная последовательность однотипных компонентов, размещенных на внешнем носителе.

1. Чем отличается файл от массива?

В массиве можно хрнанить, доставать и изменять информацию, в отличии от файлов, где можно лишь хранить информацию, а обработкой этой информации занимаются другие программы.

1. Что подразумевается под чтением и под записью в файл?

Это означает, что при чтении\записи мы постоянно работает не только с памятью файла, а и с оперативной паматью, которая заполняется полученными данными из файла.

1. Типы доступа к файлам

Есть два типа доступа к файлам:

* Текстовый – (имеет собсвтенную структуру, может быть открыт любым текстовым редактором и изменем)
* Бинарный (не имеют собственной структуры, каждая программа задает свою структур)

1. При каких режимах открытия файла его содержимое сохраняется?

Только при двух режимах: “a” и “a+”.

1. Что такое буфер и зачем он нужен?

Буфер – это выделенное промежуточное хранилище данных, которое предоставляется различным ПО, и предназначается для переноса или копирования информации для более быстрой работы с файлами а так же для более удобного доступа к самой информации.

1. Когда меняется содержание буфера?

Когда буфер заполняется.

1. Почему при чтении текстового файла передаются не все его байты?
2. Что возвращает ф-ия fopen() ?

Функция fopen() возвращает указатель на файловый, который мы открыли. В случае ошибки fopen() возвращает NULL

1. Почему ф-ии записи и чтения символов fgetc и fputc возвращают целое?

Функция fputc() возвращает значение int, т.к. возвращаемое значение – это номер позиции курсора.

Функция fgetc() возвращает значение int, т.к. возвращ. Значение – это номер символа в ASCII таблице.

1. Зачем закрывать файл?

Для того, что бы освободить память от данных, что лежат в буфере + что бы не было утечки, т.е. доступа к файлу после выполнения и закрытия программы.