 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Лабораторна робота №7**

з дисципліни «Алгоритмізація та програмування

на тему: «Бінарні файли даних»

Варіант №

**Виконав:**

**Перевірив:**

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

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

(підпис викладача)

Київ-2022

**Завдання**

1. Ознайомитися з теоретичними відомостями щодо роботи з файловими

структурами даних.

2. Побудувати блок-схему алгоритму для вирішення задачі відповідно до

свого варіанту.

3. Відповідно до свого варіанту розробити програмний застосунок, що включає вивід на екран початкових цілочислових даних, їх обробку та

збереження результатів у бінарному файлі / файлах з подальшим виводом

на екран їх змісту (імена файлів задати в коді програми, глобальні змінні

не використовувати, функція main має бути призначена тільки для виклику функцій користувача):

**Завдання за варіантом:** Компонентами бінарного файлу F є цілі числа. Записати у новий бінарний файл F1 всі числа файлу F, що кратні 5, а в новий бінарний файл F2 – всі від’ємні числа файлу F, що кратні 3.

4. Додаткове завдання\*. Розширити код програми виконання основного завдання за рахунок введення додаткових можливостей роботи з файлом / файлами на диску. Назва файлу / файлів для обробки має задаватися користувачем з клавіатури перед початком виконання завдання. Організувати перевірку наявності заданого користувачем файлу / файлів та в разі його присутності надати користувачу можливість або перейменувати старий файл та створити при цьому новий із заданою раніше назвою, або видалити старий файл з диску, після чого створити новий із заданою раніше назвою (якщо завдання потребує наявності в початковому файлі / файлах даних, то необхідно надати користувачу можливість заповнити новий файл під час його створення). Побудувати блок-схему алгоритму вирішення додаткового завдання.

5. Скласти і захистити звіт по роботі.

**Програмна реалізація:**

#include <iostream>

#include <cstdio>

#include <io.h>

#pragma warning( disable : 4996 )

#define \_CRT\_SECURE\_NO\_WARNINGS

void getFilename(char[]);

int getMainFilename(char[]);

void work(char[], char[], char[], int);

void printElems(char[]);

using namespace std;

int main() {

cout << "F";

char f[256];

int a = getMainFilename(f);

cout << "F1";

char f1[256];

getFilename(f1);

cout << "F2";

char f2[256];

getFilename(f2);

work(f, f1, f2, a);

printElems(f);

printElems(f1);

printElems(f2);

return 0;

}

int getMainFilename(char f[]){

char name[256];

cout << " enter filename of: ";

cin >> name;

int n, i, a=0;

if (!access(name, 0)){

cout << "Do you want to read data, or to enter from keyboard?(r/e): ";

char ch;

cin >> ch;

if (ch == 'r') {

FILE\* ff;

ff = fopen(name, "rb");

a = fseek(ff, 1, SEEK\_END);

a = (int)ftell(ff) / 4;

fclose(ff);

strcpy(f, name);

}

else {

FILE\* ff;

ff = fopen(name, "wb");

cout << "How much numbers to enter? ";

cin >> a;

for (i = 0; i < a; i++){

cout << "n = ";

cin >> n;

fwrite(&n, sizeof(int), 1, ff);

}

strcpy(f, name);

fclose(ff);

}

}

else {

FILE\* ff;

ff = fopen(name, "wb");

cout << "There is no file with this name\nHow much numbers to enter? ";

cin >> a;

for (i = 0; i < a; i++){

cout << "n = ";

cin >> n;

fwrite(&n, sizeof(int), 1, ff);

}

strcpy(f, name);

fclose(ff);

}

return a;

}

void getFilename(char f[]){

char name[256];

cout << " enter filename: ";

cin >> name;

if (!access(name, 0)) {

cout << "Do you want to rename or delete existing file?(r/d): ";

char ch;

cin >> ch;

if (ch == 'r') {

char newName[256];

cout << "Enter a new filename: ";

cin >> newName;

rename(name, newName);

}

}

FILE\* file = fopen(name, "wb");

strcpy(f, name);

fclose(file);

}

void work(char f[], char f1[], char f2[], int a) {

int\* arr = new int[a];

FILE\* fileF = fopen(f, "rb");

FILE\* file1F = fopen(f1, "wb");

FILE\* file2F = fopen(f2, "wb");

fread(arr, sizeof(int), a, fileF);

fclose(fileF);

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

if (arr[i] % 5==0) {

fwrite(&arr[i], sizeof(int), 1, file1F);

}

if ((arr[i] % 3==0) && (arr[i] < 0)) {

fwrite(&arr[i], sizeof(int), 1, file2F);

}

}

fclose(fileF);

fclose(file1F);

fclose(file2F);

}

void printElems(char f[]) {

int a, d;

FILE\* ff = fopen(f, "rb");

fseek(ff, 0, SEEK\_END);

a = (int)ftell(ff) / 4;

fseek(ff, 0, SEEK\_SET);

cout << "Elements of file " << f << endl;

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

fread(&d, sizeof(int), 1, ff);

cout << d << " ";

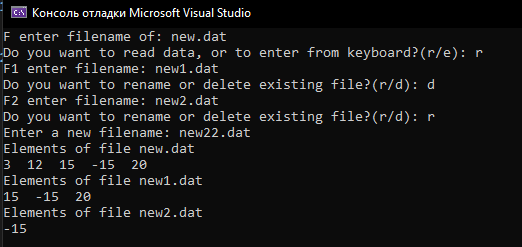
}

cout << endl;

fclose(ff);

}

**Результат виконання програми:**

****

**Контрольні питання:**

1. Що таке файл? Для яких цілей використовують файли?

Файл - це іменований набір байтів, який може бути збережений на ;

деякому накопичувані. Тобто файлом є впорядкована послідовність однотипних компонентів, розташованих на зовнішньому носії. Для управління безліччю файлів до складу операційних систем входять файлові системи, які визначають, яким чином файли іменовані і де вони розміщені.

Використання файлів доцільно у разі:

- довготривалого зберігання даних;

- доступу різних програм до одних і тих же даних;

- обробки великих масивів даних, які неможливо цілком розмістити в

оперативній пам’яті комп’ютера.

1. Чому не доцільно працювати напряму з зовнішніми носіями замість оперативної пам'яті?

Робота з інформацією в оперативній пам’яті набагато швидша за роботу з зовнішнім носієм.

1. Що спільного та в чому різниція між текстовими та двійковими файлами?

Двійковий файл — в широкому сенсі: файл, що містить послідовність довільних байтів. Назва пов'язана з тим, що байти складаються з біт, тобто двійкових цифр.

Текстовий файл — форма подання послідовності символів у комп'ютері, де кожен символ із задіяного набору символів кодується одним байтом чи послідовністю двох, трьох і т. д. байтів.

У вузькому сенсі слова двійкові файли протиставляються текстовим файлам. При цьому, з точки зору технічної реалізації на рівні апаратури, текстові файли є окремим випадком двійкових файлів, і, таким чином, в широкому значенні слова під визначення двійковий файл підходить будь-який файл.

1. Як буде виглядати нетекстовий файл при відкритті у текстовому редактору? Пояснить, чому.

Нетекстовий файл у текстовому редакторі буде виглядати як набір певних символів. Це відбувається, тому що нетекстовий файл являє собою набір байтів у певній послідовності, які текстовий редактор інтерпретує, як символи, які також являються одним, або набором байтів.

1. Як найчастіше наочно представляють двійкові файли для перегляду людиною?

Для візуального перегляду вмісту двійкового файлу він розбивається на шматки рівного розміру, що подаються у вигляді чисел, записуваних, зазвичай, в шістнадцятковій системі, іноді в вісімковій, двійковій або десятковій.

1. Що являють собою процеси читання з файлу та запису у файл? Як визначається місце в файлі, з якого буде читатися або в яке буде записуватися наступна порція даних?

Файл може бути відкритий для читання, для запису або для читання і

запису одночасно. Операція запису (виведення) даних у файл означає

пересилання їх з робочої області пам’яті у файл, а операція читання (введення) з файлу - заповнення пам’яті даними, що отримані з файлу.

Місце, з якого буде читатися, або в яке буде записуватися інформація визначається за міткою-вказівником поточної позиції, яка оновлюється після кожної операції читання/введення.

1. Що таке буфер вводу/виводу і навіщо він потрібен? Коли змінюється зміст буфера?

Буфер – область оперативної пам’яті, яка використовується для тимчасового зберігання даних при введені/виведенні.

Використання буфера при обмінах з зовнішніми носіями інформації

підвищує швидкість передачі даних, оскільки реальні пересилання

здійснюються тільки тоді, коли буфер вже заповнений або порожній: дані заносяться в буфер перед тим, як вони будуть передані до зовнішнього носія інформації (під час виведення даних) або перед тим, як вони будуть передані в область пам’яті виконуваної програми (під час введення даних).

1. Опишіть режими відкриття файлу. При відкритті в яких режимах його зміст зберігається?

* «r» - відкрити існуючий файл для введення в нього (читання). Зміст зберігається
* «w» - створити новий файл або відкрити (і звільнити від попереднього вмісту) існуючий файл для виводу (запису) у його початок. Зміст не зберігається
* «a» - створити новий файл для виводу або відкрити існуючий файл для виводу в кінець файлу. Зміст зберігається
* «r+» - відкрити існуючий файл для оновлення (тобто читання й запису), яке буде здійснюватись з його початку. Зміст зберігається
* «w+» - створити новий файл або відкрити існуючий файл для оновлення, яке буде здійснюватись з його початку. Зміст зберігається
* «a+» - створити новий файл або відкрити існуючий файл і підстроїтись у його кінець для оновлення. Зміст зберігається

1. Який результат повертає функція fopen()? Для чого він потрібен?

Якщо файл успішно відкрито, функція повертає вказівник на об’єкт FILE, який можна використовувати для ідентифікації потоку під час майбутніх операцій. В іншому випадку повертається нульовий покажчик. Вказівник на об’єкт потрібен для подальшої роботи з файлом, а саме – читанням та записуванням інформації. Вказівник показує місце, з якого зчитати, або в яке вписати інформацію.

1. Для чого необхідно закривати файли після завершення роботи з ними?

Закриття файлу необхідно через наступні причини:

* кількість одночасно відкритих файлів обмежена можливостями операційної системи, необхідно зайві файли вчасно закривати;
* відкриті файли займають ресурси комп’ютера, які треба вивільнити після завершення роботи;
* через буферизацію даних, частина інформації може бути недозаписана у файл, тому, щоб дозаписати ці дані, треба закрити файл.