 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №14**

з дисципліни «Алгоритмізація та програмування»

на тему: «Бінарні файли даних»

Варіант №17

**Виконав:**

студент гр. БС-81

Сєров О. В.

**Перевірив:**

доцент каф. БМК

к.т.н. Алхімова С.М.

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

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

(підпис викладача)

Київ-2019

**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):

Компонентами бінарного файлу F1 є цілі числа. Записати в файл F2 спочатку від’ємні числа файлу F1, потім всі нулі, а потім всі додатні числа.

1. Скласти і захистити звіт по роботі.

**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

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

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

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

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

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

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

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

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

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

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

**Блок-схема:**





**Код:**

#include <iostream>

#include <conio.h>

using namespace std;

#define num 64

void console\_output(FILE\* DATA)

{

cout << "Data from file:\n";

int val = 0;

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

{

fscanf\_s(DATA, "%i", &val);

cout << val << ' ';

}

cout << endl;

fseek(DATA, 0, SEEK\_SET);

}

void negative\_zeros\_positive(void)

{

FILE\* F1; fopen\_s(&F1, "D:\\data.txt", "rb");

console\_output(F1);

FILE\* F2; fopen\_s(&F2, "D:\\out.txt", "w");

int var = 0; unsigned short zeros = 0;

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

{

fscanf\_s(F1, "%i", &var);

if (var < 0)

fprintf(F2, "%i ", var);

else

if (var == 0)

zeros++;

}

fseek(F1, 0, SEEK\_SET);

for (;zeros > 0; zeros--)

fprintf(F2, "%i ", 0);

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

{

fscanf\_s(F1, "%i", &var);

if (var > 0)

fprintf(F2, "%i ", var);

}

fclose(F1);

fclose(F2);

fopen\_s(&F2, "D:\\out.txt", "r");

console\_output(F1);

fclose(F2);

}

void main()

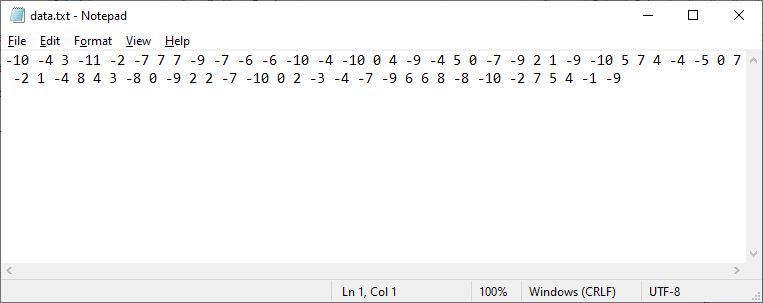
{

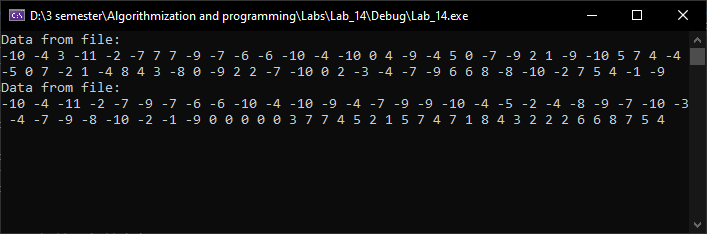
negative\_zeros\_positive();

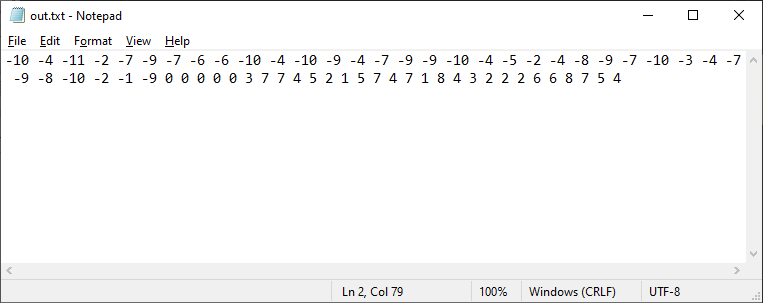
\_getch();

}

**Результати:**

****

****

****

**Контрольні питання:**

1. ***Що таке файл?***

Файл - це іменований набір байтів, який може бути збережений на деякому накопичувані. Тобто файлом є впорядкована послідовність однотипних компонентів, розташованих на зовнішньому носії.

1. ***Чим відрізняється файл від масиву?***

Масив зберігає однотипні дані, у той час як файл – різні за типом.

1. ***Що розуміється під читанням і під записом у файл?***

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.

1. ***При яких режимах відкриття файлу його вміст зберігається?***

При “r”, “a”, “r+”, “a+”, “rt”, “at”, “r+t”, “a+t”, “rb”, “ab”, “r+b”, “a+b”

1. ***Що таке буфер і навіщо він потрібен?***

Буфер обміну (англ. clipboard) — це частина оперативної пам’яті комп’ютера, яка спеціально призначена для зберігання тимчасової інформації. Застосунок може використати свій власний буфер обміну, доступний тільки в ньому, або загальний, надаваний операційною системою або іншим середовищем через певний інтерфейс.

1. ***Коли змінюється вміст буфера?***

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. ***Чому при читанні текстового файлу передаються не всі його байти?***

Якщо програма буде зчитувати інформацію байт за байтом, то після кожного читання буде заново виконуватися встановлення магнітних головок в нове положення. У результаті робота з файлом сповільниться. Щоб такого не відбувалося, файл треба читати і записувати як можна більш великими частинами (блоками). Найіфаще, коли розмір такого блоку кратний розміру блоку на носії інформації.

1. ***Що повертає функція fopen?***

Якщо функції fopen() вдалося відкрити вказаний файл, повертається покажчик на FILE. Якщо ж файл не може бути відкритий, повертається NULL.

1. ***Чому функції запису і читання символів fgetc i fputc повертають ціле?***

Функція fputc () повертає значення записаного символу. У разі помилки вона повертається EOF.

Функція fgetc () повертає наступний за поточною позицією символ з вхідного потоку і дає приріст вказівником положення в файлі.

Ці функції повертають ціле, бо один з варіантів повернення – EOF – дійсне ціле число.

1. ***Навіщо закривати файл?***

Якщо stream – це об'єкт fstream / ifstrea / etc, який можна не закривати, воно закриється в деструкції. для чого це треба? На кожний відкритий файл ОС виділяє дескриптор, за яким ідентифікується цей файл в програмі і всі операції в програмі з цим дескриптором виробляють відповідні дії з файлом.

Дескриптор – це просто число, закриття файлу має на увазі завершення всіх операцій з файлом і повернення дескриптора назад ОС, щоб та в свою чергу могла видати його при наступному відкритті файлу.