**­**

**Лабораторна робота №2.1**

З дисципліни «Програмування»

Тема: Файли.

Виконав: Землянський Едуард

Група: КВ-22

**Постановка задачі**

В загальному випадку програма повинна складатися з трьох логічних блоків:

1. Послідовний доступ до елементів бінарного файла;
2. Прямий доступ до елементі бінарного файла;
3. Робота з текстовим файлом.

***Блок 1***

1. Сформувати бінарний файл f, компоненти якого обчислюються за формулою згідно варіанта.
2. Вказані по варіантах дії над компонентами файла (файлів) виконати в рамках функції, використовуючи послідовний підхід.
3. Вивести на екран значення елементів всіх сформованих в завданні файлів та результуючих файлів.

#### *Блок 2*

1. Сформувати файл (чи файли, якщо їх декілька за умовою), тип, кількість й взаємне розташування компонентів якого повинні відповідати умові завдання за варіантом.
2. Використовуючи прямий доступ до компонентів файлів, виконати завдання за варіантом.

***Блок 3***

1. За допомогою текстового редактора підготувати текстовий файл, який відповідає умові варіанта.
2. Виконати дії, зазначені в умові варіанта.

**Вимоги до програми**

***Блок 1***

1. Обов’язковою є перевірка вхідного файла на порожність.
2. Використання додаткових файлів та масивів не допускається.

***Блок 2***

1. Сформувати файл (чи файли, якщо їх декілька за умовою), тип, кількість й взаємне розташування компонентів якого повинні відповідати умові завдання за варіантом.
2. Використовуючи прямий доступ до компонентів файлів, виконати завдання за варіантом.

***Блок 3***

1. Використовувати додаткові файли тільки в разі необхідності.
2. Обов’язковою є перевірка вхідного файла на порожність.

Словами називаються групи символів, відокремлені пробілами (одним чи кількома), і які не містять пробілів всередині.

**Варіант завдання**

|  |  |
| --- | --- |
| 12 | Сформувати файли f і g. Формулу для формування файла g взяти із наступного варіанта. Записати в файл h спочатку компоненти файла f, а потім - компоненти файла g із збереженням порядку слідування. |
| 12 | Сформувати файл невпорядкованих чисел. Відсортувати компоненти файла за неспаданням методом вибору. |
| Дано текстовий файл f, що містить програму мовою С. Перевірити цю програму на невідповідність відкриваючих та закриваючих круглих дужок. Вважати, що кожний оператор програми займає не більше одного рядка файла f. |

**Текст програми**

#include <conio.h>

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <time.h>

#include <string.h>

#define ERROR\_FILE\_OPEN -3

void block1(){

    FILE \* f;

    FILE \* g;

    FILE \* h;

    int i = 10;

    float value;

    f = fopen("f.bin", "wb");

    if (f == NULL){

        printf("\nFile opening failed!\n");

        exit(ERROR\_FILE\_OPEN);

    }

    printf("\nValues in 'f.bin':\n");

    for (float k = 1; k <= i; k++){

        value = 1 / (k \* (k + 1) \* (k + 2));

        printf("%f ", value);

        fwrite(&value, sizeof(float), 1, f);

    }

    fclose(f);

    g = fopen("g.bin", "wb");

    if (g == NULL){

        printf("\nFile opening failed!\n");

        exit(ERROR\_FILE\_OPEN);

    }

    printf("\n\nValue in 'g.bin':\n");

    value = (i - 0.1) / (pow(i, 3) + abs(tan(2\*i)));

    fwrite(&value, sizeof(float), 1, g);

    fwrite(&value, sizeof(float), 1, g);

    fwrite(&value, sizeof(float), 1, g);

    fclose(g);

    f = fopen("f.bin", "rb");

    h = fopen("h.bin", "wb");

    if (h == NULL){

        printf("\nFile opening failed!\n");

        exit(ERROR\_FILE\_OPEN);

    }

    while(fread(&value, sizeof(float), 1, f) != 0){

        fwrite(&value, sizeof(float), 1, h);

    }

    fclose(f);

    fclose(h);

    // fwrite(&value, sizeof(float), 1, h);

    g = fopen("g.bin", "rb");

    h = fopen("h.bin", "ab");

    while (fread(&value, sizeof(float), 1, g) != 0)

    {

        printf("%f ", value);

        fwrite(&value, sizeof(float), 1, h);

    }

    fclose(g);

    fclose(h);

    printf("\n\nValues in 'h.bin': \n");

    h = fopen("h.bin", "rb");

    while(fread(&value, sizeof(float), 1, h) != 0){

        printf("%f ", value);

    }

    fclose(h);

}

void take(FILE \* file, int position, int \* value)

{

    fseek(file, position \* sizeof(int), SEEK\_SET);

    fread(value, sizeof(int), 1, file);

}

void swap(FILE \* file, int position1, int \* value1, int position2, int \* value2)

{

    fseek(file, position1 \* sizeof(int), SEEK\_SET);

    fwrite(value2, sizeof(int), 1, file);

    fseek(file, position2 \* sizeof(int), SEEK\_SET);

    fwrite(value1, sizeof(int), 1, file);

}

void selectionSort(FILE \* file, int size)

{

    int min\_idx, value1, value2;

    for (int i = 0; i < size - 1; i++){

        min\_idx = i;

        for (int j = i + 1; j < size; j++)

        {

            take(file, j, &value1);

            take(file, min\_idx, &value2);

            if (value1 < value2)

                min\_idx = j;

        }

        if (min\_idx != i){

            take(file, i, &value1);

            swap(file, i, &value1, min\_idx, &value2);

        }

    }

}

void block2(){

    FILE \* numbers = fopen("numbers.bin", "w+b");

    if (numbers == NULL){

        printf("\nFile opening error.\n");

    }

    int quantity, value;

    printf("\nEnter quantity of numbers to generate: ");

    scanf("%d", &quantity);

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

        value = rand() % 100;

        fwrite(&value, sizeof(int), 1, numbers);

    }

    int counter = 1;

    fseek(numbers, 0, SEEK\_SET);

    printf("\nNumbers in 'nubmers.bin' as generated:\n");

    while (fread(&value, sizeof(int), 1, numbers) != 0)

    {

        printf("%d ", value);

        counter++;

    }

    selectionSort(numbers, counter);

    fclose(numbers);

    numbers = fopen("numbers.bin", "rb");

    printf("\nNumbers in 'nubmers.bin' sorted:\n");

    while (fread(&value, sizeof(int), 1, numbers) != 0)

    {

        printf("%d ", value);

    }

    fclose(numbers);

}

void block3(){

    #define MAX\_LINE\_LENGTH 1000

    FILE \* f = fopen("script.txt", "r");

    char line[MAX\_LINE\_LENGTH]; int open\_br = 0, close\_br = 0, line\_counter = 1;

    if (f == NULL){

        printf("Failed to open the file!");

        exit(1);

        }

    while (fgets(line, MAX\_LINE\_LENGTH, f)){

        for (int i = 0; i < strlen(line); i++){

            if (line[i] == '(') open\_br++;

            else if (line[i] == ')') close\_br++;

            if (close\_br > open\_br) break;

        }

        if (open\_br != close\_br) printf("\nMissing/extra bracket on line %d\n", line\_counter);

        line\_counter++;

        open\_br = 0;

        close\_br = 0;

    }

    fclose(f);

}

void main(){

    srand(time(0));

    printf("\n=== BLOCK 1 ===\n");

    block1();

    printf("\n\n=== BLOCK 2 ===\n");

    block2();

    printf("\n\n=== BLOCK 3 ===\n");

    block3();

    printf("\n");

}

**Тестування програми**

**script.txt**

#include <stdio.h>

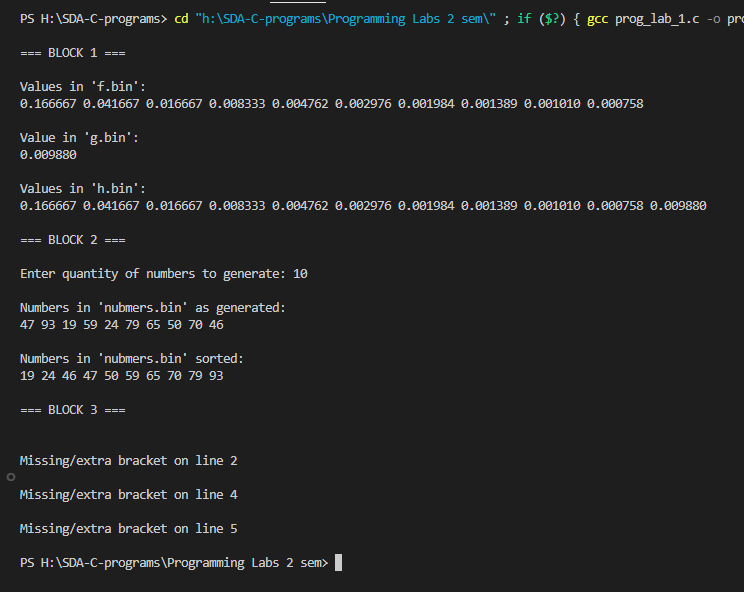
void func){};

void main(){

func);

printf("hello world!";

}

****

**script.txt**

#include <stdio.h>

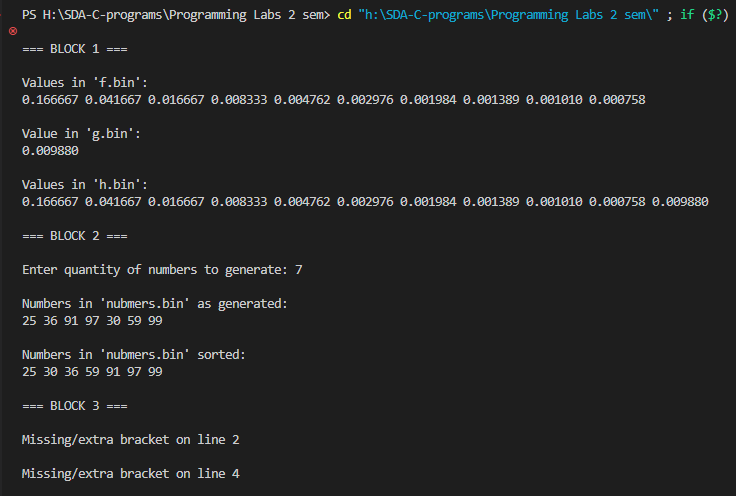
void func()){};

void main(){

func);

printf("hello world!)";

}

****

**script.txt**

#include <stdio.h>

void func()){};

void main(){

func)(;

printf("hello world!)";

}

