**Đề bài tương tác File, rand()**

**Bài 1: Quản lý thông tin sinh viên**

Viết chương trình C thực hiện các chức năng sau:

a. Tạo một file "sinhvien.txt" để lưu trữ thông tin sinh viên. Mỗi sinh viên gồm các thông tin: Mã sinh viên (tự sinh ngẫu nhiên), Tên, Điểm trung bình.

b. Ghi thông tin N sinh viên vào file (N nhập từ người dùng). Dữ liệu điểm trung bình được sinh ngẫu nhiên trong khoảng [0, 10].

c. Đọc và hiển thị danh sách sinh viên từ file.

d. Tìm và hiển thị thông tin sinh viên có điểm trung bình cao nhất.

e. Kiểm tra file và thông báo nếu file không tồn tại hoặc lỗi.

**Testcase kiểm tra:**

1. File không tồn tại.
2. File rỗng.
3. Số sinh viên nhập vào là 0.
4. Một sinh viên có điểm trung bình là 0.
5. Một sinh viên có điểm trung bình là 10.

**Bài 2: Quản lý số ngẫu nhiên**

Viết chương trình C thực hiện các chức năng sau:

a. Tạo một file "so\_ngau\_nhien.txt" để lưu trữ danh sách các số nguyên.

b. Ghi N số nguyên ngẫu nhiên (trong khoảng [1, 1000], N nhập từ người dùng) vào file.

c. Đọc và hiển thị danh sách số từ file.

d. Tìm số lớn nhất và nhỏ nhất trong danh sách.

e. Tính tổng các số trong file.

**Testcase kiểm tra:**

1. File không tồn tại.
2. File rỗng.
3. Số lượng số nguyên là 0.
4. Một số là giá trị nhỏ nhất (1).
5. Một số là giá trị lớn nhất (1000).

**Giải pháp (Code mẫu)**

**Bài 1: Quản lý thông tin sinh viên**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct

{

    int id;

    char name[50];

    float gpa;

} Student;

void writeStudentsToFile(const char \*filename, int n)

{

    system("cls");

    fflush(stdin);

    FILE \*file = fopen(filename, "w");

    if (!file)

    {

        perror("File cannot open!");

        return;

    }

    for (int i = 0; i < n; i++)

    {

        Student s;

        s.id = (rand() % (999 - 100 + 1)) + 100;

        printf("Student's name (1 word): ");

        fgets(s.name, sizeof(s.name), stdin);

        s.gpa = (float)(rand() % 101) / 10.0;

        fprintf(file, "%d %s %.2f\n", s.id, s.name, s.gpa);

    }

    fclose(file);

}

void readStudentsFromFile(const char \*filename)

{

    system("cls");

    FILE \*file = fopen(filename, "r");

    if (!file)

    {

        perror("File cannot open!");

        return;

    }

    Student s;

    printf("Student's list: \n");

    printf("%4s %10s %2s\n", "id", "name", "gpa");

    while (fscanf(file, "%d %s %f", &s.id, s.name, &s.gpa) == 3)

    {

        printf("%4d %10s %2.2f\n", s.id, s.name, s.gpa);

    }

    fclose(file);

}

void findTopStudent(const char \*filename)

{

    system("cls");

    FILE \*file = fopen(filename, "r");

    if (!file)

    {

        perror("File cannot open!");

        return;

    }

    Student s, topStudent;

    topStudent.gpa = -1.0;

    while (fscanf(file, "%d %s %f", &s.id, s.name, &s.gpa) == 3)

    {

        if (s.gpa > topStudent.gpa)

        {

            topStudent = s;

        }

    }

    fclose(file);

    if (topStudent.gpa != -1.0)

    {

        printf("The best student:  %d %s %.2f\n", topStudent.id, topStudent.name, topStudent.gpa);

    }

    else

    {

        printf("File is empty!\n");

    }

}

int main()

{

    const char \*filename = "sinhvien.txt";

    int choice, n;

    do

    {

        printf("1. Add student\n2. Read student's data\n3. Find the best student\n4. Exit\n");

        printf("->  ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            printf("Enter number of student: ");

            scanf("%d", &n);

            writeStudentsToFile(filename, n);

            break;

        case 2:

            readStudentsFromFile(filename);

            break;

        case 3:

            findTopStudent(filename);

            break;

        case 4:

            printf("Exit\n");

            break;

        default:

            printf("Invalid selection!\n");

        }

    } while (choice != 4);

    return 0;

}

**Bài 2: Quản lý số ngẫu nhiên**

#include <stdio.h>

#include <stdlib.h>

void writeRandomNumbersToFile(const char \*filename, int n)

{

    system("cls");

    FILE \*file = fopen(filename, "w");

    if (!file)

    {

        perror("File cannot open!");

        return;

    }

    for (int i = 0; i < n; i++)

    {

        fprintf(file, "%d\n", rand() % 1000 + 1);

    }

    fclose(file);

}

void readNumbersFromFile(const char \*filename)

{

    system("cls");

    FILE \*file = fopen(filename, "r");

    if (!file)

    {

        perror("File cannot open!");

        return;

    }

    int number;

    printf("List: \n");

    while (fscanf(file, "%d", &number) == 1)

    {

        printf("%5d ", number);

    }

    printf("\n");

    fclose(file);

}

void findMinMaxAndSum(const char \*filename)

{

    system("cls");

    FILE \*file = fopen(filename, "r");

    if (!file)

    {

        perror("File cannot open!");

        return;

    }

    int number, min = 1001, max = 0, sum = 0;

    while (fscanf(file, "%d", &number) == 1)

    {

        if (number < min)

            min = number;

        if (number > max)

            max = number;

        sum += number;

    }

    fclose(file);

    printf("min: %d\n", min);

    printf("max: %d\n", max);

    printf("sum: %d\n", sum);

}

int main()

{

    const char \*filename = "so\_ngau\_nhien.txt";

    int choice, n;

    do

    {

        printf("1. Add random number\n2. Read data\n3. Print max, min, sum\n4. Exit\n");

        printf("-> ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            printf("Enter number of number:");

            scanf("%d", &n);

            writeRandomNumbersToFile(filename, n);

            break;

        case 2:

            readNumbersFromFile(filename);

            break;

        case 3:

            findMinMaxAndSum(filename);

            break;

        case 4:

            printf("Exit\n");

            break;

        default:

            printf("Valid selection!\n");

        }

    } while (choice != 4);

    return 0;

}