Bài tập mẫu

Bài tập 1. Đọc 1 số nguyên



#include <stdio.h>

#define FI "Data.inp"

// Hàm đọc dữ liệu từ file

void inputData(int &n)

{

    FILE \*fi;

    fi = fopen(FI, "rt");

    if (fi == NULL)

    {

        printf("Khong the doc file\n");

        return;

    }

    fscanf(fi, "%d", &n);

    fclose(fi);

}

int main()

{

    int n;

    inputData(n);

    printf("n = %d\n", n);

    return 0;

}

Bài tập 2.Ghi các số nguyên



#include <stdio.h>

#define MAX 100

#define FO "Data.out"

// Hàm nhập mảng một chiều

void NhapMang1C(int a[], int &n)

{

    printf("Nhap so luong phan tu n: ");

    scanf("%d", &n);

    while (n <= 0 || n > MAX)

    {

        printf("So luong khong hop le! Nhap lai: ");

        scanf("%d", &n);

    }

    printf("Nhap %d phan tu cua mang:\n", n);

    for (int i = 0; i < n; i++)

    {

        printf("a[%d] = ", i);

        scanf("%d", &a[i]);

    }

}

// Hàm ghi mảng vào file

void OutputData(int a[], int n)

{

    FILE \*fo;

    fo = fopen(FO, "wt"); // Sửa lỗi biến 'fi'

    if (fo == NULL)

    {

        printf("Khong the tao file\n");

        return;

    }

    fprintf(fo, "%d\n", n);

    for (int i = 0; i < n; i++)

        fprintf(fo, "%d\n", a[i]);

    fclose(fo);

    printf("Ghi file thanh cong!\n");

}

int main()

{

    int a[MAX];

    int n;

    NhapMang1C(a, n);

    OutputData(a, n);

    return 0;

}

Bài tập cơ bản

Bài tập 1. Doc các số nguyên

Viết hàm đọc file chứa n số nguyên. Xuất kết quả đọc được ra màn hình.

Dữ liệu vào: Từ file văn bản DaySoNguyen.inp gồm

• Dòng thứ 1: Chứa số nguyên n là số phần tử của mảng.

• Dòng thứ 2: Chứa n số nguyên (các số cách nhau ít nhất một khoảng trắng).

Ví dụ: file DaySoNguyen.inp

5

4 3 5 3 2



#include <stdio.h>

#define MAX 100

#define FILE\_NAME "DaySoNguyen.inp"

// Hàm đọc mảng số nguyên từ file

void DocMangTuFile(int a[], int \*n)

{

    FILE \*fi = fopen(FILE\_NAME, "r");

    if (fi == NULL)

    {

        printf("Khong the mo file!\n");

        return;

    }

    // Đọc số lượng phần tử n

    fscanf(fi, "%d", n);

    if (\*n <= 0 || \*n > MAX)

    {

        printf("So luong phan tu khong hop le!\n");

        fclose(fi);

        return;

    }

    // Đọc n số nguyên

    for (int i = 0; i < \*n; i++)

    {

        fscanf(fi, "%d", &a[i]);

    }

    fclose(fi);

}

// Hàm xuất mảng ra màn hình

void XuatMang(int a[], int n)

{

    printf("Mang doc duoc tu file: ");

    for (int i = 0; i < n; i++)

    {

        printf("%d ", a[i]);

    }

    printf("\n");

}

// Chương trình chính

int main()

{

    int a[MAX], n;

    DocMangTuFile(a, &n);

    XuatMang(a, n);

    return 0;

}

Bài tập 2. Đọc mảng 2 chiều

Viết hàm đọc file chứa mảng 2 chiều các số nguyên. Xuất kết quả đọc được ra màn hình.

Dữ liệu vào: Từ file văn bản MangSo.inp gồm

• Dòng thứ 1: Chứa 2 số nguyên n, m là số dòng và số cột của mảng.

• n dòng tiếp theo: Mỗi dòng chứa n số nguyên.

o (các số cách nhau ít nhất một khoảng trắng)

Ví dụ: file MangSo.inp

3 5

5 3 5 7 2

4 5 5 7 1

5 4 5 3 9



#include <stdio.h>

#define MAX 100

#define FILE\_NAME "MangSo.inp"

// Hàm đọc mảng 2 chiều từ file

void DocMangTuFile(int a[MAX][MAX], int \*n, int \*m)

{

    FILE \*fi = fopen(FILE\_NAME, "r");

    if (fi == NULL)

    {

        printf("Khong the mo file!\n");

        return;

    }

    // Đọc số dòng và số cột

    fscanf(fi, "%d %d", n, m);

    if (\*n <= 0 || \*n > MAX || \*m <= 0 || \*m > MAX)

    {

        printf("Kich thuoc mang khong hop le!\n");

        fclose(fi);

        return;

    }

    // Đọc n dòng, mỗi dòng có m số nguyên

    for (int i = 0; i < \*n; i++)

    {

        for (int j = 0; j < \*m; j++)

        {

            fscanf(fi, "%d", &a[i][j]);

        }

    }

    fclose(fi);

}

// Hàm xuất mảng 2 chiều ra màn hình

void XuatMang(int a[MAX][MAX], int n, int m)

{

    printf("Mang 2 chieu doc duoc tu file:\n");

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < m; j++)

        {

            printf("%d ", a[i][j]);

        }

        printf("\n");

    }

}

// Chương trình chính

int main()

{

    int a[MAX][MAX], n, m;

    DocMangTuFile(a, &n, &m);

    XuatMang(a, n, m);

    return 0;

}

Bài tập 3. Tìm số nguyên tố Tên file chương trình:NguyenTo.\*

Cho dãy n số nguyên. Hãy liệt kê các số nguyên tố trong dãy theo thứ tự tăng dần.

Dữ liệu vào: Từ file văn bản NT.INP gồm có

• Dòng đầu là n (1<n≤10.000)

• n dòng tiếp theo, dòng thứ i ghi số ai

Kết quả: Ghi ra file văn bản NT.OUT

• Dòng đầu ghi số m là số lượng số nguyên tố (Nếu không có số nguyên tố nào thì ghi số 0)

• Dòng thứ 2 ghi m số nguyen tố tìm được (các số cách nhau ít nhất một khoảng trắng)



#include <stdio.h>

#include <stdlib.h>

#define MAX 10000

#define INPUT\_FILE "NT.INP"

#define OUTPUT\_FILE "NT.OUT"

// Hàm kiểm tra số nguyên tố

int LaSoNguyenTo(int n)

{

    if (n < 2)

        return 0;

    for (int i = 2; i \* i <= n; i++)

    {

        if (n % i == 0)

            return 0;

    }

    return 1;

}

// Hàm đọc dữ liệu từ file

void DocDuLieu(int a[], int \*n)

{

    FILE \*fi = fopen(INPUT\_FILE, "r");

    if (fi == NULL)

    {

        printf("Khong the mo file %s!\n", INPUT\_FILE);

        exit(1);

    }

    fscanf(fi, "%d", n);

    if (\*n <= 1 || \*n > MAX)

    {

        printf("So luong phan tu khong hop le!\n");

        fclose(fi);

        exit(1);

    }

    for (int i = 0; i < \*n; i++)

    {

        fscanf(fi, "%d", &a[i]);

    }

    fclose(fi);

}

// Hàm ghi kết quả ra file

void GhiKetQua(int prime[], int m)

{

    FILE \*fo = fopen(OUTPUT\_FILE, "w");

    if (fo == NULL)

    {

        printf("Khong the tao file %s!\n", OUTPUT\_FILE);

        exit(1);

    }

    fprintf(fo, "%d\n", m);

    for (int i = 0; i < m; i++)

    {

        fprintf(fo, "%d ", prime[i]);

    }

    fclose(fo);

}

// Hàm tìm và lưu các số nguyên tố

void XuLySoNguyenTo(int a[], int n)

{

    int prime[MAX], m = 0;

    // Tìm số nguyên tố

    for (int i = 0; i < n; i++)

    {

        if (LaSoNguyenTo(a[i]))

        {

            prime[m++] = a[i];

        }

    }

    // Sắp xếp mảng số nguyên tố theo thứ tự tăng dần

    for (int i = 0; i < m - 1; i++)

    {

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

        {

            if (prime[i] > prime[j])

            {

                int temp = prime[i];

                prime[i] = prime[j];

                prime[j] = temp;

            }

        }

    }

    // Ghi kết quả ra file

    GhiKetQua(prime, m);

}

// Chương trình chính

int main()

{

    int a[MAX], n;

    DocDuLieu(a, &n);

    XuLySoNguyenTo(a, n);

    printf("Ket qua da duoc ghi vao file %s\n", OUTPUT\_FILE);

    return 0;

}

Bài tập 4. Tìm cặp số Tên file chương trình:CapSo.\*

Cho dãy n số nguyên và số nguyên k. Hãy tìm các cặp (ai, aj) sao cho ai+aj=k.

Dữ liệu vào: Từ file văn bản CapSo.INP gồm có

• Dòng đầu chứa số n (1<n≤10.000) và k

• n dòng tiếp theo, dòng thứ i ghi số ai

Kết quả: Ghi ra file văn bản CapSo.OUT

• Nếu không tìm được cặp số nào ghi số 0. Ngược lại ghi các cặp số tìm được, mỗi cặp nằm trên

1 dòng (Các số cách nhau ít nhất 1 khoảng trắng).



#include <stdio.h>

#include <stdlib.h>

#define MAX 10000

#define INPUT\_FILE "CapSo.INP"

#define OUTPUT\_FILE "CapSo.OUT"

// Hàm đọc dữ liệu từ file

void DocDuLieu(int a[], int \*n, int \*k)

{

    FILE \*fi = fopen(INPUT\_FILE, "r");

    if (fi == NULL)

    {

        printf("Khong the mo file %s!\n", INPUT\_FILE);

        exit(1);

    }

    fscanf(fi, "%d %d", n, k);

    if (\*n <= 1 || \*n > MAX)

    {

        printf("So luong phan tu khong hop le!\n");

        fclose(fi);

        exit(1);

    }

    for (int i = 0; i < \*n; i++)

    {

        fscanf(fi, "%d", &a[i]);

    }

    fclose(fi);

}

// Hàm tìm và ghi cặp số có tổng bằng k

void TimCapSo(int a[], int n, int k)

{

    FILE \*fo = fopen(OUTPUT\_FILE, "w");

    if (fo == NULL)

    {

        printf("Khong the tao file %s!\n", OUTPUT\_FILE);

        exit(1);

    }

    int timThay = 0;

    // Duyệt tất cả các cặp số (ai, aj) với i < j

    for (int i = 0; i < n - 1; i++)

    {

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

        {

            if (a[i] + a[j] == k)

            {

                fprintf(fo, "%d %d\n", a[i], a[j]);

                timThay = 1;

            }

        }

    }

    // Nếu không tìm thấy cặp số nào, ghi 0 vào file

    if (!timThay)

    {

        fprintf(fo, "0\n");

    }

    fclose(fo);

}

// Chương trình chính

int main()

{

    int a[MAX], n, k;

    DocDuLieu(a, &n, &k);

    TimCapSo(a, n, k);

    printf("Ket qua da duoc ghi vao file %s\n", OUTPUT\_FILE);

    return 0;

}