**THỰC HÀNH CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT 2025**

**Lab1e**

Tên : Trịnh Tâm Như

Mssv: 3123411215

**Code**

**Bài 1**

#include <iostream>

#include <fstream>

using namespace std;

void DocCacSoNguyen(const char\* filename, int a[], int& n) {

ifstream fi(filename); // Mở tệp để đọc

if (!fi) {

cout << "Khong the mo file " << filename << endl;

return;

}

fi >> n; // Đọc số phần tử n từ tệp

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

fi >> a[i]; // Đọc các số nguyên vào mảng a

}

fi.close(); // Đóng tệp sau khi đọc

}

void XuatMang(int a[], int n) {

cout << "Day so nguyen doc duoc la: ";

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

cout << a[i] << " "; // In các phần tử của mảng ra màn hình

}

cout << endl;

}

int main() {

const char\* filename = "DaySoNguyen.inp"; // Tên tệp chứa dữ liệu

int a[1000], n;

DocCacSoNguyen(filename, a, n); // Gọi hàm để đọc dữ liệu từ tệp

XuatMang(a, n); // Xuất mảng ra màn hình

return 0;

}

**TEST CASE**

**nội dung tệp "DaySoNguyen.inp":**

5

4 3 5 3 2

**OUTPUT**

Day so nguyen doc duoc la: 4 3 5 3 2

Bài 2

#include <iostream>

#include <fstream>

using namespace std;

void DocMang2C(int a[100][100], int& n, int& m, const char\* filename) {

ifstream fi(filename); // Mở tệp để đọc

if (!fi) {

cout << "Khong the mo file " << filename << endl;

return;

}

fi >> n >> m; // Đọc số dòng n và số cột m từ tệp

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

for (int j = 0; j < m; j++) {

fi >> a[i][j]; // Đọc các phần tử mảng 2 chiều vào a

}

}

fi.close(); // Đóng tệp sau khi đọc

}

void XuatMang2C(int a[100][100], int n, int m) {

cout << "Mang 2 chieu doc duoc la: \n";

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

for (int j = 0; j < m; j++) {

cout << a[i][j] << " "; // In các phần tử mảng 2 chiều

}

cout << endl;

}

}

int main() {

const char\* filename = "MangSo.inp"; // Tên tệp chứa dữ liệu

int a[100][100], n, m;

DocMang2C(a, n, m, filename); // Gọi hàm để đọc dữ liệu từ tệp

XuatMang2C(a, n, m); // Xuất mảng ra màn hình

return 0;

}

**TEST CASE**

**MangSo.inp**

3 3

1 2 3

4 5 6

7 8 9

**Output**

Mang 2 chieu doc duoc la:

1 2 3

4 5 6

7 8 9

**Bài 3**

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

using namespace std;

// Hàm kiểm tra số nguyên tố

bool isPrime(int n) {

if (n < 2) return false; // Các số nhỏ hơn 2 không phải là số nguyên tố

for (int i = 2; i \* i <= n; i++) {

if (n % i == 0) return false; // Nếu n chia hết cho i, thì không phải số nguyên tố

}

return true;

}

// Hàm đọc dữ liệu từ tệp và trả về các số nguyên

void DocSoNguyenTuFile(const string& filename, vector<int>& arr) {

ifstream fi(filename);

if (!fi) {

cout << "Khong the mo file " << filename << endl;

return;

}

int n;

fi >> n; // Đọc số phần tử n

arr.resize(n); // Cấp phát không gian cho mảng

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

fi >> arr[i]; // Đọc từng số nguyên

}

fi.close();

}

// Hàm ghi kết quả ra tệp

void GhiKetQuaVaoFile(const string& filename, const vector<int>& arr) {

ofstream fo(filename);

if (!fo) {

cout << "Khong the ghi vao file " << filename << endl;

return;

}

// Ghi số lượng các số nguyên tố

fo << arr.size() << endl;

// Ghi các số nguyên tố, cách nhau một khoảng trắng

for (int num : arr) {

fo << num << " ";

}

fo.close();

}

int main() {

vector<int> numbers;

vector<int> primes;

// Đọc các số nguyên từ file

DocSoNguyenTuFile("NT.INP", numbers);

// Tìm các số nguyên tố trong dãy

for (int num : numbers) {

if (isPrime(num)) {

primes.push\_back(num);

}

}

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

sort(primes.begin(), primes.end());

// Ghi kết quả vào file NT.OUT

GhiKetQuaVaoFile("NT.OUT", primes);

return 0;

}

TESTCASE

**NT.inp**

10

3 4 5 6 7 8 9 10 11 12

**NT.out**

4

3 5 7 11

**Bài 4**

#include <iostream>

#include <fstream>

#include <vector>

#include <unordered\_set>

using namespace std;

void TimCapSo(const string& filenameIn, const string& filenameOut) {

ifstream fi(filenameIn);

if (!fi) {

cout << "Khong the mo file " << filenameIn << endl;

return;

}

int n, k;

fi >> n >> k; // Đọc số phần tử và số k

vector<int> a(n);

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

fi >> a[i]; // Đọc các phần tử của mảng

}

fi.close(); // Đóng tệp sau khi đọc

unordered\_set<int> seen; // Sử dụng một bộ để kiểm tra các phần tử đã gặp

vector<pair<int, int>> pairs; // Lưu các cặp tìm được

// Tìm các cặp số sao cho ai + aj = k

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

int complement = k - a[i]; // Tìm số bù complement

if (seen.find(complement) != seen.end()) {

pairs.push\_back({ complement, a[i] }); // Thêm cặp (complement, a[i]) vào danh sách

}

seen.insert(a[i]); // Đánh dấu a[i] đã gặp

}

// Ghi kết quả vào tệp

ofstream fo(filenameOut);

if (!fo) {

cout << "Khong the ghi vao file " << filenameOut << endl;

return;

}

if (pairs.empty()) {

fo << 0 << endl; // Nếu không có cặp nào, ghi 0

}

else {

for (const auto& p : pairs) {

fo << p.first << " " << p.second << endl; // Ghi các cặp số vào tệp

}

}

fo.close(); // Đóng tệp sau khi ghi

}

int main() {

TimCapSo("CapSo.INP", "CapSo.OUT");

return 0;

}

**TESTCASE**

**CapSo.inp**

5 10

1 2 5 8 9

**CapSo.out**

1 9

2 8

5 5