ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

BÁO CÁO MÔN HỌC:

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tên sinh viên : DƯƠNG TRỌNG PHƯỚC

Mã số sinh viên : 3124411250

Lớp DCT 124C6

Giảng Viên:ĐỖ NHƯ TÀI

Ngày nộp:11/1/2025

**Lab01b**

**Bài 1**

**Code**

#include <stdio.h>

#include <math.h>

void giaiPTBacHai(float \*a, float \*b, float \*c) {

float delta = (\*b) \* (\*b) - 4 \* (\*a) \* (\*c);

if (delta > 0) {

float x1 = (-(\*b) + sqrt(delta)) / (2 \* (\*a));

float x2 = (-(\*b) - sqrt(delta)) / (2 \* (\*a));

printf("Phuong trinh co hai nghiem: x1 = %.2f, x2 = %.2f\n", x1, x2);

} else if (delta == 0) {

float x = -(\*b) / (2 \* (\*a));

printf("Phuong trinh co 1 nghiem kep: x = %.2f\n", x);

} else {

printf("Phuong trinh vo nghiem\n");

}

}

int main() {

float a, b, c;

printf("Moi ban nhap he so a, b, c: ");

scanf("%f %f %f", &a, &b, &c);

if (a == 0) {

if (b == 0) {

if (c == 0) {

printf("Phuong trinh vo so nghiem\n");

} else {

printf("Phuong trinh vo nghiem\n");

}

} else {

float x = -c / b;

printf("Phuong trinh co 1 nghiem: x = %.2f\n", x);

}

} else {

giaiPTBacHai(&a, &b, &c);

}

return 0;

}

**\*mã nguồn**



**2**

**Code**

|  |
| --- |
| #include <stdio.h>  void kiemTraTinhChat(int \*a, int n) {  int chanLe = 1, toanChan = 1;    for (int i = 0; i < n; i++) {  if (a[i] % 2 != 0) {  toanChan = 0;  }  if (i > 0 && (a[i] % 2 == a[i - 1] % 2)) {  chanLe = 0;  }  }    if (chanLe) {  printf("+ Day co tinh chat chan le\n");  } else {  printf("+ Day khong co tinh chat chan le\n");  }    if (toanChan) {  printf("+ Day co tinh chat toan chan\n");  } else {  printf("+ Day khong co tinh chat toan chan\n");  }  }  int main() {  int n;  int a[100];    printf("Moi ban nhap so luong phan tu: ");  scanf("%d", &n);    for (int i = 0; i < n; i++) {  printf("Phan tu %d: ", i);  scanf("%d", &a[i]);  }    kiemTraTinhChat(a, n);    return 0;  } |

**\*mã nguồn**

****

**3**

**Code**

#include <stdio.h>

#include <ctype.h>

void tachNguyenAm(char \*a, int n) {

char b[100];

int j = 0;

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

if (a[i] == 'a' || a[i] == 'e' || a[i] == 'i' || a[i] == 'o' || a[i] == 'u' ||

a[i] == 'A' || a[i] == 'E' || a[i] == 'I' || a[i] == 'O' || a[i] == 'U') {

b[j++] = a[i];

}

}

b[j] = '\0';

printf("Day ky tu nguyen am: %s\n", b);

}

int main() {

int n;

char a[100];

printf("Moi ban nhap so luong phan tu: ");

scanf("%d", &n);

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

printf("Phan tu %d: ", i);

scanf(" %c", &a[i]);

}

tachNguyenAm(a, n);

return 0;

}

**\*mã nguồn**



**4**

Code

#include <stdio.h>

void tronHaiDay(float \*a, int n, float \*b, int m, float \*c) {

int i = 0, j = 0, k = 0;

while (i < n && j < m) {

if (a[i] < b[j]) {

c[k++] = a[i++];

} else {

c[k++] = b[j++];

}

}

while (i < n) {

c[k++] = a[i++];

}

while (j < m) {

c[k++] = b[j++];

}

}

int main() {

int n, m;

float a[100], b[100], c[200];

printf("Moi ban nhap so luong phan tu cua day a: ");

scanf("%d", &n);

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

printf("Phan tu a[%d]: ", i);

scanf("%f", &a[i]);

}

printf("Moi ban nhap so luong phan tu cua day b: ");

scanf("%d", &m);

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

printf("Phan tu b[%d]: ", i);

scanf("%f", &b[i]);

}

tronHaiDay(a, n, b, m, c);

printf("Day c sau khi tron: ");

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

printf("%.2f ", c[i]);

}

printf("\n");

return 0;

}

**\*mã nguồn**

****

**5**

**Code**

#include <stdio.h>

void tachChanLe(int \*a, int n, int \*b, int \*m, int \*c, int \*k) {

\*m = 0;

\*k = 0;

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

if (a[i] % 2 == 0) {

b[(\*m)++] = a[i];

} else {

c[(\*k)++] = a[i];

}

}

}

int main() {

int n, m, k;

int a[100], b[100], c[100];

printf("Moi ban nhap so luong phan tu: ");

scanf("%d", &n);

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

printf("Phan tu %d: ", i);

scanf("%d", &a[i]);

}

tachChanLe(a, n, b, &m, c, &k);

printf("Day so chan: ");

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

printf("%d ", b[i]);

}

printf("\n");

printf("Day so le: ");

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

printf("%d ", c[i]);

}

printf("\n");

return 0;

}

**\*mã nguồn**

****

**Lab01c**

**Bài 1**

#include <stdio.h>

#include <string.h>

typedef struct {

char MaSo[11];

char Ho[11];

char Ten[51];

int Phai;

int ThamNien;

} NhanVien;

typedef struct {

NhanVien aNhanVien[100];

int SoLuong;

} PhongBan;

void NhapPhongBan(PhongBan \*pb) {

printf("Nhap so luong nhan vien: ");

scanf("%d", &pb->SoLuong);

for (int i = 0; i < pb->SoLuong; i++) {

printf("Nhap thong tin nhan vien %d:\n", i + 1);

printf("Ma so: ");

scanf("%s", pb->aNhanVien[i].MaSo);

printf("Ho: ");

scanf("%s", pb->aNhanVien[i].Ho);

printf("Ten: ");

scanf("%s", pb->aNhanVien[i].Ten);

printf("Phai (0 - Nu, 1 - Nam): ");

scanf("%d", &pb->aNhanVien[i].Phai);

printf("Tham nien: ");

scanf("%d", &pb->aNhanVien[i].ThamNien);

}

}

void XuatPhongBan(PhongBan pb) {

printf("Danh sach nhan vien:\n");

for (int i = 0; i < pb.SoLuong; i++) {

printf("%s %s %s, Phai: %s, Tham nien: %d\n", pb.aNhanVien[i].MaSo, pb.aNhanVien[i].Ho,

pb.aNhanVien[i].Ten, pb.aNhanVien[i].Phai ? "Nam" : "Nu", pb.aNhanVien[i].ThamNien);

}

}

void DemSiSo(PhongBan pb, int \*sonam, int \*sonu) {

\*sonam = \*sonu = 0;

for (int i = 0; i < pb.SoLuong; i++) {

if (pb.aNhanVien[i].Phai == 1) (\*sonam)++;

else (\*sonu)++;

}

}

void SapXepTangTheoThamNien(PhongBan \*pb) {

for (int i = 0; i < pb->SoLuong - 1; i++) {

for (int j = i + 1; j < pb->SoLuong; j++) {

if (pb->aNhanVien[i].ThamNien > pb->aNhanVien[j].ThamNien) {

NhanVien temp = pb->aNhanVien[i];

pb->aNhanVien[i] = pb->aNhanVien[j];

pb->aNhanVien[j] = temp;

}

}

}

}

int main() {

PhongBan pb;

int sonam, sonu;

NhapPhongBan(&pb);

XuatPhongBan(pb);

DemSiSo(pb, &sonam, &sonu);

printf("Si so Nam: %d, Si so Nu: %d\n", sonam, sonu);

SapXepTangTheoThamNien(&pb);

printf("Danh sach sau khi sap xep theo tham nien:\n");

XuatPhongBan(pb);

return 0;

}

**\*mã nguồn**

****

**Bài 2**

#include <stdio.h>

#include <string.h>

#define MAX\_LOAI\_HOA 20

// Cấu trúc LoaiHoa lưu trữ thông tin một loại hoa

struct LoaiHoa {

char Ten[50]; // Tên loại hoa

int SoLuong; // Số lượng hoa hiện có

char DVT[20]; // Đơn vị tính

float DonGia; // Đơn giá

};

// Cấu trúc DanhSachLoaiHoa lưu trữ danh sách các loại hoa

struct DanhSachLoaiHoa {

struct LoaiHoa ds[MAX\_LOAI\_HOA]; // Danh sách các loại hoa

int n; // Số lượng loại hoa hiện tại

};

// Hàm nhập danh sách các loại hoa

void NhapDanhSach(struct DanhSachLoaiHoa \*ds) {

printf("Nhập số lượng loại hoa: ");

scanf("%d", &ds->n);

getchar(); // Bỏ qua ký tự '\n' còn lại trong bộ đệm

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

printf("Nhập thông tin loại hoa thứ %d:\n", i + 1);

printf("Tên loại: ");

fgets(ds->ds[i].Ten, sizeof(ds->ds[i].Ten), stdin);

ds->ds[i].Ten[strcspn(ds->ds[i].Ten, "\n")] = 0; // Xóa ký tự '\n'

printf("Số lượng: ");

scanf("%d", &ds->ds[i].SoLuong);

getchar(); // Bỏ qua ký tự '\n' còn lại

printf("Đơn vị tính: ");

fgets(ds->ds[i].DVT, sizeof(ds->ds[i].DVT), stdin);

ds->ds[i].DVT[strcspn(ds->ds[i].DVT, "\n")] = 0; // Xóa ký tự '\n'

printf("Đơn giá: ");

scanf("%f", &ds->ds[i].DonGia);

getchar(); // Bỏ qua ký tự '\n' còn lại

}

}

// Hàm xuất danh sách ra màn hình

void XuatDanhSach(struct DanhSachLoaiHoa ds) {

printf("Danh sách các loại hoa:\n");

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

printf("Loại hoa %d:\n", i + 1);

printf("Tên loại: %s\n", ds.ds[i].Ten);

printf("Số lượng: %d\n", ds.ds[i].SoLuong);

printf("Đơn vị tính: %s\n", ds.ds[i].DVT);

printf("Đơn giá: %.2f\n", ds.ds[i].DonGia);

printf("---------------------------------\n");

}

}

// Hàm tìm loại hoa theo tên loại

int TimLoaiHoa(struct DanhSachLoaiHoa ds, char \*tenloai) {

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

if (strcmp(ds.ds[i].Ten, tenloai) == 0) {

return i; // Trả về vị trí loại hoa trong danh sách

}

}

return -1; // Không tìm thấy

}

// Hàm xử lý bán hoa cho khách hàng

void XuLyBanHoa(struct DanhSachLoaiHoa \*ds, char \*tenloai, int soluong) {

int viTri = TimLoaiHoa(\*ds, tenloai);

if (viTri == -1) {

printf("Loại hoa không tồn tại trong danh sách.\n");

} else {

if (ds->ds[viTri].SoLuong >= soluong) {

float tien = soluong \* ds->ds[viTri].DonGia;

printf("Loại hoa: %s\n", ds->ds[viTri].Ten);

printf("Số lượng đủ, tổng tiền: %.2f\n", tien);

ds->ds[viTri].SoLuong -= soluong; // Giảm số lượng hoa sau khi bán

} else {

printf("Số lượng không đủ.\n");

}

}

}

// Hàm main sử dụng các hàm trên để xử lý yêu cầu đề bài

int main() {

struct DanhSachLoaiHoa ds;

int luaChon;

do {

printf("1. Nhập danh sách các loại hoa\n");

printf("2. Xuất danh sách các loại hoa\n");

printf("3. Tìm loại hoa và bán\n");

printf("4. Thoát\n");

printf("Nhập lựa chọn: ");

scanf("%d", &luaChon);

getchar(); // Bỏ qua ký tự '\n' còn lại trong bộ đệm

switch (luaChon) {

case 1:

NhapDanhSach(&ds);

break;

case 2:

XuatDanhSach(ds);

break;

case 3: {

char tenloai[50];

int soluong;

printf("Nhập tên loại hoa cần mua: ");

fgets(tenloai, sizeof(tenloai), stdin);

tenloai[strcspn(tenloai, "\n")] = 0; // Xóa ký tự '\n'

printf("Nhập số lượng: ");

scanf("%d", &soluong);

XuLyBanHoa(&ds, tenloai, soluong);

break;

}

case 4:

printf("Thoát chương trình.\n");

break;

default:

printf("Lựa chọn không hợp lệ.\n");

}

} while (luaChon != 4);

return 0;

}

**\*mã nguồn**

****

**Lab01d**

**Bài 1**

**Code**

#include <stdio.h>

#include <string.h>

// Hàm hoán đổi ký tự

void swap(char \*a, char \*b) {

char temp = \*a;

\*a = \*b;

\*b = temp;

}

// Hàm sắp xếp chuỗi theo thứ tự tăng dần (Bubble Sort)

void SapXepChuoi(char \*s) {

int n = strlen(s);

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

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

if (s[j] > s[j + 1]) {

swap(&s[j], &s[j + 1]);

}

}

}

}

int main() {

char s[1000];

// Nhập chuỗi từ người dùng

printf("Moi ban nhap chuoi s: ");

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

// Xóa ký tự '\n' nếu có

s[strcspn(s, "\n")] = '\0';

// Sắp xếp chuỗi

SapXepChuoi(s);

// Xuất kết quả

printf("Chuoi sau khi sap xep: %s\n", s);

return 0;

}

**\*mã nguồn**

****

**Bài 2**

**Code**

#include <stdio.h>

#include <string.h>

// Hàm xóa ký tự tại vị trí k trong chuỗi

void XoaKyTu(char \*s, int k) {

int n = strlen(s);

// Kiểm tra k có hợp lệ không

if (k < 0 || k >= n) {

printf("Vi tri khong hop le!\n");

return;

}

// Dịch các ký tự về phía trước để xóa ký tự tại vị trí k

for (int i = k; i < n - 1; i++) {

s[i] = s[i + 1];

}

s[n - 1] = '\0'; // Xóa ký tự cuối chuỗi

}

int main() {

char s[1000];

int k;

// Nhập chuỗi

printf("Moi ban nhap chuoi s: ");

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

s[strcspn(s, "\n")] = '\0'; // Xóa ký tự '\n' nếu có

// Nhập vị trí cần xóa

printf("Moi ban nhap vi tri can xoa: ");

scanf("%d", &k);

// Xóa ký tự tại vị trí k

XoaKyTu(s, k);

// Xuất kết quả

printf("Chuoi sau khi xoa ky tu tai vi tri %d: %s\n", k, s);

return 0;

}

**\*mã nguồn**

****

**Bài 3**

**Code**

#include <stdio.h>

#include <string.h>

// Hàm chèn ký tự c vào chuỗi s tại vị trí k

void ChenKyTu(char \*s, int k, char c) {

int n = strlen(s);

// Kiểm tra vị trí k hợp lệ

if (k < 0 || k > n) {

printf("Vi tri khong hop le!\n");

return;

}

// Dịch các ký tự sang phải để tạo chỗ trống cho ký tự mới

for (int i = n; i >= k; i--) {

s[i + 1] = s[i];

}

// Chèn ký tự mới vào vị trí k

s[k] = c;

}

int main() {

char s[1001]; // Mảng chứa chuỗi (tăng lên 1 ký tự để chứa ký tự chèn)

int k;

char c;

// Nhập chuỗi

printf("Moi ban nhap chuoi s: ");

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

s[strcspn(s, "\n")] = '\0'; // Xóa ký tự '\n' nếu có

// Nhập vị trí cần chèn

printf("Moi ban nhap vi tri can chen: ");

scanf("%d", &k);

getchar(); // Xóa ký tự '\n' trong bộ đệm

// Nhập ký tự cần chèn

printf("Moi ban nhap ky tu chen: ");

scanf("%c", &c);

// Chèn ký tự vào chuỗi

ChenKyTu(s, k, c);

// Xuất kết quả

printf("Chuoi sau khi them ky tu \"%c\" vao vi tri %d: %s\n", c, k, s);

return 0;

}

**\*mã nguồn**

****

**Lab01e**

**Bài 1**

**Code**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | #include <stdio.h>  #define MAX\_SIZE 1000 // Giới hạn số lượng phần tử tối đa  // Hàm đọc dữ liệu từ file và xuất kết quả ra màn hình  void DocCacSoNguyen(const char \*tenFile) {  FILE \*file = fopen(tenFile, "r"); // Mở file để đọc  if (file == NULL) {  printf("Khong the mo file %s!\n", tenFile);  return;  }  int n, arr[MAX\_SIZE];  // Đọc số lượng phần tử  if (fscanf(file, "%d", &n) != 1 || n <= 0 || n > MAX\_SIZE) {  printf("Loi: So luong phan tu khong hop le!\n");  fclose(file);  return;  }  // Đọc n số nguyên từ file  for (int i = 0; i < n; i++) {  if (fscanf(file, "%d", &arr[i]) != 1) {  printf("Loi: Du lieu trong file khong hop le!\n");  fclose(file);  return;  }  }  fclose(file); // Đóng file sau khi đọc xong  // Xuất kết quả ra màn hình  printf("So luong phan tu: %d\n", n);  printf("Cac so nguyen trong file: ");  for (int i = 0; i < n; i++) {  printf("%d ", arr[i]);  }  printf("\n");  }  int main() {  DocCacSoNguyen("DaySoNguyen.inp");  return 0;  } |

**\*mã nguồn**

****

**Bài 2**

**Code**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49 | #include <stdio.h>  #define MAX\_ROWS 100 // Giới hạn số dòng tối đa  #define MAX\_COLS 100 // Giới hạn số cột tối đa  // Hàm đọc mảng 2 chiều từ file và xuất ra màn hình  void DocMang2CSoNguyen(const char \*tenFile) {  FILE \*file = fopen(tenFile, "r"); // Mở file để đọc  if (file == NULL) {  printf("Khong the mo file %s!\n", tenFile);  return;  }  int n, m, arr[MAX\_ROWS][MAX\_COLS];  // Đọc số dòng (n) và số cột (m)  if (fscanf(file, "%d %d", &n, &m) != 2 || n <= 0 || m <= 0 || n > MAX\_ROWS || m > MAX\_COLS) {  printf("Loi: Kich thuoc mang khong hop le!\n");  fclose(file);  return;  }  // Đọc mảng 2 chiều từ file  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  if (fscanf(file, "%d", &arr[i][j]) != 1) {  printf("Loi: Du lieu trong file khong hop le!\n");  fclose(file);  return;  }  }  }  fclose(file); // Đóng file sau khi đọc xong  // Xuất kết quả ra màn hình  printf("Ma tran %d x %d:\n", n, m);  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  printf("%d ", arr[i][j]);  }  printf("\n");  }  }  int main() {  DocMang2CSoNguyen("MangSo.inp");  return 0;  } |

**\*mã nguồn**

****

**Bài 3**

**Code**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86 | #include <stdio.h>  #include <stdlib.h>  #define MAX\_SIZE 10000 // Giới hạn số phần tử tối đa  // 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 số từ file, tìm số nguyên tố và ghi kết quả vào file  void TimSoNguyenTo(const char \*fileInput, const char \*fileOutput) {  FILE \*fin = fopen(fileInput, "r");  if (fin == NULL) {  printf("Khong the mo file %s!\n", fileInput);  return;  }  int n, arr[MAX\_SIZE], primes[MAX\_SIZE], count = 0;  // Đọc số lượng phần tử  if (fscanf(fin, "%d", &n) != 1 || n <= 1 || n > MAX\_SIZE) {  printf("Loi: So luong phan tu khong hop le!\n");  fclose(fin);  return;  }  // Đọc dãy số nguyên từ file  for (int i = 0; i < n; i++) {  if (fscanf(fin, "%d", &arr[i]) != 1) {  printf("Loi: Du lieu trong file khong hop le!\n");  fclose(fin);  return;  }  }  fclose(fin); // Đóng file sau khi đọc xong  // Lọc các số nguyên tố và lưu vào mảng primes[]  for (int i = 0; i < n; i++) {  if (LaSoNguyenTo(arr[i])) {  primes[count++] = arr[i];  }  }  // Sắp xếp mảng số nguyên tố theo thứ tự tăng dần (Bubble Sort)  for (int i = 0; i < count - 1; i++) {  for (int j = i + 1; j < count; j++) {  if (primes[i] > primes[j]) {  int temp = primes[i];  primes[i] = primes[j];  primes[j] = temp;  }  }  }  // Ghi kết quả vào file NT.OUT  FILE \*fout = fopen(fileOutput, "w");  if (fout == NULL) {  printf("Khong the mo file %s de ghi ket qua!\n", fileOutput);  return;  }  // Ghi số lượng số nguyên tố  fprintf(fout, "%d\n", count);  // Ghi danh sách số nguyên tố (nếu có)  if (count > 0) {  for (int i = 0; i < count; i++) {  fprintf(fout, "%d ", primes[i]);  }  }  fclose(fout); // Đóng file sau khi ghi xong  printf("Ket qua da duoc ghi vao %s\n", fileOutput);  }  int main() {  TimSoNguyenTo("NT.INP", "NT.OUT");  return 0;  } |

**\*mã nguồn**

****

**Bài 4**

**Code**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66 | #include <stdio.h>  #include <stdlib.h>  #define MAX\_SIZE 10000 // Giới hạn số phần tử tối đa  // Hàm tìm các cặp (ai, aj) sao cho ai + aj = k  void TimCapSo(const char \*fileInput, const char \*fileOutput) {  FILE \*fin = fopen(fileInput, "r");  if (fin == NULL) {  printf("Khong the mo file %s!\n", fileInput);  return;  }  int n, k, arr[MAX\_SIZE];  // Đọc số lượng phần tử (n) và giá trị k  if (fscanf(fin, "%d %d", &n, &k) != 2 || n <= 1 || n > MAX\_SIZE) {  printf("Loi: So luong phan tu hoac k khong hop le!\n");  fclose(fin);  return;  }  // Đọc dãy số nguyên từ file  for (int i = 0; i < n; i++) {  if (fscanf(fin, "%d", &arr[i]) != 1) {  printf("Loi: Du lieu trong file khong hop le!\n");  fclose(fin);  return;  }  }  fclose(fin); // Đóng file sau khi đọc xong  // Mở file để ghi kết quả  FILE \*fout = fopen(fileOutput, "w");  if (fout == NULL) {  printf("Khong the mo file %s de ghi ket qua!\n", fileOutput);  return;  }  int found = 0; // Biến kiểm tra có tìm thấy cặp nào không  // Tìm các cặp số thỏa mãn ai + aj = k  for (int i = 0; i < n - 1; i++) {  for (int j = i + 1; j < n; j++) {  if (arr[i] + arr[j] == k) {  fprintf(fout, "%d %d\n", arr[i], arr[j]);  found = 1;  }  }  }  // Nếu không tìm thấy cặp nào, ghi số 0 vào file  if (!found) {  fprintf(fout, "0\n");  }  fclose(fout); // Đóng file sau khi ghi xong  printf("Ket qua da duoc ghi vao %s\n", fileOutput);  }  int main() {  TimCapSo("CapSo.INP", "CapSo.OUT");  return 0;  } |

**\*mã nguồn**

****

**Lab01f**

**Bài 1 quản lý điện thoại**

**Code**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167 | #include <stdio.h>  #include <string.h>  #define MAX\_DIENTHOAI 100  #define MAX\_MA 10  #define MAX\_NHANHIEU 20  // Cấu trúc lưu thông tin một điện thoại  typedef struct {  char ma[MAX\_MA];  char nhanHieu[MAX\_NHANHIEU];  int gia;  } DienThoai;  // Danh sách điện thoại  typedef struct {  DienThoai ds[MAX\_DIENTHOAI];  int soLuong;  } DanhSachDienThoai;  // Hàm tìm vị trí của điện thoại theo mã  int TimViTri(DanhSachDienThoai \*ds, const char \*ma) {  for (int i = 0; i < ds->soLuong; i++) {  if (strcmp(ds->ds[i].ma, ma) == 0) {  return i;  }  }  return -1;  }  // Thêm một điện thoại vào danh sách  void ThemDienThoai(DanhSachDienThoai \*ds) {  if (ds->soLuong >= MAX\_DIENTHOAI) {  printf("Danh sách đã đầy!\n");  return;  }  DienThoai dt;  printf("Nhập mã điện thoại: ");  scanf("%s", dt.ma);  // Kiểm tra xem điện thoại đã tồn tại chưa  if (TimViTri(ds, dt.ma) != -1) {  printf("Điện thoại đã tồn tại!\n");  return;  }  printf("Nhập nhãn hiệu: ");  scanf("%s", dt.nhanHieu);  printf("Nhập giá: ");  scanf("%d", &dt.gia);  // Thêm vào danh sách  ds->ds[ds->soLuong++] = dt;  printf("Đã thêm điện thoại thành công!\n");  }  // Tìm kiếm điện thoại theo mã và in thông tin  void TimDienThoai(DanhSachDienThoai \*ds) {  char ma[MAX\_MA];  printf("Nhập mã điện thoại cần tìm: ");  scanf("%s", ma);  int viTri = TimViTri(ds, ma);  if (viTri == -1) {  printf("Không tìm thấy điện thoại!\n");  } else {  printf("Nhãn hiệu: %s, Giá: %d\n", ds->ds[viTri].nhanHieu, ds->ds[viTri].gia);  }  }  // Cập nhật giá điện thoại theo mã  void CapNhatGia(DanhSachDienThoai \*ds) {  char ma[MAX\_MA];  printf("Nhập mã điện thoại cần cập nhật giá: ");  scanf("%s", ma);  int viTri = TimViTri(ds, ma);  if (viTri == -1) {  printf("Không tìm thấy điện thoại!\n");  } else {  printf("Nhập giá mới: ");  scanf("%d", &ds->ds[viTri].gia);  printf("Đã cập nhật giá thành công!\n");  }  }  // Xóa điện thoại theo mã  void XoaDienThoai(DanhSachDienThoai \*ds) {  char ma[MAX\_MA];  printf("Nhập mã điện thoại cần xóa: ");  scanf("%s", ma);  int viTri = TimViTri(ds, ma);  if (viTri == -1) {  printf("Không tìm thấy điện thoại!\n");  } else {  // Dịch các phần tử về trước  for (int i = viTri; i < ds->soLuong - 1; i++) {  ds->ds[i] = ds->ds[i + 1];  }  ds->soLuong--;  printf("Đã xóa điện thoại thành công!\n");  }  }  // Xuất danh sách điện thoại  void XuatDanhSach(DanhSachDienThoai \*ds) {  if (ds->soLuong == 0) {  printf("Danh sách rỗng!\n");  return;  }  printf("Danh sách điện thoại:\n");  for (int i = 0; i < ds->soLuong; i++) {  printf("%s - %s - %d\n", ds->ds[i].ma, ds->ds[i].nhanHieu, ds->ds[i].gia);  }  }  // Hiển thị menu  void Menu() {  printf("\n===== QUẢN LÝ ĐIỆN THOẠI =====\n");  printf("1. Thêm điện thoại\n");  printf("2. Tìm kiếm điện thoại\n");  printf("3. Cập nhật giá điện thoại\n");  printf("4. Xóa điện thoại\n");  printf("5. Xuất danh sách điện thoại\n");  printf("0. Thoát\n");  printf("================================\n");  }  int main() {  DanhSachDienThoai ds;  ds.soLuong = 0;    int luaChon;  do {  Menu();  printf("Nhập lựa chọn: ");  scanf("%d", &luaChon);  switch (luaChon) {  case 1:  ThemDienThoai(&ds);  break;  case 2:  TimDienThoai(&ds);  break;  case 3:  CapNhatGia(&ds);  break;  case 4:  XoaDienThoai(&ds);  break;  case 5:  XuatDanhSach(&ds);  break;  case 0:  printf("Thoát chương trình!\n");  break;  default:  printf("Lựa chọn không hợp lệ! Vui lòng nhập lại.\n");  }  } while (luaChon != 0);  return 0;  } |

**\*mã nguồn**

****

**Bài 2 cửa hàng tạp hoá**

**Code**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115 | #include <stdio.h>  #include <string.h>  #define MAX\_MATHANG 100  #define MAX\_MSMH 10  #define MAX\_TENMH 30  // Cấu trúc mặt hàng  typedef struct {  char msmh[MAX\_MSMH]; // Mã số mặt hàng  char tenMH[MAX\_TENMH]; // Tên mặt hàng  int soLuong; // Số lượng mặt hàng  int donGia; // Đơn giá  } MatHang;  // Danh sách mặt hàng  typedef struct {  MatHang ds[MAX\_MATHANG]; // Mảng các mặt hàng  int soLuong; // Số lượng mặt hàng hiện có  } CuaHang;  // Hàm nhập danh sách mặt hàng  void NhapMatHang(CuaHang \*ch) {  if (ch->soLuong >= MAX\_MATHANG) {  printf("Danh sách mặt hàng đã đầy!\n");  return;  }  MatHang mh;  printf("Nhập mã số mặt hàng: ");  scanf("%s", mh.msmh);  printf("Nhập tên mặt hàng: ");  scanf("%s", mh.tenMH);  printf("Nhập số lượng: ");  scanf("%d", &mh.soLuong);  printf("Nhập đơn giá: ");  scanf("%d", &mh.donGia);  // Thêm mặt hàng vào danh sách  ch->ds[ch->soLuong++] = mh;  printf("Mặt hàng đã được thêm thành công!\n");  }  // Hàm tìm mặt hàng theo tên  void TimMatHangTheoTen(CuaHang \*ch) {  char tenMH[MAX\_TENMH];  printf("Nhập tên mặt hàng cần tìm: ");  scanf("%s", tenMH);  int found = 0;  for (int i = 0; i < ch->soLuong; i++) {  if (strcmp(ch->ds[i].tenMH, tenMH) == 0) {  printf("Mã số mặt hàng: %s\n", ch->ds[i].msmh);  printf("Tên mặt hàng: %s\n", ch->ds[i].tenMH);  printf("Số lượng: %d\n", ch->ds[i].soLuong);  printf("Đơn giá: %d\n", ch->ds[i].donGia);  found = 1;  break;  }  }  if (!found) {  printf("Không có mặt hàng đang tìm.\n");  }  }  // Hàm tính tổng số lượng các mặt hàng  void TinhTongSoLuong(CuaHang \*ch) {  int tongSoLuong = 0;  for (int i = 0; i < ch->soLuong; i++) {  tongSoLuong += ch->ds[i].soLuong;  }  printf("Tổng số lượng các mặt hàng trong cửa hàng: %d\n", tongSoLuong);  }  // Hàm hiển thị menu  void Menu() {  printf("\n===== CỬA HÀNG TẠP HÓA =====\n");  printf("1. Nhập mặt hàng\n");  printf("2. Tìm mặt hàng theo tên\n");  printf("3. Tính tổng số lượng các mặt hàng\n");  printf("0. Thoát\n");  printf("============================\n");  }  int main() {  CuaHang ch;  ch.soLuong = 0; // Khởi tạo số lượng mặt hàng ban đầu  int luaChon;  do {  Menu();  printf("Nhập lựa chọn: ");  scanf("%d", &luaChon);  switch (luaChon) {  case 1:  NhapMatHang(&ch);  break;  case 2:  TimMatHangTheoTen(&ch);  break;  case 3:  TinhTongSoLuong(&ch);  break;  case 0:  printf("Thoát chương trình!\n");  break;  default:  printf("Lựa chọn không hợp lệ! Vui lòng nhập lại.\n");  }  } while (luaChon != 0);  return 0;  } |

**\*mã nguồn**

****