A. BÀI TẬP MINH HỌA

Bài 1. Khai báo cấu trúc phân số (PhanSo) gồm thông tin: tử số (tuSo) và mẫu số (mauSo). Xây dựng

các phương thức nhập, xuất, so sánh, rút gọn, phép toán cộng phân số. Viết chương trình nhập vào hai

phân số và in ra phân số sau khi rút gọn, kết quả phép so sánh, toán cộng hai phân số.

Moi ban nhap vao phan so 1: 2 6

Moi ban nhap vao phan so 2: 2 4

+ Rut gon: [1/3] [1/2]

+ So sanh: [1/3]<[1/2]

+ Cong: [1/3]+[1/2]=[5/6]



#include <stdio.h>

#include <math.h>

typedef struct

{

    int tuSo;

    int mauSo;

} PhanSo;

void NhapPhanSo(const char \*info, PhanSo \*ps)

{

    do

    {

        printf("%s", info);

        scanf("%d%d", &ps->tuSo, &ps->mauSo);

        if (ps->mauSo == 0)

        {

            printf("Mau so phai khac 0. Moi nhap lai!\n");

        }

    } while (ps->mauSo == 0);

}

void XuatPhanSo(PhanSo ps)

{

    if (ps.mauSo == 1)

    {

        printf("[%d]", ps.tuSo);

    }

    else

    {

        printf("[%d/%d]", ps.tuSo, ps.mauSo);

    }

}

int TimUCLN(int a, int b)

{

    while (b != 0)

    {

        int temp = b;

        b = a % b;

        a = temp;

    }

    return abs(a);

}

void RutGonPhanSo(PhanSo \*ps)

{

    int ucln = TimUCLN(ps->tuSo, ps->mauSo);

    ps->tuSo /= ucln;

    ps->mauSo /= ucln;

    if (ps->mauSo < 0)

    { // Đảm bảo mẫu số dương

        ps->tuSo = -ps->tuSo;

        ps->mauSo = -ps->mauSo;

    }

}

int SoSanhPhanSo(PhanSo p1, PhanSo p2)

{

    int vp1 = p1.tuSo \* p2.mauSo;

    int vp2 = p2.tuSo \* p1.mauSo;

    if (vp1 > vp2)

        return 1;

    if (vp1 == vp2)

        return 0;

    return -1;

}

void CongPhanSo(PhanSo p1, PhanSo p2, PhanSo \*pret)

{

    pret->tuSo = p1.tuSo \* p2.mauSo + p1.mauSo \* p2.tuSo;

    pret->mauSo = p1.mauSo \* p2.mauSo;

    RutGonPhanSo(pret);

}

int main()

{

    PhanSo p1, p2, pret;

    NhapPhanSo("Moi ban nhap phan so 1: ", &p1);

    NhapPhanSo("Moi ban nhap phan so 2: ", &p2);

    RutGonPhanSo(&p1);

    RutGonPhanSo(&p2);

    printf("+ Rut gon: ");

    XuatPhanSo(p1);

    printf(" ");

    XuatPhanSo(p2);

    printf("\n");

    char aDau[3] = {'<', '=', '>'};

    int ret = SoSanhPhanSo(p1, p2);

    printf("+ So sanh: ");

    XuatPhanSo(p1);

    printf(" %c ", aDau[ret + 1]);

    XuatPhanSo(p2);

    printf("\n");

    CongPhanSo(p1, p2, &pret);

    printf("+ Cong: ");

    XuatPhanSo(p1);

    printf(" + ");

    XuatPhanSo(p2);

    printf(" = ");

    XuatPhanSo(pret);

    printf("\n");

    return 0;

}

Bài 2. Cho thông tin một quyển sách gồm: Mã sách (tối đa 10 ký tự), Tên sách (tối đa 20 ký tự), Giá

(số nguyên). Viết các hàm nhập n quyển sách và tìm quyển sách có mã cho trước.



#include <stdio.h>

#include <string.h>

#define MAX 100

typedef struct SachTag

{

    char maSach[11];

    char tenSach[21];

    int gia;

} Sach;

void Nhap1QuyenSach(Sach \*s)

{

    printf("Ma sach: ");

    fgets(s->maSach, sizeof(s->maSach), stdin);

    s->maSach[strcspn(s->maSach, "\n")] = '\0'; // Xóa ký tự xuống dòng

    printf("Ten sach: ");

    fgets(s->tenSach, sizeof(s->tenSach), stdin);

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

    printf("Gia: ");

    scanf("%d", &s->gia);

    while (getchar() != '\n')

        ; // Xóa bộ đệm đầu vào

}

void Xuat1QuyenSach(Sach s)

{

    printf("Ma sach: %s \n", s.maSach);

    printf("Ten sach: %s \n", s.tenSach);

    printf("Gia: %d \n", s.gia);

}

void NhapMangSach(Sach a[], int \*n)

{

    printf("Nhap so luong sach: ");

    scanf("%d", n);

    getchar(); // Xóa ký tự '\n' còn sót lại

    for (int i = 0; i < \*n; i++)

    {

        printf("Nhap quyen sach thu %d: \n", i);

        Nhap1QuyenSach(&a[i]);

    }

}

int TimSachTheoMa(Sach a[], int n, char ma[], Sach \*kq)

{

    for (int i = 0; i < n; i++)

    {

        if (strcmp(ma, a[i].maSach) == 0)

        {

            \*kq = a[i];

            return 1;

        }

    }

    return 0;

}

int main(void)

{

    Sach a[MAX];

    int n;

    NhapMangSach(a, &n);

    char ma[11];

    printf("Nhap ma sach muon tim: ");

    fgets(ma, sizeof(ma), stdin);

    ma[strcspn(ma, "\n")] = '\0';

    Sach kq;

    if (TimSachTheoMa(a, n, ma, &kq))

    {

        printf("Quyen sach tim duoc la: \n");

        Xuat1QuyenSach(kq);

    }

    else

    {

        printf("Khong tim thay quyen sach co ma da cho\n");

    }

    return 0;

}

Bài 3. Một lớp học có sĩ số tối đa 50 học sinh, mỗi học sinh (HocSinh) gồm các thông tin sau: mã số

(MaSo) – chuỗi tối đa 10 ký tự, họ lót (Ho) – chuỗi tối đa 10 ký tự, tên (Ten) – chuỗi tối đa 50 ký tự,

Phái (Phai) – số nguyên có giá trị 0 cho Nữ và 1 cho Nam, năm sinh (NamSinh) – số nguyên dương

lớn hơn 0. Lớp học (LopHoc) bao gồm mảng các học sinh tối đa 50 phần tử (aHocSinh) và số lượng

học sinh hiện có (SoLuong). Hãy cho biết sĩ số nam, nữ và in ra danh sách học sinh tăng dần theo năm

sinh.

Yêu cầu: Viết chương trình thực hiện yêu cầu đề bài, gồm các phần sau:

a) Khai báo cấu trúc HocSinh và LopHoc với thông tin mô tả phía trên.

b) Hàm nhập danh sách các học sinh trong lớp học có ràng buộc dữ liệu.

void NhapLopHoc(LopHoc &lop) {...}

c) Hàm xuất danh sách ra màn hình.

void XuatLopHoc(LopHoc lop) {...}

d) Hàm đếm sĩ số học sinh trong lớp học.

void DemSiSo(LopHoc lop, int &sonam, int &sonu) {...}

e) Hàm sắp xếp danh sách tăng dần theo năm sinh.

void SapXepTangTheoNamSinh(LopHoc &lop) {...}

f) Hàm main sử dụng các hàm trên xử lý các yêu cầu đề bài.

int main() {...}



#include <stdio.h>

#include <string.h>

#define MAX 50

struct HocSinh

{

    char MaSo[11], Ho[11], Ten[51];

    int Phai, NamSinh;

};

struct LopHoc

{

    HocSinh aHocSinh[MAX];

    int SoLuong;

};

void NhapLopHoc(LopHoc \*lop)

{

    printf("Nhap so luong hoc sinh: ");

    scanf("%d", &lop->SoLuong);

    while (lop->SoLuong <= 0 || lop->SoLuong > MAX)

    {

        printf("Nhap sai! Nhap lai: ");

        scanf("%d", &lop->SoLuong);

    }

    getchar(); // Xóa ký tự '\n' sau scanf

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

    {

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

        printf("+ Ma so: ");

        fgets(lop->aHocSinh[i].MaSo, sizeof(lop->aHocSinh[i].MaSo), stdin);

        lop->aHocSinh[i].MaSo[strcspn(lop->aHocSinh[i].MaSo, "\n")] = '\0';

        printf("+ Ho lot: ");

        fgets(lop->aHocSinh[i].Ho, sizeof(lop->aHocSinh[i].Ho), stdin);

        lop->aHocSinh[i].Ho[strcspn(lop->aHocSinh[i].Ho, "\n")] = '\0';

        printf("+ Ten: ");

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

        lop->aHocSinh[i].Ten[strcspn(lop->aHocSinh[i].Ten, "\n")] = '\0';

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

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

        getchar(); // Xóa bộ đệm nhập

        while (lop->aHocSinh[i].Phai != 0 && lop->aHocSinh[i].Phai != 1)

        {

            printf("+ Nhap sai! Nhap lai: ");

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

            getchar();

        }

        printf("+ Nam sinh: ");

        scanf("%d", &lop->aHocSinh[i].NamSinh);

        getchar();

        while (lop->aHocSinh[i].NamSinh <= 0)

        {

            printf("+ Nhap sai! Nhap lai: ");

            scanf("%d", &lop->aHocSinh[i].NamSinh);

            getchar();

        }

    }

}

void XuatLopHoc(LopHoc lop)

{

    printf("Lop hoc co %d hoc sinh: \n", lop.SoLuong);

    printf("STT\tMa so\tHo ten\t\tPhai\tNam sinh \n");

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

    {

        printf("%d\t%s\t%s %s\t", i + 1, lop.aHocSinh[i].MaSo, lop.aHocSinh[i].Ho, lop.aHocSinh[i].Ten);

        printf("%s\t%d\n", (lop.aHocSinh[i].Phai == 1) ? "Nam" : "Nu", lop.aHocSinh[i].NamSinh);

    }

}

void DemSiSo(LopHoc lop, int \*sonam, int \*sonu)

{

    \*sonam = \*sonu = 0;

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

    {

        if (lop.aHocSinh[i].Phai == 1)

            (\*sonam)++;

        else

            (\*sonu)++;

    }

}

void SapXepTangTheoNamSinh(LopHoc \*lop)

{

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

    {

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

        {

            if (lop->aHocSinh[i].NamSinh > lop->aHocSinh[j].NamSinh)

            {

                HocSinh tmp = lop->aHocSinh[i];

                lop->aHocSinh[i] = lop->aHocSinh[j];

                lop->aHocSinh[j] = tmp;

            }

        }

    }

}

int main()

{

    LopHoc lop = {

        {{"01", "Nguyen Van", "A", 1, 1984},

         {"02", "Tran Thi", "B", 0, 1982},

         {"03", "Le Duy", "C", 1, 1982}},

        3};

    int sonam, sonu;

    // NhapLopHoc(&lop);

    XuatLopHoc(lop);

    DemSiSo(lop, &sonam, &sonu);

    printf("+ Dem si so hoc sinh trong lop hoc: \n");

    printf("Lop co %d hoc sinh, trong do %d nam va %d nu.\n", lop.SoLuong, sonam, sonu);

    printf("+ Danh sach hoc sinh sau khi sap xep theo nam sinh:\n");

    SapXepTangTheoNamSinh(&lop);

    XuatLopHoc(lop);

    return 0;

}

B. BÀI TẬP PHẢI LÀM

Bài 1. Một phòng ban có tối đa 100 nhân viên, mỗi nhân viên (NhanVien) gồm các thông tin sau: mã

số (MaSo) – chuỗi tối đa 10 ký tự, họ lót (Ho) – chuỗi tối đa 10 ký tự, tên (Ten) – chuỗi tối đa 50 ký

tự, Phái (Phai) – số nguyên có giá trị 0 cho Nữ và 1 cho Nam, số năm làm việc (ThamNien) – số

nguyên dương lớn hơn hay bằng 0. Phòng ban (PhongBan) bao gồm mảng tối đa 100 nhân viên

(aNhanVien) và số lượng nhân viên hiện có (SoLuong). Hãy cho biết sĩ số nam, nữ và in ra danh sách

nhân viên tăng dần theo thâm niên.

Yêu cầu: Viết chương trình thực hiện yêu cầu đề bài, gồm các phần sau:

a) Khai báo cấu trúc NhanVien và PhongBan với thông tin mô tả phía trên.

b) Hàm nhập danh sách các nhân viên cho phòng ban có ràng buộc dữ liệu.

void NhapPhongBan(PhongBan &pb) {...}

c) Hàm xuất thông tin phòng ban ra màn hình.

void XuatPhongBan(PhongBan pb) {...}

d) Hàm đếm sĩ số nhân viên trong phòng ban.

void DemSiSo(PhongBan ds, int &sonam, int &sonu) {...}

e) Hàm sắp xếp danh sách nhân viên tăng dần theo thâm niên.

void SapXepTangTheoThamNien(PhongBan &pb) {...}

f) Hàm main sử dụng các hàm trên xử lý các yêu cầu đề bài.

int main() {...}



#include <stdio.h>

#include <string.h>

#define MAX 100

// Khai báo cấu trúc nhân viên

typedef struct

{

    char MaSo[11];

    char Ho[11];

    char Ten[51];

    int Phai; // 0: Nữ, 1: Nam

    int ThamNien;

} NhanVien;

// Khai báo cấu trúc phòng ban

typedef struct

{

    NhanVien aNhanVien[MAX];

    int SoLuong;

} PhongBan;

// Hàm nhập danh sách nhân viên

void NhapPhongBan(PhongBan \*pb)

{

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

    scanf("%d", &pb->SoLuong);

    while (pb->SoLuong <= 0 || pb->SoLuong > MAX)

    {

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

        scanf("%d", &pb->SoLuong);

    }

    getchar(); // Xóa bộ nhớ đệm

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

    {

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

        printf("Ma so: ");

        fgets(pb->aNhanVien[i].MaSo, 11, stdin);

        strtok(pb->aNhanVien[i].MaSo, "\n");

        printf("Ho lot: ");

        fgets(pb->aNhanVien[i].Ho, 11, stdin);

        strtok(pb->aNhanVien[i].Ho, "\n");

        printf("Ten: ");

        fgets(pb->aNhanVien[i].Ten, 51, stdin);

        strtok(pb->aNhanVien[i].Ten, "\n");

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

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

        while (pb->aNhanVien[i].Phai != 0 && pb->aNhanVien[i].Phai != 1)

        {

            printf("Nhap sai! Moi nhap lai (0 - Nu, 1 - Nam): ");

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

        }

        printf("Tham nien: ");

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

        while (pb->aNhanVien[i].ThamNien < 0)

        {

            printf("Nhap sai! Moi nhap lai: ");

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

        }

        getchar(); // Xóa bộ nhớ đệm

    }

}

// Hàm xuất danh sách nhân viên

void XuatPhongBan(PhongBan pb)

{

    printf("\nDanh sach nhan vien trong phong ban:\n");

    printf("%-10s %-10s %-15s %-5s %-10s\n", "MaSo", "Ho", "Ten", "Phai", "ThamNien");

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

    {

        printf("%-10s %-10s %-15s %-5s %-10d\n",

               pb.aNhanVien[i].MaSo,

               pb.aNhanVien[i].Ho,

               pb.aNhanVien[i].Ten,

               pb.aNhanVien[i].Phai == 1 ? "Nam" : "Nu",

               pb.aNhanVien[i].ThamNien);

    }

}

// Hàm đếm sĩ số nhân viên nam và nữ

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

{

    \*sonam = 0;

    \*sonu = 0;

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

    {

        if (pb.aNhanVien[i].Phai == 1)

        {

            (\*sonam)++;

        }

        else

        {

            (\*sonu)++;

        }

    }

}

// Hàm sắp xếp danh sách nhân viên theo thâm niên tăng dần

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;

            }

        }

    }

}

// Hàm main

int main()

{

    PhongBan pb;

    int sonam, sonu;

    NhapPhongBan(&pb);

    XuatPhongBan(pb);

    DemSiSo(pb, &sonam, &sonu);

    printf("\nPhong ban co %d nam va %d nu.\n", sonam, sonu);

    printf("\nDanh sach nhan vien sau khi sap xep tang theo tham nien:\n");

    SapXepTangTheoThamNien(&pb);

    XuatPhongBan(pb);

    return 0;

}

Bài 2. Một cửa hàng bán hoa có tối đa 20 loại hoa, mỗi loại hoa gồm các thông tin sau: Tên loại (Ten),

Số lượng (SoLuong), Đơn vị tính (DVT), Đơn giá (DonGia). Với tên loại, số lượng mà khách hàng yêu

cầu, hãy cho biết loại đó có hay không. Nếu có, có đủ số lượng bán hay không và tổng tiền nếu đủ số

lượng bán.

Yêu cầu: Viết chương trình thực hiện yêu cầu đề bài, gồm các phần sau:

a) Khai báo cấu trúc LoaiHoa và DanhSachLoaiHoa với thông tin mô tả phía trên.

b) Hàm nhập danh sách các loại hoa.

void NhapDanhSach(DanhSachLoaiHoa &ds) {...}

c) Hàm xuất danh sách ra màn hình.

void XuatDanhSach(DanhSachLoaiHoa ds) {...}

d) Hàm tìm loại hoa theo tên loại.

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

Trả về vị trí loại hoa trong danh sách, nếu không có trả về -1.

e) Hàm xử lý bán hoa cho khách hàng biết trước tên loại và số lượng.

void XuLyBanHoa(DanhSachLoaiHoa &ds,char \*tenloai, int soluong) {...}

Chúc năng: Kiểm tra hoa có tồn tại không. Nếu có, kiểm tra xem số lượng có đủ không. Nếu đủ, in

giá tiền, biết giá tiền bằng số lượng khách hàng mua \* đơn giá.

f) Hàm main sử dụng các hàm trên xử lý các yêu cầu đề bài.

void main() {...}



#include <stdio.h>

#include <string.h>

#define MAX 20

// Khai báo cấu trúc loại hoa

typedef struct

{

    char Ten[50];

    int SoLuong;

    char DVT[10];

    float DonGia;

} LoaiHoa;

// Khai báo danh sách loại hoa

typedef struct

{

    LoaiHoa dsHoa[MAX];

    int SoLuongHoa;

} DanhSachLoaiHoa;

// Hàm nhập danh sách loại hoa

void NhapDanhSach(DanhSachLoaiHoa \*ds)

{

    printf("Nhap so luong loai hoa: ");

    scanf("%d", &ds->SoLuongHoa);

    while (ds->SoLuongHoa <= 0 || ds->SoLuongHoa > MAX)

    {

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

        scanf("%d", &ds->SoLuongHoa);

    }

    getchar(); // Xóa bộ nhớ đệm

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

    {

        printf("\nNhap thong tin loai hoa thu %d:\n", i + 1);

        printf("Ten loai: ");

        fgets(ds->dsHoa[i].Ten, 50, stdin);

        strtok(ds->dsHoa[i].Ten, "\n");

        printf("So luong: ");

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

        printf("Don vi tinh: ");

        scanf("%s", ds->dsHoa[i].DVT);

        printf("Don gia: ");

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

        getchar();

    }

}

// Hàm xuất danh sách loại hoa

void XuatDanhSach(DanhSachLoaiHoa ds)

{

    printf("\nDanh sach loai hoa trong cua hang:\n");

    printf("%-20s %-10s %-10s %-10s\n", "Ten loai", "So luong", "DVT", "Don gia");

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

    {

        printf("%-20s %-10d %-10s %-10.2f\n",

               ds.dsHoa[i].Ten,

               ds.dsHoa[i].SoLuong,

               ds.dsHoa[i].DVT,

               ds.dsHoa[i].DonGia);

    }

}

// Hàm tìm loại hoa theo tên

int TimLoaiHoa(DanhSachLoaiHoa ds, char \*tenloai)

{

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

    {

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

        {

            return i;

        }

    }

    return -1;

}

// Hàm xử lý bán hoa

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

{

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

    if (index == -1)

    {

        printf("\nLoai hoa %s khong co trong cua hang.\n", tenloai);

    }

    else

    {

        if (ds->dsHoa[index].SoLuong >= soluong)

        {

            float tongTien = soluong \* ds->dsHoa[index].DonGia;

            ds->dsHoa[index].SoLuong -= soluong;

            printf("\nBan thanh cong %d %s, tong tien: %.2f\n", soluong, ds->dsHoa[index].DVT, tongTien);

        }

        else

        {

            printf("\nKhong du so luong %s de ban!\n", tenloai);

        }

    }

}

// Hàm main

int main()

{

    DanhSachLoaiHoa ds;

    NhapDanhSach(&ds);

    XuatDanhSach(ds);

    char tenHoa[50];

    int soLuongMua;

    printf("\nNhap ten loai hoa can mua: ");

    fgets(tenHoa, 50, stdin);

    strtok(tenHoa, "\n");

    printf("Nhap so luong can mua: ");

    scanf("%d", &soLuongMua);

    XuLyBanHoa(&ds, tenHoa, soLuongMua);

    XuatDanhSach(ds);

    return 0;

}