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

**Lab1c**

Tên : Trịnh Tâm Như

Mssv: 3123411215

**Code**

Bài 1

#include <iostream>

#include <string>

#include <algorithm> // Để sử dụng std::sort

using namespace std;

#define MAX\_NV 100 // Tối đa 100 nhân viên

// Khai báo cấu trúc NhanVien

struct NhanVien {

char MaSo[11]; // Mã số (tối đa 10 ký tự)

char Ho[11]; // Họ lót (tối đa 10 ký tự)

char Ten[51]; // Tên (tối đa 50 ký tự)

int Phai; // 0: Nữ, 1: Nam

int ThamNien; // Số năm làm việc

};

// Khai báo cấu trúc PhongBan

struct PhongBan {

NhanVien aNhanVien[MAX\_NV]; // Mảng nhân viên

int SoLuong; // Số lượng nhân viên hiện có

};

// Hàm nhập thông tin cho một nhân viên

void NhapNhanVien(NhanVien& nv) {

cout << "Nhap ma so: ";

cin >> nv.MaSo;

cout << "Nhap ho: ";

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

cin.getline(nv.Ho, sizeof(nv.Ho));

cout << "Nhap ten: ";

cin.getline(nv.Ten, sizeof(nv.Ten));

cout << "Nhap phai (0: Nu, 1: Nam): ";

cin >> nv.Phai;

cout << "Nhap tham nien: ";

cin >> nv.ThamNien;

}

// Hàm nhập thông tin cho phòng ban

void NhapPhongBan(PhongBan& pb) {

cout << "Nhap so luong nhan vien: ";

cin >> pb.SoLuong;

// Ràng buộc dữ liệu

if (pb.SoLuong < 0 || pb.SoLuong > MAX\_NV) {

cout << "So luong nhan vien khong hop le!" << endl;

return;

}

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

cout << "Nhap thong tin nhan vien " << i + 1 << ":\n";

NhapNhanVien(pb.aNhanVien[i]);

}

}

// Hàm xuất thông tin phòng ban

void XuatPhongBan(PhongBan pb) {

cout << "\nDanh sach nhan vien:\n";

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

cout << "Ma so: " << pb.aNhanVien[i].MaSo << endl;

cout << "Ho: " << pb.aNhanVien[i].Ho << endl;

cout << "Ten: " << pb.aNhanVien[i].Ten << endl;

cout << "Phai: " << (pb.aNhanVien[i].Phai == 1 ? "Nam" : "Nu") << endl;

cout << "Tham nien: " << pb.aNhanVien[i].ThamNien << endl;

cout << "-------------------------------\n";

}

}

// Hàm đếm sĩ số nam, nữ trong phòng ban

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

sonam = 0;

sonu = 0;

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

if (ds.aNhanVien[i].Phai == 1) {

sonam++;

}

else {

sonu++;

}

}

cout << "Si so nam: " << sonam << endl;

cout << "Si so nu: " << sonu << endl;

}

// Hàm so sánh để sắp xếp nhân viên theo thâm niên (tăng dần)

bool soSanhThamNien(NhanVien a, NhanVien b) {

return a.ThamNien < b.ThamNien;

}

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

void SapXepTangTheoThamNien(PhongBan& pb) {

sort(pb.aNhanVien, pb.aNhanVien + pb.SoLuong, soSanhThamNien);

}

int main() {

PhongBan pb;

int sonam, sonu;

// Nhập thông tin phòng ban

NhapPhongBan(pb);

// Xuất thông tin phòng ban

XuatPhongBan(pb);

// Đếm sĩ số nam, nữ

DemSiSo(pb, sonam, sonu);

// Sắp xếp danh sách nhân viên theo thâm niên

SapXepTangTheoThamNien(pb);

// Xuất lại danh sách nhân viên sau khi sắp xếp

cout << "\nDanh sach nhan vien sau khi sap xep theo tham nien:\n";

XuatPhongBan(pb);

return 0;

}

**TEST CASE**

**DỮ LIỆU ĐẦU VÀO**

Nhap so luong nhan vien: 3

Nhap thong tin nhan vien 1:

Nhap ma so: NV001

Nhap ho: Nguyen

Nhap ten: Anh

Nhap phai (0: Nu, 1: Nam): 1

Nhap tham nien: 5

Nhap thong tin nhan vien 2:

Nhap ma so: NV002

Nhap ho: Tran

Nhap ten: Bich

Nhap phai (0: Nu, 1: Nam): 0

Nhap tham nien: 3

Nhap thong tin nhan vien 3:

Nhap ma so: NV003

Nhap ho: Le

Nhap ten: Minh

Nhap phai (0: Nu, 1: Nam): 1

Nhap tham nien: 7

KẾT QUẢ ĐẦU RA

Danh sach nhan vien:

Ma so: NV001

Ho: Nguyen

Ten: Anh

Phai: Nam

Tham nien: 5

-------------------------------

Ma so: NV002

Ho: Tran

Ten: Bich

Phai: Nu

Tham nien: 3

-------------------------------

Ma so: NV003

Ho: Le

Ten: Minh

Phai: Nam

Tham nien: 7

-------------------------------

Si so nam: 2

Si so nu: 1

Danh sach nhan vien sau khi sap xep theo tham nien:

Ma so: NV002

Ho: Tran

Ten: Bich

Phai: Nu

Tham nien: 3

-------------------------------

Ma so: NV001

Ho: Nguyen

Ten: Anh

Phai: Nam

Tham nien: 5

-------------------------------

Ma so: NV003

Ho: Le

Ten: Minh

Phai: Nam

Tham nien: 7

-------------------------------

**Bài 2**

#include <iostream>

#include <cstring>

using namespace std;

#define MAX\_LOAI\_HOA 20 // Tối đa 20 loại hoa

// Khai báo cấu trúc LoaiHoa

struct LoaiHoa {

char Ten[31]; // Tên loại hoa (tối đa 30 ký tự)

int SoLuong; // Số lượng hoa có sẵn

char DVT[21]; // Đơn vị tính (tối đa 20 ký tự)

int DonGia; // Đơn giá

};

// Khai báo cấu trúc DanhSachLoaiHoa

struct DanhSachLoaiHoa {

LoaiHoa ds[MAX\_LOAI\_HOA]; // Mảng chứa các loại hoa

int SoLuong; // Số lượng loại hoa hiện có

};

// Hàm nhập danh sách các loại hoa

void NhapDanhSach(DanhSachLoaiHoa& ds) {

cout << "Nhap so loai hoa: ";

cin >> ds.SoLuong;

cin.ignore(); // Xóa ký tự '\n' còn lại trong bộ đệm sau khi nhập số

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

cout << "Nhap thong tin loai hoa thu " << i + 1 << ":\n";

cout << "Ten loai hoa: ";

cin.getline(ds.ds[i].Ten, 31);

cout << "So luong: ";

cin >> ds.ds[i].SoLuong;

cin.ignore(); // Xóa ký tự '\n' còn lại sau khi nhập số

cout << "Don vi tinh: ";

cin.getline(ds.ds[i].DVT, 21);

cout << "Don gia: ";

cin >> ds.ds[i].DonGia;

cin.ignore(); // Xóa ký tự '\n' còn lại sau khi nhập số

}

}

// Hàm xuất danh sách các loại hoa

void XuatDanhSach(DanhSachLoaiHoa ds) {

cout << "\nDanh sach loai hoa:\n";

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

cout << "Ten loai: " << ds.ds[i].Ten << endl;

cout << "So luong: " << ds.ds[i].SoLuong << endl;

cout << "Don vi tinh: " << ds.ds[i].DVT << endl;

cout << "Don gia: " << ds.ds[i].DonGia << endl;

cout << "----------------------------\n";

}

}

// Hàm tìm loại hoa theo tên loại

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

for (int i = 0; i < ds.SoLuong; 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(DanhSachLoaiHoa& ds, const char\* tenloai, int soluong) {

int viTri = TimLoaiHoa(ds, tenloai); // Tìm loại hoa theo tên

if (viTri == -1) {

cout << "Khong tim thay loai hoa co ten: " << tenloai << endl;

}

else {

LoaiHoa& loai = ds.ds[viTri];

if (loai.SoLuong >= soluong) {

// Nếu có đủ số lượng

loai.SoLuong -= soluong; // Giảm số lượng hoa đã bán

int tongTien = soluong \* loai.DonGia; // Tính tổng tiền

cout << "Hoa ban thanh cong! Tong tien: " << tongTien << " VND." << endl;

}

else {

// Nếu không đủ số lượng

cout << "Khong du so luong hoa, co " << loai.SoLuong << " hoa trong kho." << endl;

}

}

}

int main() {

DanhSachLoaiHoa ds;

char tenloai[31];

int soluong;

// Nhập danh sách các loại hoa

NhapDanhSach(ds);

// Xuất danh sách hoa

XuatDanhSach(ds);

// Xử lý bán hoa cho khách hàng

cout << "\nNhap ten loai hoa muon mua: ";

cin.getline(tenloai, 31);

cout << "Nhap so luong hoa muon mua: ";

cin >> soluong;

XuLyBanHoa(ds, tenloai, soluong);

return 0;

}

**TEST CASE**

**INPUT**

Nhap so loai hoa: 2

Nhap thong tin loai hoa thu 1:

Ten loai hoa: Hoa Hong

So luong: 50

Don vi tinh: Hoa

Don gia: 100000

Nhap thong tin loai hoa thu 2:

Ten loai hoa: Hoa Lan

So luong: 30

Don vi tinh: Hoa

Don gia: 150000

**OUTPUT**

Danh sach loai hoa:

Ten loai: Hoa Hong

So luong: 50

Don vi tinh: Hoa

Don gia: 100000

----------------------------

Ten loai: Hoa Lan

So luong: 30

Don vi tinh: Hoa

Don gia: 150000

----------------------------