CHƯƠNG TRÌNH QUẢN LÝ NHÂN VIÊN CỦA MỘT NHÀ HÀNG

Hàm in hoa chữ cái đầu tiên của tiên:

void inhoa(char name[]) {

        for(int i=0; i<strlen(name); i++){

            if(i==0|| (i>0 && name[i-1]==' ')){

                if(name[i]>='a' && name[i]<='z')

                    name[i]=name[i]-32;

            }else{

                if(name[i]>='A' && name[i]<='Z'){

                    name[i]=name[i]+32;

                }

            }

        }

    }

Hàm lấy chữ cái đầu trong tên để sắp xếp:

    char chudau(char name[]) {

        inhoa(name);

        for (int i = strlen(name) - 1; i >= 0; i--) {

            if (i == 0 || (i > 0 && isspace(name[i - 1]))) {

                return name[i];

            }

        }

        return '\0';

    }

Hàm đổi vị trí của 2 người trong danh sách:

    void doivitri(NhanVien& nv1, NhanVien& nv2) {

        NhanVien temp = nv1;

        nv1 = nv2;

        nv2 = temp;

    }

Hàm sắp xếp:

    void sapxep(NhanVien nvien[], int n) {

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

            if (chudau(nvien[i].name) > chudau(nvien[i + 1].name)) {

                doivitri(nvien[i], nvien[i + 1]);

            }

        }

    }

Hàm hiển thị thông tin của nhân viên:

    void hienThi(const NhanVien& nv) {

        char cname[100], gend[10], pos[100];

        strcpy(cname, nv.name);

        strcpy(gend, nv.gender);

        strcpy(pos, nv.position);

        inhoa(cname);

        inhoa(gend);

        inhoa(pos);

        printf("| ID: %5s | Name: %20s | Gender: %5s | Age: %3d | Phone Number: %11s | Position: %15s |",

        nv.id, cname, gend, nv.age, nv.phonenumber, pos);

        cout << "\n|-----------|----------------------------|---------------|----------|---------------------------|---------------------------|" << endl;

    }

Hàm nhập thông tin của nhân viên:

    void nhapThongTin(NhanVien& nv) {

        cout << "\nNhap ID: ";

        cin >> nv.id;

        cin.ignore();

        cout << "Nhap ten nhan vien: ";

        cin.getline(nv.name, 100);

        cout << "Nhap gioi tinh: ";

        cin >> nv.gender;

        cout << "Nhap tuoi: ";

        cin >> nv.age;

        cin.ignore();

        cout << "Nhap so dien thoai: ";

        cin >> nv.phonenumber;

        cin.ignore();

        cout << "Nhap chuc vu: ";

        cin.getline(nv.position, 100);

    }

Hàm nhập danh sách nhân viên:

    void nhapDS(NhanVien nvien[], int& n) {

        for (int i = 0; i < n; i++) {

            bool ktraID;

            do {

                ktraID = true;

                cout << "\nNhap thong tin nhan vien thu " << i + 1 << ": ";

                nhapThongTin(nvien[i]);

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

                    if (strcmp(nvien[j].id, nvien[i].id) == 0) {

                        cout << "\nID da ton tai, vui long nhap lai: ";

                        ktraID = false;

                        break;

                    }

                }

            } while (!ktraID);

        }

    }

Hàm hiển thị danh sách nhân viên:

void hienThiDS(NhanVien nvien[], int n) {

        cout << "\n------------- Danh Sach Nhan Vien ------------\n";

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

        sapxep(nvien, n);

        for (int i = 0; i < n; i++) {

            hienThi(nvien[i]);

        }

    }

Hàm thêm nhân viên:

    void themNV(NhanVien nvien[], int& n, int t) {

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

            bool ktraID;

            do {

                ktraID = true;

                cout << "\nNhap thong tin nhan vien thu " << i + 1 << ": ";

                nhapThongTin(nvien[i]);

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

                    if (strcmp(nvien[j].id, nvien[i].id) == 0) {

                        cout << "\nID da ton tai, vui long nhap lai: ";

                        ktraID = false;

                        break;

                    }

                }

            } while (!ktraID);

        }

        n += t;

    }

Hàm tìm kiếm nhân viên theo tên:

    void timNV(NhanVien nvien[], int n) {

        char ttin[100];

        bool found = false;

        cout << "\nNhap ten cua nhan vien muon tim kiem: ";

        cin.getline(ttin, 100);

        for (int i = 0; i < n; i++) {

            inhoa(ttin);

            if (strstr(nvien[i].name, ttin) != NULL) {

                found = true;

                hienThi(nvien[i]);

            }

        }

        if (!found) {

            cout << "\nKhong tim thay nhan vien can tim! ";

        }

    }

Hàm xóa nhân viên:

    void timNV(NhanVien nvien[], int n) {

        char ttin[100];

        bool found = false;

        cout << "\nNhap ten cua nhan vien muon tim kiem: ";

        cin.getline(ttin, 100);

        for (int i = 0; i < n; i++) {

            inhoa(ttin);

            if (strstr(nvien[i].name, ttin) != NULL) {

                found = true;

                hienThi(nvien[i]);

            }

        }

        if (!found) {

            cout << "\nKhong tim thay nhan vien can tim! ";

        }

    }

Hàm cập nhật thông tin cho nhân viên:

void capNhat(NhanVien nvien[], int n) {

        char ttin[100];

        bool found = false;

        cout << "\nNhap ten cua nhan vien muon cap nhat: ";

        cin.getline(ttin, 100);

        for (int i = 0; i < n; i++) {

            inhoa(ttin);

            if (strstr(nvien[i].name, ttin) != NULL) {

                found = true;

                cout << "\nNhap thong tin cap nhat cho nhan vien: \n";

                nhapThongTin(nvien[i]);

                break;

            }

        }

        if (found) {

            cout << "\nNhan vien da duoc cap nhat!";

        } else {

            cout << "\nKhong tim thay nhan vien can tim!";

        }

    }