**BÁO CÁO LAB – QUẢN LÝ SINH VIÊN BẰNG DANH SÁCH LIÊN KẾT**

**1. Đặt vấn đề**

Trong bài thực hành này, mục tiêu là xây dựng một chương trình quản lý sinh viên sử dụng danh sách liên kết đơn. Qua đó, sinh viên được làm quen với cách khai báo cấu trúc dữ liệu, thao tác thêm, duyệt, tách, sắp xếp danh sách, tính toán thống kê và lưu kết quả ra tệp tin.

**2. Nội dung chương trình**

**2.1. Cấu trúc dữ liệu**

typedef struct SinhVien {

char mssv[10];

char hoten[50];

float diem;

struct SinhVien\* next;

} SinhVien;

**2.2. Các chức năng chính**

* **Tạo node mới:** cấp phát bộ nhớ động và khởi tạo dữ liệu.
* **Thêm vào cuối danh sách.**
* **Nhập danh sách sinh viên từ bàn phím.**
* **Hiển thị danh sách.**
* **Tách danh sách gốc thành 2 danh sách: ĐẠT (điểm >= 5.0) và KHÔNG ĐẠT (điểm < 5.0).**
* **Đếm số lượng sinh viên trong danh sách.**
* **Sắp xếp danh sách theo điểm tăng dần.**
* **Ghi danh sách ra tệp tin dat.txt và khongdat.txt.**

**2.3. Mã nguồn đầy đủ**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Định nghĩa cấu trúc sinh viên

typedef struct SinhVien {

char mssv[10];

char hoten[50];

float diem;

struct SinhVien\* next;

} SinhVien;

// Tạo node mới

SinhVien\* taoNode(char\* mssv, char\* hoten, float diem) {

SinhVien\* sv = (SinhVien\*)malloc(sizeof(SinhVien));

strcpy(sv->mssv, mssv);

strcpy(sv->hoten, hoten);

sv->diem = diem;

sv->next = NULL;

return sv;

}

// Thêm node vào cuối danh sách

void themCuoi(SinhVien\*\* head, SinhVien\* sv) {

if (\*head == NULL) {

\*head = sv;

} else {

SinhVien\* temp = \*head;

while (temp->next != NULL) temp = temp->next;

temp->next = sv;

}

}

// Nhập danh sách sinh viên

void nhapDanhSach(SinhVien\*\* head) {

int n;

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

scanf("%d", &n);

getchar();

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

char mssv[10], hoten[50];

float diem;

printf("Nhap MSSV: ");

fgets(mssv, sizeof(mssv), stdin);

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

printf("Nhap ho ten: ");

fgets(hoten, sizeof(hoten), stdin);

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

printf("Nhap diem: ");

scanf("%f", &diem);

getchar();

SinhVien\* sv = taoNode(mssv, hoten, diem);

themCuoi(head, sv);

}

}

// Hiển thị danh sách

void hienThi(SinhVien\* head) {

while (head != NULL) {

printf("MSSV: %s | Ho ten: %s | Diem: %.2f\n", head->mssv, head->hoten, head->diem);

head = head->next;

}

}

// Tách danh sách theo điểm

void tachDanhSach(SinhVien\* goc, SinhVien\*\* dat, SinhVien\*\* khongdat) {

while (goc != NULL) {

SinhVien\* sv = taoNode(goc->mssv, goc->hoten, goc->diem);

if (sv->diem < 5.0) {

themCuoi(khongdat, sv);

} else {

themCuoi(dat, sv);

}

goc = goc->next;

}

}

// Đếm số lượng sinh viên

int demSoLuong(SinhVien\* head) {

int dem = 0;

while (head != NULL) {

dem++;

head = head->next;

}

return dem;

}

// Sắp xếp danh sách theo điểm

void sapXepTheoDiem(SinhVien\* head) {

SinhVien \*i, \*j;

for (i = head; i != NULL; i = i->next) {

for (j = i->next; j != NULL; j = j->next) {

if (i->diem > j->diem) {

float d = i->diem;

i->diem = j->diem;

j->diem = d;

char tmp[50];

strcpy(tmp, i->hoten);

strcpy(i->hoten, j->hoten);

strcpy(j->hoten, tmp);

strcpy(tmp, i->mssv);

strcpy(i->mssv, j->mssv);

strcpy(j->mssv, tmp);

}

}

}

}

// Ghi danh sách ra file

void ghiFile(SinhVien\* head, char\* filename) {

FILE\* f = fopen(filename, "w");

while (head != NULL) {

fprintf(f, "%s,%s,%.2f\n", head->mssv, head->hoten, head->diem);

head = head->next;

}

fclose(f);

}

// Hàm main

int main() {

SinhVien\* goc = NULL;

SinhVien\* dat = NULL;

SinhVien\* khongdat = NULL;

nhapDanhSach(&goc);

tachDanhSach(goc, &dat, &khongdat);

printf("\nDanh sach DAT:\n");

sapXepTheoDiem(dat);

hienThi(dat);

printf("\nDanh sach KHONG DAT:\n");

sapXepTheoDiem(khongdat);

hienThi(khongdat);

int so\_dat = demSoLuong(dat);

int so\_khongdat = demSoLuong(khongdat);

int tong = so\_dat + so\_khongdat;

if (tong > 0) {

float tyle\_dat = (float)so\_dat / tong \* 100.0;

float tyle\_khongdat = (float)so\_khongdat / tong \* 100.0;

printf("\nTy le dat: %.2f%%\n", tyle\_dat);

printf("Ty le khong dat: %.2f%%\n", tyle\_khongdat);

}

ghiFile(dat, "dat.txt");

ghiFile(khongdat, "khongdat.txt");

return 0;

}

**3. Kết quả chạy chương trình**

* Nhập vào danh sách sinh viên từ bàn phím.
* Danh sách được tách thành 2 nhóm: ĐẠT và KHÔNG ĐẠT.
* In danh sách mỗi nhóm đã được sắp xếp tăng dần theo điểm.
* In tỷ lệ phần trăm ĐẠT và KHÔNG ĐẠT.
* Ghi dữ liệu ra file dat.txt và khongdat.txt.

**4. Kết luận**

Bài thực hành giúp sinh viên nắm vững:

* Cách khai báo và sử dụng danh sách liên kết đơn.
* Kỹ năng duyệt, chèn, tách, sắp xếp và thao tác với file.
* Tư duy chia nhỏ bài toán và tổ chức mã lệnh rõ ràng.