**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**MÔN: KỸ THUẬT LẬP TRÌNH**

**ĐỀ TÀI: BÀI TẬP LỚN**

**Giảng viên: Ths. TRẦN PHONG NHÃ**

**Sinh viên thực hiện: HÌNH VĨNH PHÁT**

**Lớp: CQ.65.CNTT**

**Khóa: 65**

Tp. Hồ Chí Minh, tháng 5 năm 2025

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**MÔN: KỸ THUẬT LẬP TRÌNH**

**ĐỀ TÀI: BÀI TẬP LỚN**

**Giảng viên: Ths. TRẦN PHONG NHÃ**

**Sinh viên thực hiện: HÌNH VĨNH PHÁT**

**Lớp: CQ.65.CNTT**

**Khóa: 65**

Tp. Hồ Chí Minh, tháng 5 năm 2025

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc198916705)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN 2](#_Toc198916706)

[A. LÝ THUYẾT 3](#_Toc198916707)

[1 HÀM: 3](#_Toc198916708)

[1.1 khái niệm: 3](#_Toc198916709)

[1.2 Ví dụ: 3](#_Toc198916710)

[2 CON TRỎ: 6](#_Toc198916711)

[2.1 Khái niệm: 6](#_Toc198916712)

[2.2 Ví dụ 6](#_Toc198916713)

[3 CON TRỎ MẢNG: 8](#_Toc198916714)

[3.1 Khái niệm: 8](#_Toc198916715)

[3.2 Ví dụ 8](#_Toc198916716)

[4 MẢNG CON TRỎ: 9](#_Toc198916717)

[4.1 Khái niệm 9](#_Toc198916718)

[4.2 Ví dụ 9](#_Toc198916719)

[5 CON TRỎ HÀM: 10](#_Toc198916720)

[5.1 Khái niệm 10](#_Toc198916721)

[5.2 Ví dụ 10](#_Toc198916722)

[6 CẤP PHÁT ĐỘNG 11](#_Toc198916723)

[6.1 Khái niệm 11](#_Toc198916724)

[6.2 Ví dụ 11](#_Toc198916725)

[7 XỬ LÝ TỆP 12](#_Toc198916726)

[7.1 Khái niệm 12](#_Toc198916727)

[7.2 Ví dụ 12](#_Toc198916728)

[8 KIỂU CẤU TRÚC 15](#_Toc198916729)

[8.1 Khái niệm 15](#_Toc198916730)

[8.2 Ví dụ 15](#_Toc198916731)

[9 DANH SÁCH LIÊN KẾT 17](#_Toc198916732)

[9.1 Khái niệm 17](#_Toc198916733)

[9.2 Ví dụ 17](#_Toc198916734)

[B. ỨNG DỤNG 20](#_Toc198916735)

[KẾT LUẬN 33](#_Toc198916736)

[TÀI LIỆU THAM KHẢO 34](#_Toc198916737)

# LỜI CẢM ƠN

Lời cảm ơn đầu tiên, em xin gửi tới Thầy Trần Phong Nhã môn kỹ thuật lập trình Trường Đại học Giao thông vận tải phân hiệu tại thành phố Hồ Chí Minh cùng với lời chúc sức khỏe và lòng biết ơn sâu sắc.

Em cảm ơn thầy đã tạo điều kiện, giúp đỡ em hoàn thành khoá học. Em cảm ơn thầy vì đã luôn giúp đỡ, hướng dẫn cho em kiến thức, định hướng và kỹ năng để có thể hoàn thành bài báo cáo này.

Em biết bản thân vẫn còn thiếu kinh nghiệm, kiến thức còn hạn chế nên em hy vọng nhận được những lời đánh giá, lời góp ý của thầy để em cải thiện kỹ năng viết báo cáo của mình.

Cuối cùng, em xin gửi lời chúc sức khỏe tới những người xem bài báo cáo này và hơn hết là thầy Trần Phong Nhã, em chúc thầy có thật nhiều niềm vui trong cuộc sống và nhiều thành công trong công việc. Em xin chân thành cảm ơn!

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

**Tp. Hồ Chí Minh, ngày 18 tháng 5 năm 2025  
Giảng viên hướng dẫn**

**Ths. Trần Phong Nhã**

# LÝ THUYẾT

## HÀM:

### 1.1 khái niệm:

- Hàm là một khối mã có thể tái sử dụng, giúp thực hiện một nhiệm vụ cụ thể. Một hàm có thể nhận đầu vào (tham số), thực hiện xử lý, và trả về giá trị đầu ra. Có hai loại chính:

+ Hàm được dựng sẵn: là các hàm có sẵn trong ngôn ngữ lập trình được cung cấp bởi hệ thống hay qua các thư viện chuẩn ví dụ như hàm printf(“…”) hay hàm scanf(…).

+ Hàm do người dùng tự định nghĩa: là các hàm mà người dùng tự sáng chế ra qua các hàm có sẵn trong ngôn ngữ lập trình

* Ngoài ra, giá trị trả về của hàm có thể có hoặc không tuỳ theo kiểu hàm, nếu là void thì hàm sẽ ko trả về giá trị mà sẽ chỉ thực hiện các lệnh trong hàm, còn nếu là các kiểu như int, float, thì hàm vẫn sẽ trả về giá trị theo đúng kiểu đó.
* Lưu ý: nếu hàm theo kiểu int tức kiểu nguyên nhưng giá trị hàm sẽ trả về là kiểu float hay kiểu khác thì hàm khi chạy sẽ xuất hiện lỗi và ko cho ra được kết quả chính xác như yêu cầu

### 1.2 Ví dụ:

* Hàm có sẵn

#include <stdio.h>

int tinhtong(int a,int b){

return a+b;}

int main(){

int s = tinhtong(5,3);

printf("tong a va b la:%d",s);

}

* Hàm tự định nghĩa:

#include <stdio.h>

int laSoChan(int n) {

if (n % 2 == 0) {

return 1;

} else {

return 0;

}

}

int main() {

int num;

printf("Nhập một số: ");

scanf("%d", &num);

if (laSoChan(num)) {

printf("%d là số chẵn\n", num);

} else {

printf("%d là số lẻ\n", num);

}

return 0;

}

* Hàm theo kiểu void:

#include <stdio.h>

void inThongTin(char \*ten, int tuoi) {

printf("Ho ten: %s\n", ten);

printf("Tuoi: %d\n", tuoi);

}

int main() {

char ten[] = "Nguyen Van A";

int tuoi = 20;

inThongTin(ten, tuoi);

return 0;

}

## 2 CON TRỎ:

### 2.1 Khái niệm:

- Con trỏ là một biến lưu trữ địa chỉ của một biến khác trong bộ nhớ, với mục đích truy cập trực tiếp vào bộ nhớ, hỗ trợ cấp phát động, làm việc hiệu quả với mảng hay chuỗi, truyền tham chiếu vào hàm

- Một dạng nâng cao hơn của con trỏ là con trỏ cấp 2, con trỏ trỏ đến địa chỉ con trỏ khác thường được sử dụng để quản lý bộ nhớ động, truyền tham chiếu vào hàm, xử lý danh sách liên kết hay thậm chí là dùng để xử lý các bài toán ma trận

### 2.2 Ví dụ

* Con trỏ cấp 1:

#include <stdio.h>

int main() {

int a = 10;

int \*p;

p = &a;

printf("Gia tri cua a: %d\n", a);

printf("Dia chi cua a: %p\n", &a);

printf("Gia tri cua con tro p: %p\n", p);

printf("Gia tri tai dia chi p tro den: %d\n", \*p);

return 0;

}

* Con trỏ cấp 2:

#include <stdio.h>

int main() {

int a = 10;

int \*p = &a; // Con trỏ p trỏ đến a

int \*\*pp = &p; // Con trỏ pp trỏ đến p

printf("Gia tri cua a: %d\n", a);

printf("Gia tri cua con tro p: %d\n", \*p);

printf("Gia tri cua con tro pp: %d\n", \*\*pp);

return 0;

}

## 3 CON TRỎ MẢNG:

### 3.1 Khái niệm:

Con trỏ mảng là con trỏ trỏ đến phần tử đầu tiên của mảng, cho phép truy cập và thao tác với mảng thông qua địa chỉ bộ nhớ thay vì chỉ số.

Lưu ý: Tên mảng cũng là phần tử đầu tiên của mảng

### 3.2 Ví dụ

#include <stdio.h>

int main() {

int a[] = {0,1, 2};

int \*p = a;

printf("Phan tu dau tien: %d\n", \*p);

printf("Phan tu thu hai: %d\n", \*(p + 1));

printf("Phan tu thu ba: %d\n", \*(p + 2));

}

## 4 MẢNG CON TRỎ:

### 4.1 Khái niệm

Mảng con trỏ là một mảng trong đó mỗi phần tử là một con trỏ. Điều này giúp quản lý động dữ liệu như chuỗi ký tự hoặc mảng động dễ dàng hơn. Thường được dùng để tạo các cấu trúc dữ liệu động phức tạp, quản lý bộ nhớ linh hoạt, xậy dựng hệ thống đa cấp, làm nền tảng cho nhiều thuật toán nâng cao.

### 4.2 Ví dụ

#include <stdio.h>

int main() {

int a = 10, b = 20, c = 30;

int \*a[] = {&a, &b, &c};

printf("a: %d\n", \*a[0]);

printf("b: %d\n", \*a[1]);

printf("c: %d\n", \*a[2]);

}

## 5 CON TRỎ HÀM:

### 5.1 Khái niệm

**Con trỏ hàm** là một con trỏ trỏ đến một hàm. Nó lưu trữ địa chỉ của hàm cho phép ta gọi hàm gián tiếp, truyền hàm như tham số, tạo mảng hàm hoặc cấu trúc hàm.

### 5.2 Ví dụ

#include <stdio.h>

int cong(int a, int b) { return a + b; }

int tru(int a, int b) { return a - b; }

int nhan(int a, int b) { return a \* b; }

void tinhToan(int x, int y, int (\*phepToan)(int, int)) {

printf("Ket qua: %d\n", phepToan(x, y));

}

int main() {

tinhToan(10, 5, cong);

tinhToan(10, 5, tru);

tinhToan(10, 5, nhan);

}

## 6 CẤP PHÁT ĐỘNG

### 6.1 Khái niệm

Là cơ chế quản lý bộ nhớ trong thời gian chương trình vận hành giúp người dùng tự do kiểm soát kích thước bộ nhớ , tăng, giảm, xoá hạy tạo thêm bộ nhớ khi cần kèm với khả năng quản lý bộ nhớ linh hoạt hơn mảng tĩnh.

### 6.2 Ví dụ

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*p = (int \*)malloc(sizeof(int));

if (p == NULL) {

printf("Cap phat that bai!\n");

return 1;

}

\*p = 10;

printf("\*p: %d\n", \*p);

free(p);

}

## 7 XỬ LÝ TỆP

### 7.1 Khái niệm

Là các thao tác làm việc với tệp như:

* + mở têp fopen()
  + Ghi/đọc dữ liệu tệp:fscanf(),fprintf(),fputs(),fgets(),fread(),fwrite()
  + Đóng tệp:fclose()
  + Xoá tệp remove()
  + Đổi tên tệp rename()

….

Những thao tác này giúp người dùng lưu trữ dữ liệu lâu dài, trao đổi dữ liệu giữa các chương trình, hay xử lý dữ liệu lớn không thể lưu trữ hoàn toàn trong ram, chỉnh sửa nội dung trong tệp và hơn thế nữa ….

### 7.2 Ví dụ

#include <stdio.h>

#include <stdlib.h>

void ghiFile(const char \*tenfile) {

FILE \*file = fopen(tenfile, "w");

if (file == NULL) {

printf("Khong the mo tep de ghi!\n");

return;

}

fprintf(file, "Ten: An\nTuoi: 25\n");

fclose(file);

printf("Ghi du lieu thanh cong vao tep %s!\n", tenfile);

}

void docFile(const char \*tenfile) {

FILE \*file = fopen(tenfile, "r");

if (file == NULL) {

printf("Khong the mo tep de doc!\n");

return;

}

char line[100];

printf("Noi dung tep %s:\n", tenfile);

while (fgets(line, sizeof(line), file)) {

printf("%s", line);

}

fclose(file);

}

void ghiTiepFile(const char \*tenfile) {

FILE \*file = fopen(tenfile, "a");

if (file == NULL) {

printf("Khong the mo tep de ghi tiep!\n");

return;

}

fputs("Dia chi: Ha Noi\n", file);

fclose(file);

printf("Ghi tiep du lieu vao tep %s thanh cong!\n", tenfile);

}

void doiTenFile(const char \*ten, const char \*tenmoi) {

if (rename(ten, tenmoi) == 0) {

printf("Doi ten tep tu %s thanh %s thanh cong!\n", ten, tenmoi);

} else {

printf("Doi ten tep that bai!\n");

}

}

void xoaFile(const char \*tenfile) {

if (remove(tenfile) == 0) {

printf("Xoa tep %s thanh cong!\n", tenfile);

} else {

printf("Xoa tep that bai!\n");

}

}

int main() {

char tenfile[] = "vd.txt";

char tenmoi[] = "newvd.txt";

ghiFile(tenfile);

docFile(tenfile);

ghiTiepFile(tenfile);

docFile(tenfile);

doiTenFile(tenfile, tenmoi);

docFile(tenmoi);

xoaFile(tenmoi);

}

## 8 KIỂU CẤU TRÚC

### 8.1 Khái niệm

Là một kiểu dữ liệu do người dùng định nghĩa, giúp nhóm nhiều biến có kiểu dữ liệu khác nhau vào cùng một đối tượng. Với những đặc tính như quản lý tốt dữ liệu có nhiều thuộc tính liên quan, truyền dữ liệu nhóm vào hàm dễ dàng nên một vài ứng dụng của kiểu cấu trúc là dùng để tạo một vài loại danh sách như danh sách sinh viên, bảng lương nhân viên.

### 8.2 Ví dụ

#include <stdio.h>

struct NhanVien {

char ten[50];

int tuoi;

char bophan[40];

};

int main() {

struct NhanVien nv;

printf("Nhap ten: ");

fgets(nv.ten, sizeof(nv.ten), stdin);

printf("Nhap tuoi: ");

scanf("%d", &nv.tuoi);

getchar();

printf("Nhap ten bo phan: ");

fgets(nv.bophan,sizeof(nv.bophan),stdin);

printf("\nThong tin nhan vien:\n");

printf("Ten: %s", nv.ten);

printf("Tuoi: %d\n", nv.tuoi);

printf("Bo phan: %s\n", nv.bophan);

return 0;

}

## 9 DANH SÁCH LIÊN KẾT

### 9.1 Khái niệm

**Danh sách liên kết** là một cấu trúc dữ liệu gồm nhiều **nút liên kết với nhau bằng con trỏ,** trong đó mỗi nút chứa hai phần: dữ liệu và con trỏ trỏ đến nút tiếp theo trong danh sách.

### 9.2 Ví dụ

#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\* next;

};

void themDau(struct Node\*\* head, int data) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

newNode->data = data;

newNode->next = \*head;

\*head = newNode;

}

void themCuoi(struct Node\*\* head, int data) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

struct Node\* t = \*head;

newNode->data = data;

newNode->next = NULL;

if (\*head == NULL) {

\*head = newNode;

return;

}

while (t ->next != NULL) {

t = t ->next;

}

t ->next = newNode;

}

void xoaNode(struct Node\*\* head, int key) {

struct Node\* t = \*head;

struct Node\* prev = NULL;

if (t != NULL && t ->data == key) {

\*head = t ->next;

free(t );

return;

}

while (t != NULL && t ->data != key) {

prev = t ;

t = t ->next;

}

if (t == NULL) return;

prev->next = t ->next;

free(t );

}

void hienThi(struct Node\* head) {

struct Node\* t = head;

while (t != NULL) {

printf("%d -> ", t ->data);

t = t ->next;

}

printf("NULL\n");

}

int main() {

struct Node\* head = NULL;

themDau(&head, 10);

themDau(&head, 20);

themDau(&head, 30);

printf("Danh sach sau khi them vao dau:\n");

hienThi(head);

themCuoi(&head, 40);

themCuoi(&head, 50);

printf("Danh sach sau khi them vao cuoi:\n");

hienThi(head);

xoaNode(&head, 20);

printf("Danh sach sau khi xoa 20:\n");

hienThi(head);

return 0;

}

# B. ỨNG DỤNG

**XÂY DỰNG ỨNG DỤNG CHO VIỆC QUẢN LÝ SINH VIÊN**

**Các tính năng bao gồm:**

* Thêm sinh viên mới
* Xoá sinh viên
* Hiển thị danh sách
* Sắp xếp thứ tự các sinh viên (theo điểm trung bình tăng dần hoặc giảm dần)
* Lưu dữ liệu từ file
* Đọc dữ liệu từ file
* Thống kê số sinh viên có trong các lớp
* Chỉnh sửa thông tin của một sinh viên
* Giao diện menu để người dùng dễ dàng tương tác

**Mục đích:**

* Để giúp nhà trường dễ dàng quản lý các sinh viên với số lượng sinh viên thay đổi theo từng năm.
* Chương trình thống kê theo 4 dữ liệu thông tin là mã số sinh viên, họ tên của sinh viên, điểm trung bình của sinh viên và lơp của sinh viên đó.
* Với giao diện menu người dùng sẽ ko phải khởi động lại chương trình mỗi lần muốn thay đổi thông tin hay dùng lại các chức năng qua đó tăng cảm giác thoải mái khi quản lý các dữ liệu trên máy.

**Phần code:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct SinhVien {

int mssv;

char hoten[50];

float diemtb;

char lop[10];

struct SinhVien\* next;

} SinhVien;

SinhVien\* addSinhVien(int mssv, char\* hoten, float diemtb, char\* lop) {

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

if (sv == NULL) {

printf("Cap phat bo nho that bai!\n");

return NULL;

}

sv->mssv = mssv;

strcpy(sv->hoten, hoten);

sv->diemtb = diemtb;

strcpy(sv->lop, lop);

sv->next = NULL;

return sv;

}

void themSinhVien(SinhVien\*\* head, int mssv, char\* hoten, float diemtb, char\* lop) {

SinhVien\* sv = addSinhVien(mssv, hoten, diemtb, lop);

sv->next = \*head;

\*head = sv;

}

void hienThiDanhSach(SinhVien\* head) {

if (head == NULL) {

printf("Danh sach sinh vien trong!\n");

return;

}

printf("\nDanh sach sinh vien:\n");

printf("%-10s | %-20s | %-10s | %-10s\n", "MSSV", "Ho ten", "Diem TB", "Lop");

while (head != NULL) {

printf("%-10d | %-20s | %-10.2f | %-10s\n", head->mssv, head->hoten, head->diemtb, head->lop);

head = head->next;

}

}

void xoaSinhVien(SinhVien\*\* head, int mssv) {

SinhVien\* temp = \*head, \*prev = NULL;

if (temp != NULL && temp->mssv == mssv) {

\*head = temp->next;

free(temp);

printf("Da xoa sinh vien co MSSV: %d\n", mssv);

return;

}

while (temp != NULL && temp->mssv != mssv) {

prev = temp;

temp = temp->next;

}

if (temp == NULL) {

printf("Khong tim thay sinh vien co MSSV: %d\n", mssv);

return;

}

prev->next = temp->next;

free(temp);

printf("Da xoa sinh vien co MSSV: %d\n", mssv);

}

void sapXepTangDan(SinhVien\*\* head) {

if (\*head == NULL || (\*head)->next == NULL)

return;

SinhVien \*i, \*j;

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

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

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

int tempMSSV = i->mssv;

char tempHoten[50], tempLop[10];

float tempDiem = i->diemtb;

strcpy(tempHoten, i->hoten);

strcpy(tempLop, i->lop);

i->mssv = j->mssv;

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

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

i->diemtb = j->diemtb;

j->mssv = tempMSSV;

strcpy(j->hoten, tempHoten);

strcpy(j->lop, tempLop);

j->diemtb = tempDiem;

}

}

}

printf("Danh sach da duoc sap xep theo diem tang dan!\n");

hienThiDanhSach(\*head);

}

void sapXepGiamDan(SinhVien\*\* head) {

if (\*head == NULL || (\*head)->next == NULL)

return;

SinhVien \*i, \*j;

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

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

if (i->diemtb < j->diemtb) {

int tempMSSV = i->mssv;

char tempHoten[50], tempLop[10];

float tempDiem = i->diemtb;

strcpy(tempHoten, i->hoten);

strcpy(tempLop, i->lop);

i->mssv = j->mssv;

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

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

i->diemtb = j->diemtb;

j->mssv = tempMSSV;

strcpy(j->hoten, tempHoten);

strcpy(j->lop, tempLop);

j->diemtb = tempDiem;

}

}

}

printf("Danh sach da duoc sap xep theo diem giam dan!\n");

hienThiDanhSach(\*head);

}

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

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

if (fp == NULL) {

printf("Mo file de ghi that bai!\n");

return;

}

while (head != NULL) {

fprintf(fp, "%d,%s,%.2f,%s\n", head->mssv, head->hoten, head->diemtb, head->lop);

head = head->next;

}

fclose(fp);

printf("Da ghi danh sach vao file: %s\n", filename);

}

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

FILE\* fp = fopen(filename, "r");

if (fp == NULL) {

printf("Mo file de doc that bai!\n");

return;

}

int mssv;

char hoten[50], lop[10];

float diemtb;

while (fscanf(fp, "%d,%49[^,],%f,%9s", &mssv, hoten, &diemtb, lop) != EOF) {

themSinhVien(head, mssv, hoten, diemtb, lop);

}

fclose(fp);

printf("Da doc danh sach tu file: %s\n", filename);

}

void thongKeTheoLop(SinhVien\* head) {

if (head == NULL) {

printf("Danh sach sinh vien trong!\n");

return;

}

char cacLop[100][10];

int soLuong[100] = {0};

int lopCount = 0;

SinhVien\* temp = head;

while (temp != NULL) {

int daTonTai = 0;

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

if (strcmp(cacLop[i], temp->lop) == 0) {

soLuong[i]++;

daTonTai = 1;

break;

}

}

if (!daTonTai) {

strcpy(cacLop[lopCount], temp->lop);

soLuong[lopCount] = 1;

lopCount++;

}

temp = temp->next;

}

printf("\nThong ke sinh vien theo lop:\n");

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

printf("Lop: %s - So sinh vien: %d\n", cacLop[i], soLuong[i]);

}

}

SinhVien\* timSinhVien(SinhVien\* head, int mssv) {

while (head != NULL) {

if (head->mssv == mssv)

return head;

head = head->next;

}

return NULL;

}

void chinhSuaSinhVien(SinhVien\* head, int mssv) {

SinhVien\* sv = timSinhVien(head, mssv);

if (sv == NULL) {

printf("Khong tim thay sinh vien co MSSV: %d\n", mssv);

return;

}

printf("Nhap ho ten moi: ");

fgets(sv->hoten, sizeof(sv->hoten), stdin);

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

printf("Nhap diem trung binh moi: ");

scanf("%f", &sv->diemtb);

getchar();

printf("Nhap lop moi: ");

fgets(sv->lop, sizeof(sv->lop), stdin);

sv->lop[strcspn(sv->lop, "\n")] = '\0';

printf("Da cap nhat thong tin sinh vien!\n");

}

int main() {

SinhVien\* danhSach = NULL;

int choice, mssv;

char hoten[50], lop[10];

float diemtb;

char filename[50] = "sinhvien.txt";

do {

printf("\n===== QUAN LY SINH VIEN =====\n");

printf("1: Them sinh vien\n");

printf("2: Xoa sinh vien\n");

printf("3: Hien thi danh sach\n");

printf("4: Sap xep theo diem tang dan\n");

printf("5: Sap xep theo diem giam dan\n");

printf("6: Ghi danh sach vao file\n");

printf("7: Doc danh sach tu file\n");

printf("8: Thong ke sinh vien theo lop\n");

printf("9: Tim sinh vien theo MSSV\n");

printf("10: Chinh sua thong tin sinh vien\n");

printf("11: Thoat\n");

printf("Chon: ");

scanf("%d", &choice);

getchar();

switch (choice) {

case 1:

printf("Nhap MSSV: ");

scanf("%d", &mssv);

getchar();

printf("Nhap ho ten: ");

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

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

printf("Nhap diem trung binh: ");

scanf("%f", &diemtb);

getchar();

printf("Nhap lop: ");

fgets(lop, sizeof(lop), stdin);

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

themSinhVien(&danhSach, mssv, hoten, diemtb, lop);

break;

case 2:

printf("Nhap MSSV de xoa: ");

scanf("%d", &mssv);

getchar();

xoaSinhVien(&danhSach, mssv);

break;

case 3:

hienThiDanhSach(danhSach);

break;

case 4:

sapXepTangDan(&danhSach);

break;

case 5:

sapXepGiamDan(&danhSach);

break;

case 6:

ghiFile(danhSach, filename);

printf("Ghi file thanh cong");

break;

case 7:

docFile(&danhSach, filename);

break;

case 8:

thongKeTheoLop(danhSach);

break;

case 9:

{

printf("Nhap MSSV can tim: ");

scanf("%d", &mssv);

getchar();

SinhVien\* sv = timSinhVien(danhSach, mssv);

if (sv != NULL) {

printf("\nThong tin sinh vien:\n");

printf("MSSV: %d\n", sv->mssv);

printf("Ho ten: %s\n", sv->hoten);

printf("Diem TB: %.2f\n", sv->diemtb);

printf("Lop: %s\n", sv->lop);

} else {

printf("Khong tim thay sinh vien co MSSV: %d\n", mssv);

}

}

break;

case 10:

printf("Nhap MSSV cua sinh vien can chinh sua: ");

scanf("%d", &mssv);

getchar();

chinhSuaSinhVien(danhSach, mssv);

break;

case 11:

printf("Thoat chuong trinh thanh cong!\n");

break;

default:

printf("Lua chon khong hop le!Vui long chi chon tu 1 den 11\n");

}

} while (choice != 11);

return 0;

}

# KẾT LUẬN

Qua quá trình thực hiện bài tập lớn môn kỹ thuật lập trình, em đã xây dựng thành công chương trình quản lý sinh viên sử dụng ngôn ngữ C với các chức năng cơ bản như thêm, xóa, hiển thị, sắp xếp sinh viên và lưu trữ dữ liệu vào file cũng như rèn luyện lại các thao tác lập trình trong ngôn ngữ C. Ngoài ra, em còn được rèn luyện kỹ năng phân tích, thiết kế và triển khai phần mềm.

Em mong nhận được những lời đánh giá chân thành của thầy để sửa chữa và cải thiện khả năng tư duy, lập trình của bản thân

# TÀI LIỆU THAM KHẢO

[1]” Tài liệu lập trình C và cách học lập trình C hiệu quả” <https://blog.28tech.com.vn/c-tai-lieu-lap-trinh-c-va-cach-hoc-lap-trinh-c-hieu-qua>. [Truy cập vào14/5/2025]

[2] “Lập trình C cơ bản - Ứng dụng của ngôn ngữ C”. [truy cập vào 14/5/2025]

[3] “Stack Overflow” <https://stackoverflow.com/questions> [Truy cập vào 15/5/2025]