**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**HỌC PHẦN: KĨ THUẬT LẬP TRÌNH**

**Giảng viên hướng dẫn: TRẦN PHONG NHÃ**

**Sinh viên thực hiện:  ĐOÀN NGUYỄN ANH KHOA**

**Lớp:    CQ.65. CNTT**

**Khoá:   K65**

**Tp. Hồ Chí Minh, năm 2025**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**HỌC PHẦN: KĨ THUẬT LẬP TRÌNH**

**Giảng viên hướng dẫn: TRẦN PHONG NHÃ**

**Sinh viên thực hiện:  ĐOÀN NGUYỄN ANH KHOA**

**Lớp:    CQ.65. CNTT**

**Khoá:   K65**

**Tp. Hồ Chí Minh, năm 2025**

# MỤC LỤC

**[MỤC LỤC i](#_Toc164725518)**

**[LỜI CẢM ƠN ii](#_Toc164725519)**

**[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN iii](#_Toc164725520)**

**[A. LÝ THUYẾT 5](#_Toc164725521)**

**[1.HÀM: 5](#_Toc164725522)**

**[2.CON TRỎ:](#_Toc164725523) 6**

**[3.CON TRỎ MẢNG:](#_Toc164725524) 7**

**4.[MẢNG CON TRỎ:](#_Toc164725525) 8**

**[5.CON TRỎ HÀM:](#_Toc164725526) 9**

**6.[CẤP PHÁT ĐỘNG:](#_Toc164725527) 10**

**[7.XỬ LÍ TỆP:](#_Toc164725528) 11**

**8.KIỂU CẤU TRÚC......................................................................................12**

**[9.DANH SÁCH LIÊN KẾT:](#_Toc164725529) 13**

**[B. ỨNG DỤNG](#_Toc164725530) 16**

**[KẾT LUẬN](#_Toc164725531) 22**

# **LỜI CẢM ƠN**

Chào thầy và các bạn sinh viên thân mến!

Em xin gửi lời cảm ơn chân thành nhất đến tất cả những người đã đóng góp và hỗ trợ cho bài báo cáo tiểu luận này. Sự giúp đỡ và động viên của thầy và các bạn đã đóng vai trò vô cùng quan trọng trong quá trình chuẩn bị và hoàn thiện bài báo cáo này.

Em muốn bày tỏ lòng biết ơn sâu sắc đến các giảng viên và người hướng dẫn đã cung cấp cho em những kiến thức và hướng dẫn quý báu. Nhờ sự chỉ dạy tận tâm của thầy, em đã có cơ hội nắm vững kiến thức và kỹ năng cần thiết để thực hiện bài báo cáo này.

Em xin gửi lời cảm ơn đến tất cả những người đã đọc báo cáo này và quan tâm đến công trình của em. Sự quan tâm và phản hồi của thầy cô và các bạn là nguồn động lực lớn để em tiếp tục phát triển và nỗ lực hơn nữa trong tương lai.

Một lần nữa, em xin chân thành cảm ơn tất cả mọi người vì sự đóng góp và hỗ trợ của thầy cô và các bạn. Sự giúp đỡ của mọi người đã góp phần quan trọng vào thành công của bài báo cáo này.

Trân trọng,

*Đoàn Nguyễn Anh Khoa*

# 

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
..........................................................................................................................................  
.........................................................................................................................................  
..........................................................................................................................................  
....................................................................................................................................................................................................................................................................................  
..........................................................................................................................................

|  |
| --- |
| **Tp. Hồ Chí Minh, ngày tháng năm 2025 Giảng viên hướng dẫn**  **Trần Phong Nhã** |

# 

# LÝ THUYẾT

**1)Hàm**

**1.1)Khái niệm:**

Là khối lệnh thực hiện một công việc cụ thể. Có thể truyền tham số và trả về giá trị. Giúp tái sử dụng mã và tổ chức chương trình rõ ràng.

Các thành phần của hàm:

* data\_type : Kiểu trả về của hàm, có thể là các kiểu dữ liệu như int, long long, float, char, double, hoặc void (tương ứng với kiểu trả về là rỗng)
* function\_name : Tên của hàm, cần tuân theo quy tắc như đặt tên biến
* parameter : Tham số của hàm, đây được coi như đầu vào của hàm. Bạn có thể xây dựng bao nhiêu tham số tùy ý và lựa chọn kiểu dữ liệu cho từng tham số.
* code : Các câu lệnh bên trong của hàm

**1.2)Ví dụ:**

**#include <stdio.h>**

**// Khai báo hàm**

**int tinhTong(int a, int b);**

**int main() {**

**int x = 5, y = 7;**

**int tong = tinhTong(x, y);**

**printf("Tong cua %d va %d la: %d\n", x, y, tong);**

**return 0;**

**}**

**// Định nghĩa hàm**

**int tinhTong(int a, int b) {**

**return a + b;**

**}**

# 

# Con trỏ

**2.1)Khái niệm:**

Con trỏ hay biến con trỏ cũng là một biến thông thường nhưng giá trị mà nó lưu lại là địa chỉ của 1 biến khác.

Ví dụ biến kiểu int N trong chương trình sẽ có địa chỉ nhất định trong bộ nhớ, để lưu trữ giá trị địa chỉ này ta cần biến con trỏ kiểu int

**2.2)Ví dụ:**

**#include <stdio.h>**

**int main() {**

**int a = 10;**

**int \*p; // khai báo con trỏ p trỏ tới kiểu int**

**p = &a; // p trỏ đến địa chỉ của biến a**

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

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

**printf("Gia tri cua p (dia chi): %p\n", p);**

**printf("Gia tri ma p tro toi (\*p): %d\n", \*p); // truy cập giá trị qua con trỏ**

**return 0;**

**}**

# Con trỏ mảng

# Con trỏ mảng

# Con trỏ mảng

**3.1)Khái niệm:**

Là con trỏ trỏ tới phần tử đầu tiên của mảng. Có thể duyệt mảng thông qua con trỏ bằng cách tăng địa chỉ.

**3.2)Ví dụ:**

**#include <stdio.h>**

**int main() {**

**int arr[5] = {10, 20, 30, 40, 50};**

**int (\*ptr)[5]; // khai báo con trỏ trỏ tới mảng gồm 5 phần tử kiểu int**

**ptr = &arr; // gán địa chỉ của mảng arr cho con trỏ**

**printf("Duyet mang bang con tro tro toi mang:\n");**

**for (int i = 0; i < 5; i++) {**

**printf("%d ", (\*ptr)[i]); // truy cập phần tử mảng thông qua con trỏ**

**}**

**return 0;**

**}**

# 4)Mảng con trỏ

**4.1)Khái niệm**

Là mảng mà mỗi phần tử là một con trỏ. Dùng phổ biến để lưu danh sách chuỗi (mảng ký tự).

**4.2)Ví dụ**

**#include <stdio.h>**

**int main() {**

**const char \*dsTen[] = {**

**"Nam",**

**"Lan",**

**"Binh",**

**"Hoa"**

**};**

**int soLuong = sizeof(dsTen) / sizeof(dsTen[0]);**

**printf("Danh sach ten:\n");**

**for (int i = 0; i < soLuong; i++) {**

**printf("%s\n", dsTen[i]);**

**}**

**return 0;**

**}**

**5)Con trỏ hàm**

**5.1)Khái niệm:**

Là con trỏ trỏ tới một hàm. Dùng để gọi hàm thông qua con trỏ, thích hợp cho các hàm callback hoặc bảng hàm

**5.2)Ví dụ:**

**#include <stdio.h>**

**// Hàm cộng 2 số nguyên**

**int cong(int a, int b) {**

**return a + b;**

**}**

**int main() {**

**// Khai báo con trỏ hàm: trỏ đến hàm nhận 2 int và trả về int**

**int (\*ptrHam)(int, int);**

**// Gán địa chỉ hàm cong cho con trỏ**

**ptrHam = &cong;**

**// Gọi hàm thông qua con trỏ**

**int ketQua = ptrHam(5, 7);**

**printf("Ket qua: %d\n", ketQua); // In ra 12**

**return 0;**

**}**

1. **Cấp phát động**

**6.1)Khái niệm**

-Là kỹ thuật cấp phát bộ nhớ khi chương trình đang chạy (dùng malloc, calloc, new...). Phải giải phóng bộ nhớ khi không dùng nữa để tránh rò rỉ.

**6.2)Ví dụ**

**#include <stdio.h>**

**#include <stdlib.h> // cần để dùng malloc và free**

**int main() {**

**int \*arr;**

**int n;**

**printf("Nhap so phan tu: ");**

**scanf("%d", &n);**

**// Cấp phát động vùng nhớ cho mảng n phần tử**

**arr = (int \*)malloc(n \* sizeof(int));**

**// Kiểm tra nếu cấp phát thất bại**

**if (arr == NULL) {**

**printf("Khong du bo nho!\n");**

**return 1;**

**}**

**// Nhập giá trị cho mảng**

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

**printf("arr[%d] = ", i);**

**scanf("%d", &arr[i]);**

**}**

**// In ra mảng**

**printf("Gia tri trong mang:\n");**

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

**printf("%d ", arr[i]);**

**}**

**// Giải phóng bộ nhớ**

**free(arr);**

**return 0;**

**}**

1. **Xử lí tệp**

**7.1)Khái niệm**

Là thao tác đọc/ghi dữ liệu vào tệp. Sử dụng fopen, fread, fwrite, fprintf, fscanf, fclose…

**7.2)Ví dụ:**

**#include <stdio.h>**

**int main() {**

**FILE \*f;**

**char tenFile[] = "dulieu.txt";**

**// GHI FILE**

**f = fopen(tenFile, "w"); // Mở file ở chế độ ghi ("w" - write)**

**if (f == NULL) {**

**printf("Khong mo duoc file!\n");**

**return 1;**

**}**

**fprintf(f, "Xin chao!\n");**

**fprintf(f, "Day la file van ban.\n");**

**fclose(f); // Đóng file sau khi ghi**

**// ĐỌC FILE**

**f = fopen(tenFile, "r"); // Mở file ở chế độ đọc ("r" - read)**

**if (f == NULL) {**

**printf("Khong mo duoc file de doc!\n");**

**return 1;**

**}**

**char line[100];**

**printf("Noi dung file:\n");**

**while (fgets(line, sizeof(line), f)) {**

**printf("%s", line);**

**}**

**fclose(f); // Đóng file sau khi đọc**

**return 0;**

**}**

1. **Kiểu cấu trúc**

**8.1)Khái niệm:**

Là kiểu dữ liệu do người dùng định nghĩa (dùng struct) để nhóm nhiều biến liên quan lại với nhau

**8.2)Ví dụ:**

**#include <stdio.h>**

**#include <string.h>**

**// Định nghĩa kiểu cấu trúc sinh viên**

**struct SinhVien {**

**int maSV;**

**char ten[50];**

**float diemTB;**

**};**

**int main() {**

**// Khai báo biến sv1 thuộc kiểu struct SinhVien**

**struct SinhVien sv1;**

**// Gán giá trị cho các thành phần**

**sv1.maSV = 1001;**

**strcpy(sv1.ten, "Nguyen Van A");**

**sv1.diemTB = 8.5;**

**// In thông tin sinh viên**

**printf("Ma SV: %d\n", sv1.maSV);**

**printf("Ten : %s\n", sv1.ten);**

**printf("Diem : %.2f\n", sv1.diemTB);**

**return 0;**

**}**

1. **Danh sách liên kết**

**9.1)Khái niệm:**

Là cấu trúc dữ liệu gồm các nút, mỗi nút chứa dữ liệu và con trỏ trỏ đến nút kế tiếp. Dễ chèn/xóa phần tử hơn mảng.

**9.2)Ví dụ:**

**#include <stdio.h>**

**#include <stdlib.h>**

**// Định nghĩa một node trong danh sách liên kết**

**struct Node {**

**int data;**

**struct Node\* next;**

**};**

**// Hàm tạo node mới**

**struct Node\* taoNode(int giaTri) {**

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

**nodeMoi->data = giaTri;**

**nodeMoi->next = NULL;**

**return nodeMoi;**

**}**

**// Hàm thêm node vào cuối danh sách**

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

**struct Node\* nodeMoi = taoNode(giaTri);**

**if (\*head == NULL) {**

**\*head = nodeMoi;**

**} else {**

**struct Node\* temp = \*head;**

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

**temp = temp->next;**

**temp->next = nodeMoi;**

**}**

**}**

**// Hàm in danh sách**

**void inDanhSach(struct Node\* head) {**

**printf("Danh sach lien ket: ");**

**while (head != NULL) {**

**printf("%d -> ", head->data);**

**head = head->next;**

**}**

**printf("NULL\n");**

**}**

**// Hàm chính**

**int main() {**

**struct Node\* danhSach = NULL;**

**themCuoi(&danhSach, 10);**

**themCuoi(&danhSach, 20);**

**themCuoi(&danhSach, 30);**

**inDanhSach(danhSach);**

**// Giải phóng bộ nhớ**

**struct Node\* temp;**

**while (danhSach != NULL) {**

**temp = danhSach;**

**danhSach = danhSach->next;**

**free(temp);**

**}**

**return 0;**

**}**

**B. Ứng dụng:**

**Xây dựng ứng dụng cho bài toán cụ thể với đầy đủ tính năng**

**cần thiết (VD: bài quản lý SV, Đoàn viên, Số Phức…)**

**1. Ứng dụng con trỏ, cấp phát động để lấy ví dụ minh họa về cấu trúc mảng**

**2. Ứng dụng danh sách liên kết lấy tối thiểu 1 ví dụ**

**3. Giao diện/menu chương trình: sẽ được cộng điểm**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**// Cấu trúc sinh viên**

**typedef struct SinhVien {**

**int maSV;**

**char ten[50];**

**float diemTB;**

**struct SinhVien \*next;**

**} SinhVien;**

**// Tạo node mới**

**SinhVien\* taoSinhVien(int ma, const char \*ten, float diem) {**

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

**sv->maSV = ma;**

**strcpy(sv->ten, ten);**

**sv->diemTB = diem;**

**sv->next = NULL;**

**return sv;**

**}**

**// Thêm sinh viên vào cuối danh sách**

**void themSinhVien(SinhVien \*\*head) {**

**int ma;**

**char ten[50];**

**float diem;**

**printf("Nhap ma sinh vien: ");**

**scanf("%d", &ma);**

**printf("Nhap ten sinh vien: ");**

**fflush(stdin); // Xóa bộ đệm**

**fgets(ten, sizeof(ten), stdin);**

**ten[strcspn(ten, "\n")] = '\0'; // Xóa ký tự newline**

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

**scanf("%f", &diem);**

**SinhVien \*svMoi = taoSinhVien(ma, ten, diem);**

**if (\*head == NULL) {**

**\*head = svMoi;**

**} else {**

**SinhVien \*temp = \*head;**

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

**temp = temp->next;**

**temp->next = svMoi;**

**}**

**printf(">> Da them sinh vien!\n");**

**}**

**// Hiển thị danh sách sinh viên**

**void hienThiDanhSach(SinhVien \*head) {**

**if (head == NULL) {**

**printf(">> Danh sach rong!\n");**

**return;**

**}**

**printf("\n%-10s %-30s %-10s\n", "MaSV", "Ten", "DiemTB");**

**printf("-----------------------------------------------\n");**

**SinhVien \*temp = head;**

**while (temp != NULL) {**

**printf("%-10d %-30s %-10.2f\n", temp->maSV, temp->ten, temp->diemTB);**

**temp = temp->next;**

**}**

**}**

**// Xóa sinh viên theo mã**

**void xoaSinhVien(SinhVien \*\*head) {**

**int ma;**

**printf("Nhap ma sinh vien muon xoa: ");**

**scanf("%d", &ma);**

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

**while (temp != NULL && temp->maSV != ma) {**

**prev = temp;**

**temp = temp->next;**

**}**

**if (temp == NULL) {**

**printf(">> Khong tim thay sinh vien co ma %d.\n", ma);**

**return;**

**}**

**if (prev == NULL) {**

**\*head = temp->next; // xóa đầu**

**} else {**

**prev->next = temp->next;**

**}**

**free(temp);**

**printf(">> Da xoa sinh vien co ma %d.\n", ma);**

**}**

**// Giải phóng toàn bộ danh sách**

**void giaiPhong(SinhVien \*head) {**

**SinhVien \*temp;**

**while (head != NULL) {**

**temp = head;**

**head = head->next;**

**free(temp);**

**}**

**}**

**// Menu chính**

**void menu() {**

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

**printf("1. Them sinh vien\n");**

**printf("2. Hien thi danh sach\n");**

**printf("3. Xoa sinh vien theo ma\n");**

**printf("4. Thoat\n");**

**printf("Chon chuc nang (1-4): ");**

**}**

**int main() {**

**SinhVien \*danhSach = NULL;**

**int luaChon;**

**do {**

**menu();**

**scanf("%d", &luaChon);**

**switch (luaChon) {**

**case 1:**

**themSinhVien(&danhSach);**

**break;**

**case 2:**

**hienThiDanhSach(danhSach);**

**break;**

**case 3:**

**xoaSinhVien(&danhSach);**

**break;**

**case 4:**

**giaiPhong(danhSach);**

**printf(">> Da thoat chuong trinh.\n");**

**break;**

**default:**

**printf(">> Lua chon khong hop le. Vui long nhap tu 1-4.\n");**

**}**

**} while (luaChon != 4);**

**return 0;**

**}**

# KẾT LUẬN

Thông qua cách xây dựng bài toán trên, cùng với việc kết hợp giữa con trỏ, cấp phát động, danh sách liên kết , cơ bản đã đáp ứng được các điều cần có của việc quản lí sinh viên. Tuy nhiên, đây chỉ mới là ý tưởng và đang trong quá trình cải thiện, nâng cấp. Không tránh khỏi một số sai sót trong khâu thiết kế chương trình.

Bên cạnh đó, em mong được nhận những nhận xét từ thầy để phần nào cải thiện khả năng lập trình của mình. Cũng như có thể nâng cấp được bài toán đã trình bày trên từ đó mà tối ưu nhất cách thực thi của bài toán trong việc quản lí sinh viên