Đại học Đà Nẵng

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT



BÁO CÁO CÁ NHÂN

**THỰC HÀNH CTDL&GT**

**Đề số 5**

Giảng viên: Nguyễn Thị Thùy Trang

Họ tên sinh viên: Nguyễn Văn Linh

Mã sinh viên: 22115053122225

Lớp HP:123TC>03 (Sáng thứ 3, Tiết 1-2)

**Đà nẵng, ngày 23 tháng 11 năm 2023**

**Đề số 5:**

[**Phần I** **: Code** 3](#_Toc154257059)

[Bài 1: Khai báo cấu trúc dữ liệu để tổ chức danh sách liên kết đơn quản lý điểm của sinh viên gồm các thông tin của mỗi sinh viên bao gồm: Mã sinh viên, tên tên sinh viên, điểm chuyên cần, điểm giữa kỳ, điểm cuối kỳ.Yêu cầu: 3](#_Toc154257060)

[Bài 2: Viết chương trình thực hiện sắp xếp một dãy gồm N (N ≤ 100) số nguyên yêu cầu cụ thể như sau: 11](#_Toc154257061)

[**Phần II** **: Kết quả chạy chương trình của từng bài** 16](#_Toc154257062)

[Bài 1: Hỉnh ảnh chạy kết quả 16](#_Toc154257063)

[a) Thêm ,sửa, xóa theo mã sinh viên ở vị trí bất kỳ 16](#_Toc154257064)

[b) Tính điểm trung bình của mỗi sinh viện như sau điểm chuyên cần hệ số 2, điểm giữa kỳ hệ số 2, điểm cuỗi kỳ hệ số 6 . 21](#_Toc154257065)

[c) Tìm vị trí của sinh viên có điểm trung bình lớn nhất. 21](#_Toc154257066)

[d) Sắp xếp tăng dần danh sách sinh viên theo điểm trung bình. 22](#_Toc154257067)

[Bài 2: Hình ảnh chạy kết quả 22](#_Toc154257068)

[a) **Sắp** xếp Quick Sort tăng dần theo trị tuyệt đối 22](#_Toc154257069)

[b) Shell Sort: Sắp tăng dần với bước chạy tùy chọn nhập vào từ bàn phím 23](#_Toc154257070)

[c) HeapSort Sắp xếp giảm dần . 24](#_Toc154257071)

1. **: Code**

## Bài 1: Khai báo cấu trúc dữ liệu để tổ chức danh sách liên kết đơn quản lý điểm của sinh viên gồm các thông tin của mỗi sinh viên bao gồm: Mã sinh viên, tên tên sinh viên, điểm chuyên cần, điểm giữa kỳ, điểm cuối kỳ.Yêu cầu:

1. Thêm ,sửa, xóa theo mã sinh viên ở vị trí bất kỳ;

2. Tính điểm trung bình của mỗi sinh viện như sau điểm chuyên cần hệ số 2, điểm giữa kỳ hệ số 2, điểm cuỗi kỳ hệ số 6 .

3. Tìm vị trí của sinh viên có điểm trung bình lớn nhất.

4. Sắp xếp tang dần danh sách sinh viên theo điểm trung bình.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct SinhVien {

int maSV;

char tenSV[50];

float diemCC;

float diemGK;

float diemCK;

float diemTB;

struct SinhVien\* next;

};

typedef struct SinhVien SinhVien;

SinhVien\* makeNode(int maSV, const char\* tenSV, float diemCC, float diemGK, float diemCK) {

SinhVien\* p = new SinhVien;

p->maSV = maSV;

strcpy(p->tenSV, tenSV);

p->diemCC = diemCC;

p->diemGK = diemGK;

p->diemCK = diemCK;

p->diemTB = 0;

p->next = NULL;

return p;

}

// ham them sinh vien

SinhVien\* addAft(SinhVien\* H, int maSV, const char\* tenSV, float diemCC, float diemGK, float diemCK) {

SinhVien\* p = makeNode(maSV, tenSV, diemCC, diemGK, diemCK);

if (H == NULL)

H = p;

else {

SinhVien\* tmp = H;

while (tmp->next != NULL)

tmp = tmp->next;

tmp->next = p;

}

return H;

}

// ham in danh sach sinh vien

void duyet(SinhVien\* H) {

SinhVien\* tmp = H;

printf("%8s \t %25s \t %10s \t %10s \t %10s \t %10s\n", "MaSV", "Ten SV", "CC", "GK", "CK", "DTB");

while (tmp != NULL) {

printf("%8d \t %25s \t %10.2f \t %10.2f \t %10.2f \t %10.2f\n",

tmp->maSV, tmp->tenSV, tmp->diemCC, tmp->diemGK, tmp->diemCK, tmp->diemTB);

tmp = tmp->next;

}

}

// tinh diem trung binh

void tinhDiemTB(SinhVien\* sv) {

sv->diemTB = (sv->diemCC \* 2 + sv->diemGK \* 2 + sv->diemCK \* 6) / 10;

}

void tinhDiemTrungBinh(SinhVien\* danhSachSV) {

SinhVien\* tmp = danhSachSV;

while (tmp != NULL) {

tinhDiemTB(tmp);

tmp = tmp->next;

}

printf("Da tinh diem trung binh cho tat ca sinh vien.\n");

}

// them sinh vien moi theo ma sinh vien

SinhVien\* themSinhVienTheoMaSV(SinhVien\* danhSachSV, int maSV) {

SinhVien\* svMoi = new SinhVien;

svMoi->maSV = maSV;

strcpy(svMoi->tenSV, " ");

// Cap nhat cac cot diem bang 0

svMoi->diemCC = 0.0;

svMoi->diemGK = 0.0;

svMoi->diemCK = 0.0;

svMoi->diemTB = 0.0;

svMoi->next = danhSachSV;

danhSachSV = svMoi;

printf("Da them sinh vien voi ma %d vao danh sach.\n", maSV);

return danhSachSV;

}

// hàm tìm kiem sinh vien theo ma sinh vien de cho ham xoa su dung

SinhVien\* timSinhVienTheoMaSV(SinhVien\* danhSachSV, int maSV) {

SinhVien\* sv = danhSachSV;

while (sv != NULL) {

if (sv->maSV == maSV) {

return sv; // Tra ve con tro den sinh vien neu tim thay

}

sv = sv->next;

}

// Tra ve NULL neu không tìm thay sinh viên có mã tuong ung

return NULL;

}

// ham xoa sinh vien theo ma sinh vien.

SinhVien \*xoaMaSV(SinhVien \*H, int maSV){

SinhVien \*newSv, \*temp;

while(timSinhVienTheoMaSV(H, maSV)!=NULL){

newSv= timSinhVienTheoMaSV(H, maSV);

if(H==newSv){

H=H->next;

}else{

temp =H;

while(temp->next != newSv)

temp = temp->next;

temp->next = newSv->next;

}

newSv->next = NULL;

}

return H;

}

// sua sinh vien theo ma sinh vien

void suaSinhVien(SinhVien\* danhSachSV, int maSV, const char\* tenSV, float diemCC, float diemGK, float diemCK) {

SinhVien\* tmp = danhSachSV;

while (tmp != NULL && tmp->maSV != maSV) {

tmp = tmp->next;

}

if (tmp == NULL) {

printf("Khong tim thay sinh vien co maSV = %d\n", maSV);

return;

}

// Cap nhat thong tin

strcpy(tmp->tenSV, tenSV);

tmp->diemCC = diemCC;

tmp->diemGK = diemGK;

tmp->diemCK = diemCK;

//tinhDiemTB(tmp);// ban co the them dong nay neu ban muon cap nhat luon DTB

printf("Da sua thong tin cho sinh vien co maSV = %d\n", maSV);

}

// ham tim vi tri max cua sinh vien có diem TB Cao nhat

int timViTriMax(SinhVien\* danhSachSV) {

SinhVien\* tmp = danhSachSV;

int viTri = 1;

float maxDiemTB = tmp->diemTB;

for (int i = 1; tmp != NULL; ++i) {

if (tmp->diemTB > maxDiemTB) {

maxDiemTB = tmp->diemTB;

viTri = i;

}

tmp = tmp->next;

}

return viTri;

}

// ham sap xep sinh vien

SinhVien\* sapXepDanhSach(SinhVien\* danhSachSV) {

SinhVien\* tmp = danhSachSV;

SinhVien\* sortedList = NULL;// danh sach sinh vien sau khi sap xep

while (tmp != NULL) {

SinhVien\* current = tmp;

tmp = tmp->next;

// Chèn nút hien tai vào danh sách dã sap xep

if (sortedList == NULL || current->diemTB <= sortedList->diemTB) {

current->next = sortedList;

sortedList = current;

} else {

SinhVien\* prev = NULL;

SinhVien\* nextNode = sortedList;

while (nextNode != NULL && current->diemTB > nextNode->diemTB){

prev = nextNode;

nextNode = nextNode->next;

}

current->next = nextNode;

// Cap nhat các con tro truoc dó

if (prev != NULL) {

prev->next = current;

} else {

sortedList = current;

}

}

}

return sortedList;

}

int main() {

SinhVien\* danhSachSV = NULL;

int choice;

do {

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

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

printf("2. In danh sach sinh vien\n");

printf("3. Them sinh vien voi ma sinh vien\n");

printf("4. Tinh diem trung binh cho sinh vien\n");

printf("5. Sua sinh vien theo maSV\n");

printf("6. Xoa sinh vien theo maSV\n");

printf("7. Tim vi tri sinh vien co diem trung binh cao nhat\n");

printf("8. Sap xep danh sach sinh vien tang theo diem trung binh\n");

printf("0. Thoat\n");

printf("Chon: ");

scanf("%d", &choice);

switch (choice) {

// Them sinh vien voi tat ca thong tin

case 1: {

int maSV;

char tenSV[50];

float diemCC, diemGK, diemCK;

printf("Nhap ma sinh vien: ");

scanf("%d", &maSV);

printf("Nhap ten sinh vien: ");

scanf(" %[^\n]s", tenSV);

printf("Nhap diem chuyen can: ");

scanf("%f", &diemCC);

printf("Nhap diem giua ky: ");

scanf("%f", &diemGK);

printf("Nhap diem cuoi ky: ");

scanf("%f", &diemCK);

// Kiem tra diem nhap vào co >=0 hay khong

if (diemCC < 0 || diemGK < 0 || diemCK < 0) {

printf("Diem khong hop le. Vui long nhap lai.\n");

break;

}

danhSachSV=addAft(danhSachSV,maSV, tenSV, diemCC, diemGK, diemCK);

printf("Da them sinh vien moi.\n");

break;

}

// in Danh sach sinh vien có trong danh sách

case 2:

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

duyet(danhSachSV);

break;

case 3:

// Them sinh vien chi voi ma sinh vien.

int maSV ;

printf("Nhap maSV sinh vien Can them: ");

scanf("%d",&maSV);

danhSachSV=themSinhVienTheoMaSV(danhSachSV,maSV);

printf("Danh sach sinh vien sau khi them:\n");

break;

case 4: {

// tinh diem trung binh cho cac sinh vien

tinhDiemTrungBinh(danhSachSV);

duyet(danhSachSV);

break;

}

case 5:{

// Sua thong tin sinh vien theo ma Sinh vien

int maSV;

char tenSV[50];

float diemCC, diemGK, diemCK;

printf("Nhap ma sinh vien muon sua: ");

scanf("%d", &maSV);

printf("Nhap ten sinh vien: ");

scanf(" %[^\n]s", tenSV);

printf("Nhap diem chuyen can: ");

scanf("%f", &diemCC);

printf("Nhap diem giua ky: ");

scanf("%f", &diemGK);

printf("Nhap diem cuoi ky: ");

scanf("%f", &diemCK);

// Kiem tra diem nhap vao có hop le >=0 ?

if (diemCC < 0 || diemGK < 0 || diemCK < 0) {

printf("Diem khong hop le. Vui long nhap lai.\n");

break;

}

suaSinhVien(danhSachSV, maSV, tenSV, diemCC, diemGK, diemCK);

duyet(danhSachSV);

break;

}

case 6: {

// Xóa sinh vien theo ma sinh vien

int maSV;

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

scanf("%d", &maSV);

danhSachSV=xoaMaSV(danhSachSV, maSV);

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

duyet(danhSachSV);

break;

}

case 7: {

//Tim vi tri cua sinh vien có diem TB max.

if (danhSachSV == NULL) {

printf("Danh sach rong.\n");

} else {

int viTriMax = timViTriMax(danhSachSV);

printf("Sinh vien co diem trung binh cao nhat o vi tri %d\n", viTriMax);

}

break;

}

case 8:{

// Sap xep sinh vien.

danhSachSV = sapXepDanhSach(danhSachSV);

printf("Da sap xep tang dan danh sach sinh vien theo diem trung binh.\n");

duyet(danhSachSV);

break;

}

default:

printf("Lua chon khong hop le. Vui long nhap lai.\n");

break;

}

} while (choice != 0);

}

## Bài 2: Viết chương trình thực hiện sắp xếp một dãy gồm N (N ≤ 100) số nguyên yêu cầu cụ thể như sau:

1. Quick Sort: Sắp tăng dần theo trị tuyệt đối.
2. Shell Sort: Sắp tăng dần với bước chạy tùy chọn (nên lấy theo sách).
3. Heap Sort: Sắp giảm dần

**Code**:

#include <stdio.h>

#include <stdlib.h>

// Hàm in day so

void hienThi(int arr[], int n) {

printf("Danh sach sau khi sap xep: ");

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

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

}

printf("\n");

}

// Hàm hoán vi

void HoanVi(int\* a, int\* b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

// Hàm sap xep Quick Sort theo gia tri tuyet doi

void quickSort(int arr[], int low, int high) {

if (low < high) {

int pivot = arr[high];

int i = low - 1;

for (int j = low; j <= high - 1; j++) {

if (abs(arr[j]) < abs(pivot)) {

i++;

HoanVi(&arr[i], &arr[j]);

}

}

HoanVi(&arr[i + 1], &arr[high]);

int viTriPhanDoan = i + 1;

quickSort(arr, low, viTriPhanDoan - 1);

quickSort(arr, viTriPhanDoan + 1, high);

}

}

// Hàm sap xep ShellSort voi buoc nhay nhap vao ban phim.

void shellSort(int arr[], int n, int buocNhay) {

for (int gap = buocNhay; gap > 0; gap /= 2) {

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

int temp = arr[i];

int j;

for (j = i; j >= gap && arr[j - gap] > temp; j -= gap) {

arr[j] = arr[j - gap];

}

arr[j] = temp;

}

}

}

// hàm sx ho tro cho ham heapsort

void hamSXheapsortGiam(int arr[], int n, int i) {

int viTriNhoNhat = i;

int trai = 2 \* i;

int phai = 2 \* i + 1;

if (trai <= n && arr[trai] < arr[viTriNhoNhat]) {

viTriNhoNhat = trai;

}

if (phai <= n && arr[phai] < arr[viTriNhoNhat]) {

viTriNhoNhat = phai;

}

if (viTriNhoNhat != i) {

HoanVi(&arr[i], &arr[viTriNhoNhat]);

hamSXheapsortGiam(arr, n, viTriNhoNhat);

}

}

void heapSortGiam(int arr[], int n) {

for (int i = n / 2; i >= 1; i--) {

hamSXheapsortGiam(arr, n, i);

}

for (int i = n; i >= 1; i--) {

HoanVi(&arr[1], &arr[i]);

hamSXheapsortGiam(arr, i - 1, 1);

}

}

int main() {

int luaChon;

int soPhanTu;

do {

printf("\nMenu:\n");

printf("1. Quick Sort\n");

printf("2. Shell Sort\n");

printf("3. Heap Sort\n");

printf("4. Thoat\n");

printf("Chon lua cua ban: ");

scanf("%d", &luaChon);

printf("Nhap so phan tu cua mang: ");

scanf("%d", &soPhanTu);

int mang[soPhanTu + 1]; // M?ng b?t d?u t? index 1

printf("Nhap cac phan tu cua mang:\n");

for (int i = 1; i <= soPhanTu; i++) {

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

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

}

switch (luaChon) {

case 1:

quickSort(mang, 1, soPhanTu);

hienThi(mang, soPhanTu);

break;

case 2:

int buocChay; // Buoc chay

printf("Nhap buoc chay cho Shell Sort: ");

scanf("%d", &buocChay);

shellSort(mang, soPhanTu, buocChay);

hienThi(mang, soPhanTu);

break;

case 3:

heapSortGiam(mang, soPhanTu);

hienThi(mang, soPhanTu);

break;

case 4:

printf("Ket thuc chuong trinh.\n");

break;

default:

printf("Lua chon khong hop le. Vui long chon lai.\n");

}

} while (luaChon != 4);

}

1. **: Kết quả chạy chương trình của từng bài**

## Bài 1: Hỉnh ảnh chạy kết quả

1. Thêm ,sửa, xóa theo mã sinh viên ở vị trí bất kỳ

#### Thêm 4 sinh viên với đầy đủ thông tin nhập vào từ bàn phím

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Kết quả

A screenshot of a computer

Description automatically generated

* Thêm 2 sinh viên: “chỉ có mã sinh viên là “245” và “246” nhập vào bàn phím”

A screenshot of a computer

Description automatically generated

Kết quả:

A screenshot of a computer

Description automatically generated

* Sửa thông tin của 3 sinh viên : gồm 2 sv chỉ có mã sinh viên “245” , “246”(sinh viên chỉ có mã sinh viên ) , và 1 sinh viên có mã sv= “123” Ten = “Nguyen Van Linh” (sinh viên có đủ thông tin nhập vào từ ban đầu)

+Sửa thông tin sinh viên có mã “245”

A screenshot of a computer

Description automatically generated

+Sửa thông tin sinh vien có ma “246”

A screenshot of a computer

Description automatically generated

+Sửa thông tin sinh viên có mã “123” ( sinh viên có đủ thông tin nhập vào từ ban đầu )

A screenshot of a computer

Description automatically generated

* Xóa sinh viên có mã là “245”

A screenshot of a computer

Description automatically generated

1. Tính điểm trung bình của mỗi sinh viện như sau điểm chuyên cần hệ số 2, điểm giữa kỳ hệ số 2, điểm cuỗi kỳ hệ số 6 .

A screenshot of a computer

Description automatically generated

1. Tìm vị trí của sinh viên có điểm trung bình lớn nhất.

A screenshot of a computer

Description automatically generated

1. Sắp xếp tăng dần danh sách sinh viên theo điểm trung bình.

**A screenshot of a computer

Description automatically generated**

## Bài 2: Hình ảnh chạy kết quả

1. **Sắp** xếp Quick Sort tăng dần theo trị tuyệt đối

* **Sử dụng với mảng có số nguyên âm và số nguyên dương**

**A screen shot of a computer

Description automatically generated**

* **Sử dụng với mảng không có số nguyên âm :**

**A screen shot of a computer

Description automatically generated**

1. Shell Sort: Sắp tăng dần với bước chạy tùy chọn nhập vào từ bàn phím

-Bước nhảy là 1:

A computer screen shot of a black screen

Description automatically generated

- Bước nhảy 3

A computer screen with white text

Description automatically generated

1. HeapSort Sắp xếp giảm dần .

A screen shot of a computer

Description automatically generated