**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CƠ SỞ**

**ĐỀ TÀI 704**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ ĐIỂM CỦA SINH VIÊN THEO TỪNG MÔN HỌC**

**SINH VIÊN THỰC HIỆN**

1. **Lê Hoàng Ngọc Hân Lớp : 19TCLCDT4 Nhóm : 19.10A**
2. **Trương Thị Mỹ Duyên Lớp : 19TCLCDT4 Nhóm : 19.10A**

**GIÁO VIÊN HƯỚNG DẪN**

**TRẦN HỒ THỦY TIÊN**

**Đà Nẵng, 5-2020**

LỜI MỞ ĐẦU

Ngày nay khoa học ngày càng phát triển, con người đang sống trong thời đại kĩ thuật số 4.0. Việc hiện đại hóa, máy móc hóa các công việc hằng ngày của con người là vô cùng cần thiết và ngày càng nhiều. Máy tính là một trong những công cụ hiện đại mà con người dùng để làm việc. Một trong những việc mà máy tính có thể làm được là nhập thông tin, xử lí và lưu trữ. “Quản lí điểm sinh viên ” nhóm em chọn là một trong những đề tài hay và cần thiết để thực hiện trong tương lai.

Từ khi bắt đầu làm đồ án cơ sở đến nay, nhóm chúng em đã nhận được nhiều sự quan tâm giúp đỡ của cô Trần Hồ Thủy Tiên. Với lòng biết ơn sâu sắc, chúng em gửi đến cô lời cảm ơn vì đã giành tâm huyết của mình truyền đạt kiến thức cho chúng em. Trong thời gian làm đồ án, nhóm chúng em đã nỗ lực làm việc, nhưng sẽ không tránh khỏi thiếu sót vì vậy chúng em rất mong được thầy cô trong khoa Công nghệ Thông tin tận tình giúp đỡ.

Sinh viên thực hiện

TRƯƠNG THỊ MỸ DUYÊN

**LÊ HOÀNG NGỌC HÂN**

MỤC LỤC

[**LỜI MỞ ĐẦU**](#_30j0zll)……………………………………………………………………....2

[**MỤC LỤC**](#_1fob9te)……………………………………………………….……….…………3

[**DANH MỤC HÌNH VẼ**](#_tyjcwt)……………………………………….…………………...4

[**DANH MỤC BẢNG**](#_tyjcwt)…..……………………………………….…………………...4

**1**[**. GIỚI THIỆU ĐỀ TÀI**](#_4d34og8)**.**……………………………………………………….......5

**2**[**. CƠ SỞ LÝ THUYẾT**](#_2s8eyo1)……………………………………………………….........5

2[.1. Ý tưởng](#_1hmsyys)……………………………………………………………………….5

2[.2. Cơ sở lý thuyết](#_17dp8vu)……………………..…………………………………………5

**3**[**. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN**](#_41mghml)………………….........8

3[.1. Phát biểu bài toán](#_26in1rg)……………………………………………………..............8

3[.2. Cấu trúc dữ liệu](#_lnxbz9)……………………………………………...……………......9

3[.3. Thuật toán](#_44sinio)………………………………………………………….…..........11

**4**[**. CHƯƠNG TRÌNH VÀ KẾT QUẢ**](#_2jxsxqh)…………………………………….…........12

4[.1. Tổ chức chương trình](#_z337ya)………………………………………………...……...12

4[.2. Kết quả](#_2grqrue)………………………………………………………………………17

4[.2.1. Giao diện chính của chương trình](#_vx1227)………………………………............17

4[.2.2. Kết quả thực thi của chương trình](#_3fwokq0)……………………………….….......19

4[.2.3. Nhận xét](#_3j2qqm3)…………..……………………………………………….........22

**5.** [**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**](#_1y810tw)……………………………….…....23

5.1[. Kết luận](#_4i7ojhp)………………………………………………………………….......23

5.2. [Hướng phát triển](#_1ci93xb)………...………………………………………………......23

[**TÀI LIỆU THAM KHẢO**](#_3whwml4)**…**……………………………...……………………...24

[[1] Tên tác giả, Tên tài liệu, Tên nhà xuất bản, năm xuất bản](#_3as4poj)……………………..24

[[2] Tên chủ sở hữu, Tên bài viết, url, ngày truy cập](#_1pxezwc)……………………………….24

DANH MỤC HÌNH VẼ

***Hình 3.1.4.*** Mô tả chương trình quản lí điểm sinh viên ………………………..10

***Hình 3.3.1.*** Sơ đồ khối thuật toán chương trình…………………...…….………13

***Hình 4.1.*** Minh hoạ tổ chức chương trình quản lí điểm sinh viên..……………13

***Hình 4.2.1.1.***Nhập sai tên đăng nhập………...……………….………………..18

***Hình 4.2.1.2.*** Nhập sai mật khẩu………………….…………………………….18

***Hình 4.2.1.3.*** Chương trình bắt đầu thực hiện…….…………………………….19

***Hình 4.2.1.4.*** Giao diện Menu……….…...……………………………..……....19

***Hình 4.2.2.1.***Giao diện chức năng nhập điểm…………………………………..20

***Hình 4.2.2.2.*** Giao diện chức năng xem điểm…………………………………..20

***Hình 4.2.2.3.*** Giao diện chức năng sửa điểm... ………………………………...21

***Hình 4.2.2.4.*** Giao diện chức năng sắp xếp………….………………………....22

***Hình 4.2.2.5.*** Giao diện chức năng in file điểm…………...……………………23

***Bảng 3.3.1.*** Độ phức tạp thuật toán……………………………...…………….11

***Bảng 4.1.1.*** Chức năng các hàm………...…………………………………..…13

DANH MỤC BẢNG

***Bảng 3.1.*** Bảng độ phức tạp thuật toán các hàm………………………………11

***Bảng 4.1.*** Bảng chức năng các hàm………...………………………………….13

# GIỚI THIỆU ĐỀ TÀI

Công nghệ đang ngày càng phát triển và lượng thông tin đang mỗi ngày một tăng lên với tốc độ cao. Đồng thời, việc lưu giữ thông tin bằng giấy tờ là điều rất bất cập, tốn kém (về thời gian, vật chất) và hơn nữa, thông tin luôn luôn được sửa đổi, bổ sung và mở rộng theo thời gian. Chính vì vậy, trong môi trường giáo dục, để tối ưu hóa việc ghi chép, tính toán và sửa đổi thông tin điểm, chúng ta sử dụng những phần mềm, chương trình quản lí điểm phù hợp với lượng dữ liệu và các chức năng cần thiết như Excel. Báo cáo này sẽ trình bày chương trình quản lí điểm của sinh viên trong ngôn ngữ C.

# CƠ SỞ LÝ THUYẾT

## 2.1. Ý tưởng

Dùng ngôn ngữ C để thực hiện các chức năng quản lí điểm sinh viên. Quản lí tên, mã sinh viên, các cột điểm, xếp loại, sắp xếp giảm dần theo điểm trung bình, sửa điểm,… Quản lí nhiều môn học, mỗi môn học được lưu trữ trên một file, thực hiện các chức năng kể trên.

## 2.2. Cơ sở lý thuyết

**2.2.1. File**

* ***Tác dụng của file:***
* Dữ liệu được lưu ở biến của chương trình, và nó sẽ biến mất khi chương trình kết thúc. Sử dụng file để lưu trữ dữ liệu cần thiết để đảm bảo dữ liệu của chúng ta không bị mất ngay cả khi chương trình của chúng ta ngừng chạy.
* Nếu chương trình của bạn có đầu vào(input) là lớn, bạn sẽ rất vất vả nếu phải nhập mỗi khi chạy. Thay vào đó, hãy lưu vào file và chương trình của bạn sẽ tự đọc mỗi lần khởi chạy
* Dễ dàng sao chép, di chuyển dữ liệu giữa các thiết bị với nhau
* ***Các kiểu file:***
* File văn bản
* File nhị phân
* ***Thao tác trên file:***
* ***Mở file:***
* Mở file tên sinh viên: Chương trình thực hiện việc mở file ghi danh sách tên sinh viên để đọc thông tin ra.

*ften=fopen("C:\\Users\\PC\\Desktop\\project\\ten.txt","r");*

* Mở file mã sinh viên: Chương trình thực hiện việc mở file ghi danh sách mã sinh viên để đọc thông tin ra.

*fmsv=fopen("C:\\Users\\PC\\Desktop\\project\\msv.txt","r");*

* Mở file điểm sinh viên: Chương trình thực hiện việc mở file ghi điểm sinh viên để ghi thông tin vào.

*fd=fopen("C:\\Users\\PC\\Desktop\\project\\diem13.txt","w");*

* ***Đọc file:***
* Đọc tên sinh viên: Chương trình đọc dữ liệu tên sinh viên từ file ten.txt sau đó truyền dữ liệu vào mảng tensv[][40].

*fgets(ten[i],100,ften);*

* Đọc mã sinh viên: Chương trình đọc dữ liệu tên sinh viên từ file msv.txt sau đó truyền dữ liệu vào mảng msv[][40].

*fscanf(fmsv,"%s",&msv[i]);*

* Đọc điểm sinh viên: Chương trình đọc dữ liệu từ file diem13.txt sau đó truyền dữ liệu vào mảng diem[40][6].

*fscanf(fd,"%f",&diem[i][j]);*

* ***Ghi file:***
* Ghi điểm sinh viên: Chương trình thực hiện việc nhập điểm sinh viên từ bàn phím sau đó tính điểm trung bình và đồng thời xếp loại sinh viên sau đó ghi điểm vào file diem13.txt, d

*fprintf(fd,"%.2f\n",sv.dtb);*

* ***Đóng file:***
* Đóng file tên sinh viên:

*fclose(ften);*

* Đóng file mã sinh viên:

*fclose(fmsv);*

* Đóng file điểm sinh viên:

*fclose(fd);*

**2.2.2. Mảng:**

* ***Định nghĩa:***

***Mảng:*** là một loại cấu trúc dữ liệu trong [ngôn ngữ lập trình C/C++](https://quantrimang.com/cplusplus), lưu trữ một tập hợp tuần tự các phần tử cùng kiểu với độ dài cố định. Mảng thường được sử dụng để lưu trữ tập hợp dữ liệu, nhưng nó cũng hữu dụng khi dùng để lưu trữ một tập hợp biến có cùng kiểu.

* ***Cách khai báo:***

*Kieutenmang[Kichthuocmang];*

* ***Thao tác trên mảng :***
* ***Mảng một chiều:*** Dùng mảng một chiều để thực hiện việc lưu trữ dữ liệu từ file tensv vào mảng tensv[40] với kiểu dữ liệu là char[30] và file msv[40] với kiểu dữ liệu là char[15]. Sau đó chương trình sẽ làm việc trên mảng.

*char (msv[100])[100];*

*char (ten[100])[100];*

* ***Mảng hai chiều:*** Dùng mảng hai chiều để thực hiện việc lưu trữ dữ liệu từ file diemsv[40][5] với kiều dữ liệu float 5.2.

*float diem[40][5];*

**2.2.3. Struct:**

* ***Định nghĩa:*** Struct là một kiểu dữ liệu người dùng tự định nghĩa (user defined datatype) cho phép lưu trữ các loại phần tử khác nhau. Mỗi phần tử bao gồm nhiều trường khác nhau với các kiểu dữ liệu khác nhau.Các biến khai báo trong struct được gọi là phần tử của cấu trúc hoặc thành phần của cấu trúc, còn được gọi tắt là các trường cấu trúc.
* ***Cách khai báo:***

struct tencautruc{

kieudulieu ten;

kieudulieu ten;

……

}

typedef struct tencautruc ten;

* Thay thế tên gọi struct bằng hàm typedef.
* Gọi lại struct bằng tên gọi của nó (struct tencautruc) hoặc bằng tên thay thế (ten).
* Chức năng: Khai báo một dữ liệu bằng cách gộp nhiều kiểu dữ liệu với nhau thông qua các trường cấu trúc.
* ***Thao tác trên struct:***

struct sinhvien {

float lab;

float pres;

float mid;

float final;

float dtb;

};

typedef struct sinhvien diem;

**2.2.4. Mảng song song:**

* ***Đinh nghĩa:*** là sự kết hợp nhiều lần của mảng 1 chiều, có cách lưu trữ các phần tử như một bảng (vì các phần tử của các mảng khác nhau khi có chung một chỉ số sẽ chỉ đến cùng một đối tượng).
* ***Tác dụng:*** Mỗi sinh viên đều có thông tin được thể hiện như một struct bao gồm các trường : họ tên, mã sinh viên, các cột điểm, … Dùng mảng song song có thể dễ dàng thể hiện và kiểm soát được một lúc nhiều sinh viên.
* ***Thao tác trên mảng song song:***

void nhapN(diem sv[],int n,char (msv[100])[100] ,char (ten[100])[100],FILE \*fd){

printf("\nNhap diem cho msv %s : \n",msv[i]);

nhap(sv[i],fd);

}

# TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

## 3.1. Phát biểu bài toán

**3.1.1 Mô tả:** Chương trình viết bằng ngôn ngữ C cho phép quản lí tài khoản của 40 sinh viên/ môn. Gồm có Menu như sau:

1 - Nhập điểm

2 - Xem điểm

3 - Sửa điểm

4 - Sắp xếp điểm

5 - In danh sách điểm của tất cả sinh viên

0 - Thoát

**Trong đó:**

- Mỗi môn học sẽ được lưu trên 1 file.

- **Chức năng nhập điểm:** Nhập điểm cho toàn bộ sinh viên theo thứ tự các cột điểm sau: Laboratory (lab), Presentation (pres), Mid - Semester (mid), Final - Semester (final) và chương trình sẽ tính toán thêm 1 cột điểm Average (avg) theo công thức sau:

avg = 0.1\*lab + 0.2\*pres + 0.2\*mid + 0.5\*final

- **Chức năng xem điểm:** Nhập vào mã sinh viên cần xem điểm. Nếu tồn tại mã sinh viên thì hiển thị tất cả các cột điểm của sinh viên đó, ngược lại thì hiển thị thông báo mã sinh viên không tồn tại.

- **Chức năng sửa điểm:** Nhập vào mã sinh viên cần sửa điểm. Nếu tồn tại thì thực hiện:

+ Nhập cột điểm cần sửa theo menu sau:

1. Lab
2. Pres
3. Mid
4. Final

+ Nhập điểm mới.

+ Hiển thị ra điểm sau khi sửa.

Ngược lại, hiển thị mã sinh viên không tồn tại.

-  **Chức năng sắp xếp:** Tính điểm trung bình, quy về điểm theo hệ A, B, C, D rồi sắp xếp theo thứ tự giảm dần. In ra màn hình bảng xếp hạng gồm: Mã sinh viên, tên sinh viên, điểm cột avg và điểm hệ A, B, C, D được quy đổi từ avg.

**3.1.2. Đầu vào (Input):**

**-** Mã sinh viên.

- Họ tên sinh viên.

- Điểm các cột lab, pres, mid, final.

**3.1.3 Đầu ra (Output):**

- Điểm trung bình (avg).

- Điểm theo thang A, B, C, D.

- Danh sách điểm của 1 hoặc nhiều sinh viên .

- Danh sách điểm của sinh viên sau khi sắp xếp.

- Điểm sinh viên được sửa, file kết quả chương trình.

**3.1.4. Chương trình tổng quát**

|  |  |
| --- | --- |
|  |  |
|  |  |

*Hình 3.1.4 Mô tả chương trình quản lí điểm sinh viên*

## 3.2. Cấu trúc dữ liệu

### 3.2.1. Kiểu dữ liệu cơ bản

int n, chon; // Kiểu số nguyên

diem sv[10]; // Mảng kiểu struct

char key[20], key1[20]; // Kiểu kí tự

char (msv[100])[100]; // Mảng kiểu kí tự

char (ten[100])[100]; // Mảng kiểu kí tự

float diem[40][5], dtb[40]; // Mảng kiểu số thực (1-2 chiều)

### 3.2.2. Kiểu dữ liệu kết hợp: Sử dụng các mảng một chiều và mảng song song

* Struct

struct sinhvien {

float lab;

float pres;

float mid;

float final;

float dtb;

};

* Mảng một chiều

void mangmsv(int n, char msv[][100],FILE \*fmsv)

void mangten(int n, char ten[][100],FILE \*ften)

* Mảng hai chiều

void mangdiem(int n, float diem[][5],FILE \*fd, float dtb[]) {

* Mảng song song

void nhapN(diem sv[],int n,char (msv[100])[100] ,char (ten[100])[100],FILE \*fd) // Nhap diem n SV vao file

- Các mảng mangten, mangmsv, mangdiem là các mảng song song có chức năng lưu trữ thông tin sau khi đọc từ file dữ liệu. Trong đó, mangdiem chứa nhiều phần tử mà mỗi phần tử là một struct bao gồm điểm các cột: Lab, Presentation (pres), Mid-Semester, Final-Semester và Average (avg).

### 3.2.3.Lưu trữ dữ liệu trong các file

C:\\Users\\PC\\Desktop\\project\\msv.txt (File lưu mã sinh viên)

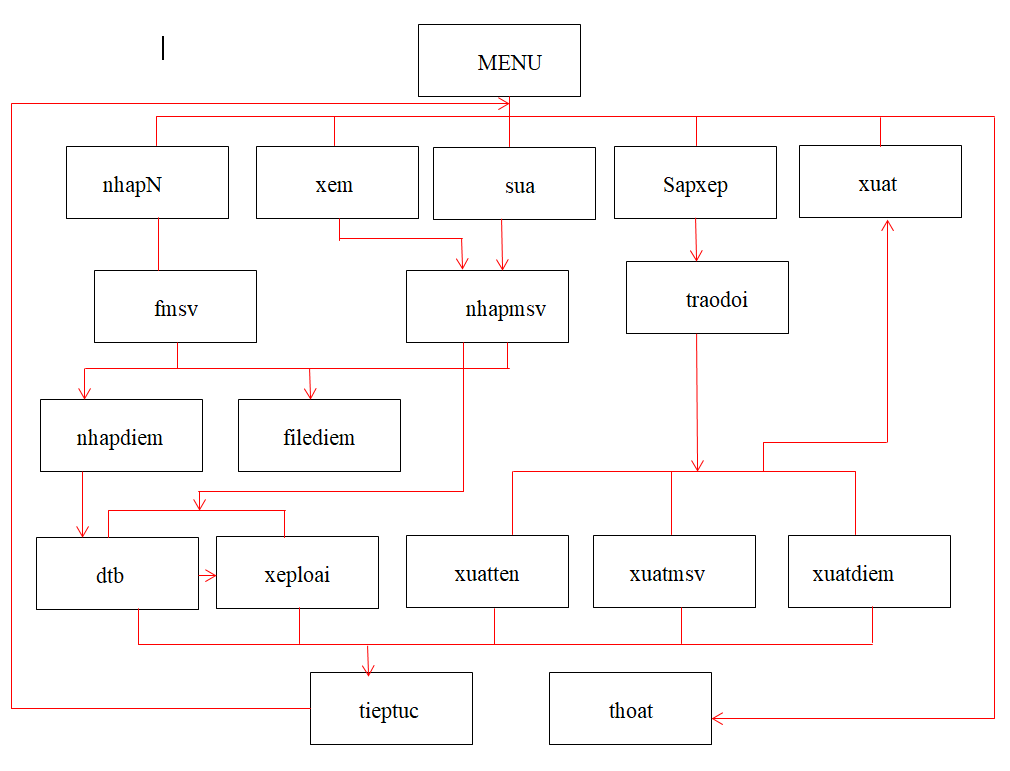
C:\\Users\\PC\\Desktop\\project\\ten.txt (File lưu tên)

C:\\Users\\PC\\Desktop\\project\\diem13.txt (File lưu điểm)

## 3.3.Thuật toán

Trình bày các thuật toán (sử dụng giả lệnh hoặc sơ đồ khối) và phân tích độ phức tạp của các thuật toán.

* ***Sơ đồ khối của thuật toán:***

 *Hình 3.3.1. Sơ đồ khối thuật toán chương trình*

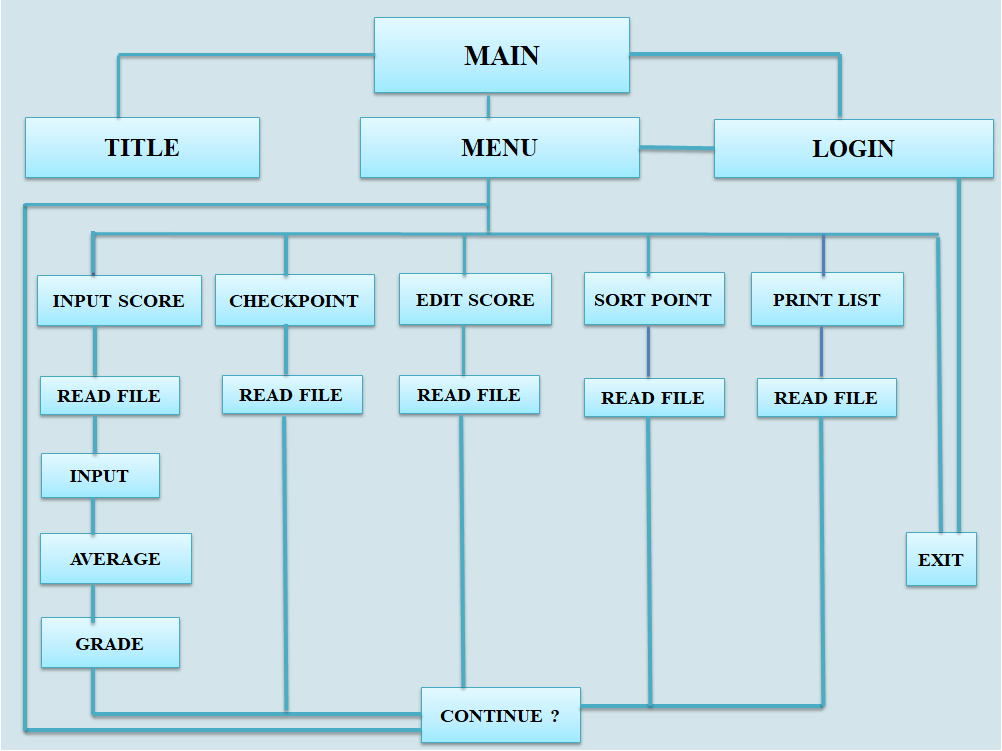
* ***Độ phức tạp thuật toán :***

*Bảng 3.1. Bảng độ phức tạp thuật toán của các hàm*

|  |  |
| --- | --- |
| **Tên Hàm** | **Độ phức tạp thuật toán** |
| void Title() | O(1) |
| void Login() | O(1) |
| void Menu() | O(1) |
| void mangmsv(int n, char msv[][100],FILE \*fmsv) | O(n) |
| void DTB(diem &sv) | O(1) |
| void nhap(diem &sv,FILE \*fd) | O(n) |
| void nhapN(diem sv[],int n,char (msv[100])[100],char (ten[100])[100],FILE \*fd) | O(n) |
| char xeploai(float x) | O(1) |
| void xem(char (msv[100])[100], char (ten[100])[100], int n, float diem[][5],char key[]) | O(n) |
| void sua(FILE \*fd, char (msv[100])[100],char (ten[100])[100],int n, float diem[][5],char key1[]) | O(n) |
| void swap(float &a,float &b) | O(1) |
| void traodoi(char a[],char b[]) | O(1) |
| void sapxep(char (msv[100])[100],int n,char (ten[100])[100],float dtb[]) | O(n2) |
| void xuat(int n, char(ten[100])[100], char(msv[100])[100], float diem[][5]) | O(n) |
| int main() | O(n2) |

# 4. CHƯƠNG TRÌNH VÀ KẾT QUẢ

## 4.1 Tổ chức chương trình



*Hình 4.1. Minh hoạ tổ chức chương trình quản lí điểm sinh viên*

- Chương trình gồm có:

+ Một struct sinhvien gồm điểm các cột Lab, Pres, Mid, Final và Avg.

+ Các hàm thực hiện chức năng của chương trình.

*Bảng 4.1. Chức năng các hàm*

|  |  |
| --- | --- |
| Tên Hàm | Chức năng (Ghi chú) |
| 1. void Title() | Hàm in thông tin đồ án |
| 1. void Login() | Hàm đăng nhập |
| 1. void Menu() | Hàm in Menu |
| 1. void mangmsv(int n, char msv[][100],FILE \*fmsv) | Hàm đọc mã SV từ file vào mảng 1 chiều |
| 1. void mangten(int n, char ten[][100],FILE \*ften) | Hàm đọc họ và tên SV từ file vào mảng 1 chiều |
| 1. void mangdiem(int n, float diem[][5],FILE \*fd, float dtb[]) | Hàm đọc điểm SV từ file vào mảng 2 chiều |
| 1. void DTB(diem &sv) | Hàm tính điểm trung bình |
| 1. void nhap(diem &sv,FILE \*fd) | Hàm nhập điểm 1 SV vào file |
| 1. void nhapN(diem sv[],int n,char (msv[100])[100] ,char (ten[100])[100],FILE \*fd) | Hàm nhập điểm n SV vào file |
| 1. char xeploai(float x) | Hàm quy đổi điểm hệ 10 sang A,B,C,D |
| 1. void xem(char (msv[100])[100], char (ten[100])[100], int n, float diem[][5],char key[]) | Hàm xem các cột điểm của 1 SV với mã SV được nhập từ bàn phím |
| 1. void sua(FILE \*fd, char (msv[100])[100],char (ten[100])[100],int n, float diem[][5],char key1[]) | Hàm sửa điểm theo cột của SV với mã SV được nhập từ bàn phím |
| 1. void swap(float &a,float &b) | Hàm tráo đổi 2 số thực |
| 1. void traodoi(char a[],char b[]) | Hàm tráo đổi 2 xâu kí tự |
| 1. void sapxep(char (msv[100])[100],int n,char (ten[100])[100],float dtb[]) | Hàm sắp xếp điểm trung bình theo thứ tự giảm dần |
| 1. void xuat(int n, char(ten[100])[100], char(msv[100])[100], float diem[][5]) | Hàm in danh sách sinh viên |
| 1. int main() | Hàm chương trình chính gọi các hàm con trên |

* ***Hàm mangmsv***
* **Input:** int n, FILE \*fmsv
* **Output:** msv[][100] (mảng 1 chiều lưu mã sinh viên)
* ***Thuật toán:***

**Bước 1:** Mở file msv

**Bước 2:** Lặp i: 1-> n

**Bước 3:** Ghi msv của mỗi sinh viên từ FILE fmsv vào mảng msv[][100]

**Bước 4:** Kết thúc.

* ***Hàm mangten***
* **Input:** int n,FILE \*ften.
* **Output:** ten[][100] (mảng 1 chiều lưu họ tên sinh viên)
* ***Thuật toán:***

**Bước 1:** Mở FILE ften.

**Bước 2:** Lặp i: 1-> n

**Bước 3:** Ghi tên mỗi sinh viên từ FILE ften vào mảng ten[][100].

**Bước 4:** Kết thúc.

* ***Hàm mangdiem***
* **Input:** int n, FILE \*fd, float dtb[].
* **Output:** dtb[], float diem[][5] (mảng 2 chiều lưu điểm các cột của sinh viên bao gồm đã điểm trung bình vừa được tính)
* ***Thuật toán:***

**Bước 1:** Mở FILE fd.

**Bước 2:** Lặp i: 1-> n

**Bước 3:** Ghi điểm sinh viên từ file vào mảng 2 chiều diem[][100], mỗi hàng tương ứng với điểm của một sinh viên.

**Bước 4:** Kết thúc

* ***Hàm DTB***
* **Input:** diem &sv (gồm điểm lab, pres, mid, final)
* **Output:** diem sv.dtb (điểm trung bình)
* ***Thuật toán:***

**Bước 1:** Tính sv.dtb bằng công thức *sv.dtb=0.1\*sv.lab+0.1\*sv.pres+0.3\*sv.mid+0.5\*sv.final;*

**Bước 2:** Kết thúc.

* ***Hàm nhap***
* **Input:** Điểm các cột theo thứ tự: lab, pres, mid, final
* **Output:** File fd lưu điểm của một sinh viên gồm: lab, pres, mid, final, avg (điểm của mỗi sinh viên là một dòng)
* ***Thuật toán:***

**Bước 1:** Nhập điểm Lab (0<=Lab<=10).

**Bước 2:** Nhập điểm Pres(0<=Pres<=10).

**Bước 3:** Nhập điểm Mid(0<=Mid<=10).

**Bước 4:** Nhập điểm Final(0<=Final<=10).

**Bước 5:** Kết thúc.

* ***Hàm nhapN***
* **Input:** struct diem sv[] để ghi điểm của mỗi sinh viên vào FILE fd, msv[] và ten[] để hiển thị thông tin sinh viên ra màn hình.
* **Output:** File fd lưu điểm của 40 sinh viên gồm: lab, pres, mid, final, avg (điểm của mỗi sinh viên là một dòng)
* ***Thuật toán:***

**Bước 1:** Mở FILE fd để ghi thông tin điểm của 40 sinh viên vào file.

**Bước 2:** Lặp i -> n.

**Bước 3:** Thực hiện hàm **nhap**

**Bước 4:** Kết thúc.

* ***Hàm xeploai***
* **Input:** điểm trung bình của sinh viên.
* **Output:** xếp loại của sinh viên.
* ***Thuật toán:***

**Bước 1:** Kiểm tra sv.dtb, nếu thuộc (8<=sv.dtb<=10 ) xuất ‘A’, nếu thuộc (6.5<=sv.dtb<8 ) xuất ‘B’, nếu thuộc (5<=sv.dtb<6.5 ) xuất ‘C’, nếu không xuất ‘D’.

**Bước 2:** Kết thúc.

* ***Hàm xem***
* **Input:** Mã sinh viên của sinh viên cần xem điểm.
* **Output:** Thông tin tên sinh viên và điểm của sinh viên có mã nhập vào
* ***Thuật toán:***

**Bước 1:** Nhập mã sinh viên cần tìm.

**Bước 2:** So sánh mã sinh viên nhập vào với các mã sinh viên trong mảng msv[].

**Bước 3:** Xuất tên, điểm, xếp loại của mã sinh viên nhập vào.

**Bước 4:** Kết thúc.

* ***Hàm sua***
* **Input:** Mã sinh viên cần sửa điểm, cột điểm cần sửa.
* **Output:** Cập nhật điểm của sinh viên sau khi sửa điểm.
* ***Thuật toán:***

**Bước 1:** Nhập mã sinh viên cần sửa điểm.

**Bước 2:**  So sánh mã sinh viên nhập vào với mảng msv[] -> chỉ số của đối tượng cần sửa.

**Bước 3:** Chọn cột điểm sinh viên cần sửa.

**Bước 4:** Nhập điểm mới.

**Bước 5:** Tính lại điểm trung bình, lưu điểm mới vào file.

**Bước 6:** Kết thúc.

* ***Ham xuat***
* **Input:** Các mảng chứa thông tin sinh viên msv[], ten[], diem[][], dtb[], xeploai[].
* **Output:** Thông tin của 40 sinh viên bao gồm tên, mã sinh viên, điểm, điểm trung bình, xếp loại.
* ***Thuật toán:***

**Bước 1:** Lặp i->n.

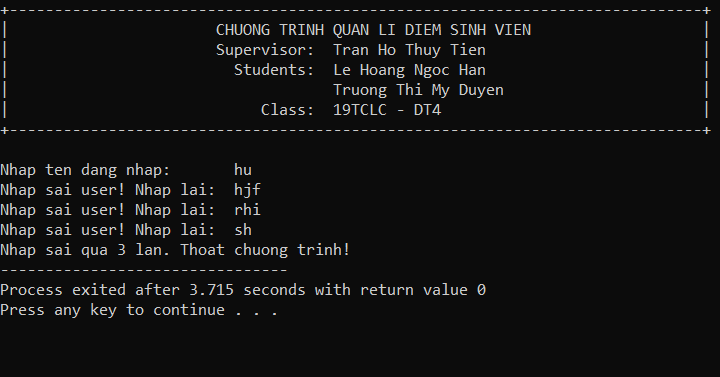
**Bước 2:** Xuất thông tin của mỗi sinh viên.

**Bước 3:** Kết thúc.

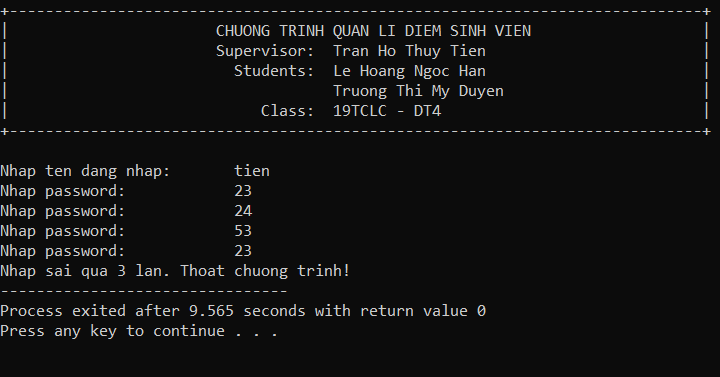
## 4.2. Kết quả

**4.2.1. Giao diện chính của chương trình**

* ***Khi đăng nhập***

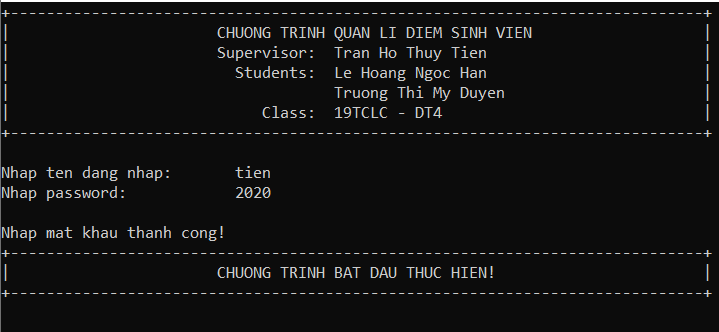


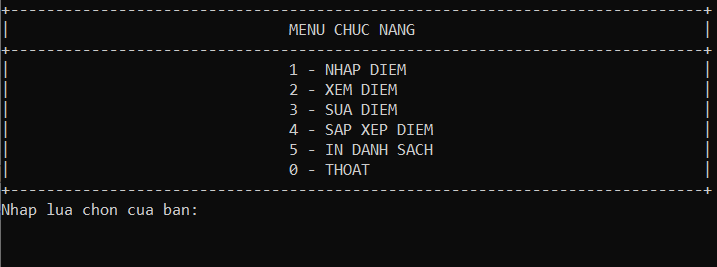
*Hình 4.2.1.1. Nhập sai tên đăng nhập quá 3 lần, màn hình sẽ hiển thị thông báo và thoát chương trình*



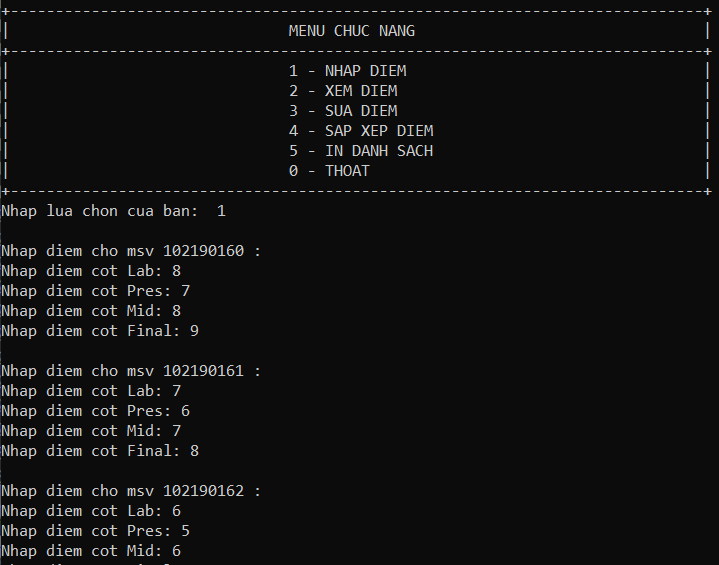
*Hình 4.2.1.2 Tương tự, nhập sai mật khẩu quá 3 lần, màn hình sẽ hiển thị thông báo và thoát chương trình*

* ***Khi đăng nhập thành công***

*Hình 4.2.1.3 Chương trình bắt đầu thực hiện.*

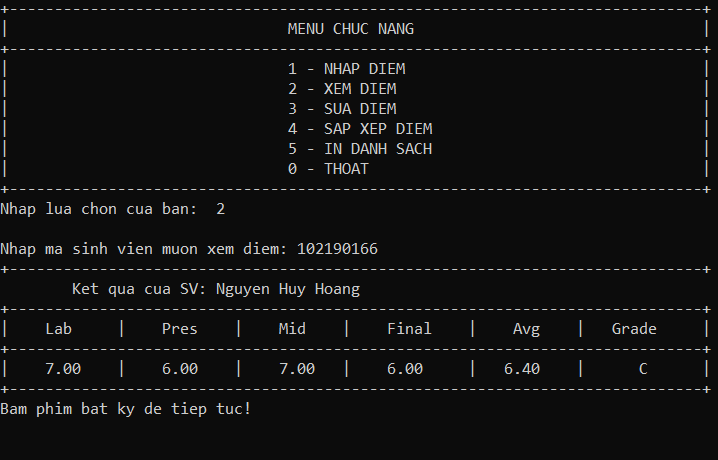
*Hình 4.2.1.4 Sau khi nhấn Enter, màn hình sẽ hiển thị Menu.*

**4.2.2. Kết quả thực thi chương trình**

* ***Nhập điểm*** 

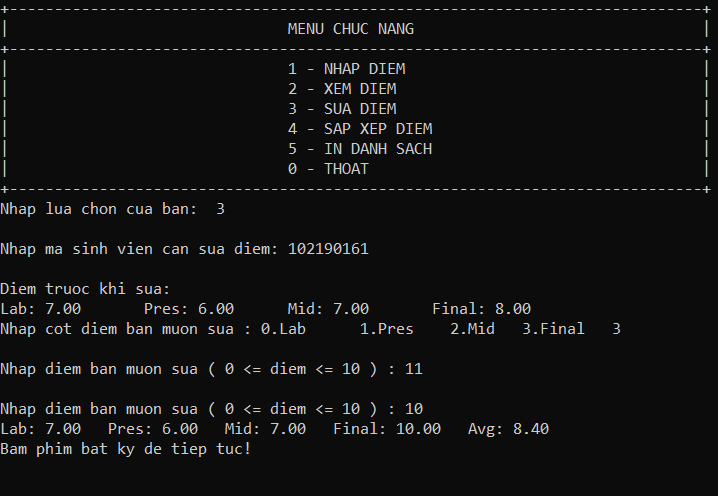
*Hình 4.2.2.1 Giao diện chức năng nhập điểm*

* ***Xem điểm***



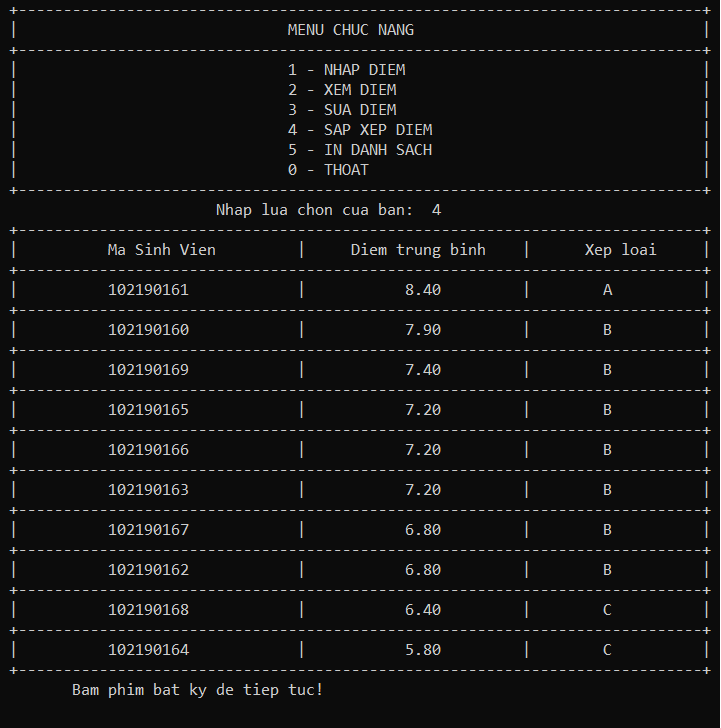
*Hình 4.2.2.2 Giao diện chức năng xem điểm*

* ***Sửa điểm***



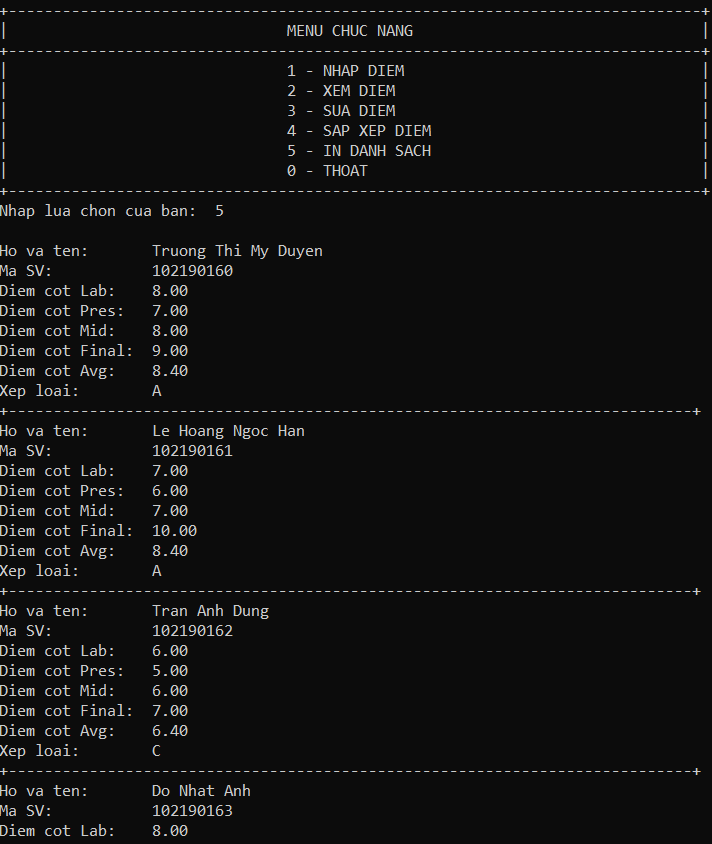
*Hình 4.2.2.3 Giao diện chức năng sửa điểm*

* ***Sắp xếp***



*Hình 4.2.2.4. Giao diện chức năng sắp xếp*

* ***In danh sách***



*Hình 4.2.2.5. Giao diện chức năng in danh sách*

### 4.2.3. Nhận xét

1. ***Ưu điểm***

* Cập nhật, tìm kiếm và sửa chữa điểm nhanh chóng, thuận tiện, chính xác.
* Bố cục, giao diện rõ ràng, thân thiện với người dùng.

1. ***Nhược điểm***

* Yêu cầu người dùng phải nhập thông tin cẩn thận.
* Giao diện, màu sắc chưa bắt mắt.
* Một số thuật toán (sắp xếp,…) chưa được tối ưu hóa.

# 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết luận

## 5.1.1. Đối với mục tiêu đặt ra của đề án:

* Hiểu và áp dụng được mảng song song , cách lưu trữ dữ liệu trong file và đọc thông tin từ file .
* Áp dụng được các thuật toán thao tác trên các mảng song song.
* Hiểu và sử dụng các hàm xử lý chuỗi .
* Chia nhỏ chương trình thành các hàm con để dễ dàng cho việc quan sát và chỉnh sửa.
* Viết hàm menu và sửa điểm bằng cách sử dụng switch - case để nhập lựa chọn
* Biết cách căn chỉnh màn hình giao diện .

**5.1.2** **Đối với chương trình quản lí điểm:**

* Đáp ứng đầy đủ các chức năng của một chương trình quản lí điểm: Nhập điểm, xem điểm, sửa điểm, sắp xếp điểm, in danh sách.
* Có khả năng bảo mật nhờ có hàm Login. Tuy nhiên, hàm vẫn còn hạn chế về số lượng account và độ bảo mật thấp.
* Áp dụng được nhiều thao tác trên mảng, file,… Tuy nhiên, một số thuật toán chưa được tối ưu hóa tối đa.

5.2. Hướng phát triển

* Bổ sung thêm nhiều dữ liệu như số lượng sinh viên, số lượng môn, các cột điểm rèn luyện.
* Phát triển thuật toán để thực hiện nhiều chức năng hơn như tìm danh sách sinh viên có điểm trung bình thuộc khoảng [a,b], thực hiện kết thúc chương trình khi hiệu lệnh nào đó (ví dụ nhập 4 điểm 0).
* Tối ưu hóa thuật toán sắp xếp, sửa điểm và nâng cao bảo mật.
* Thiết kế giao diện cho chương trình bằng các phần mềm đồ họa chuyên dụng khác.
* Sử dụng danh sách liên kết và struct thay cho các mảng song song.

TÀI LIỆU THAM KHẢO

***[1] Tên tác giả, Tên tài liệu, Tên nhà xuất bản, năm xuất bản***

1. *Đỗ Xuân Lôi, Cấu trúc dữ liệu và giải thuât, NXB Khoa học và kĩ thuật, 2003*
2. *Nguyễn Hồng Chương, Cấu trúc dữ liệu ứng dụng và cài đặt bằng C, NXB TPHCM, 2003*
3. *Lê Xuân Trường, Cấu trúc dữ liệu bằng ngôn ngữ C, NXB Thống kê, 1999*
4. *Giáo trình Cấu trúc dữ liệu - Thầy Phan Thanh Tùng - ĐH Bách Khoa Đà Nẵng.*

***[2] Tên chủ sở hữu, Tên bài viết, url, ngày truy cập***

1. Bài 70. Đọc ghi file trong C

***<https://nguyenvanhieu.vn/doc-ghi-file-trong-c/>***

1. File và các thao tác cơ bản của File trong C

[***https://daynhauhoc.com/t/file-va-cac-thao-tac-co-ban-voi-file-trong-c/35276***](https://daynhauhoc.com/t/file-va-cac-thao-tac-co-ban-voi-file-trong-c/35276)

1. Bài 13: Kiểu cấu trúc - Struct

[***https://cachhoc.net/2014/12/19/lap-trinh-c-bai-11-kieu-cau-truc-struct/***](https://cachhoc.net/2014/12/19/lap-trinh-c-bai-11-kieu-cau-truc-struct/)