

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC



BÁO CÁO BÀI TẬP LỚN
KỸ THUẬT LẬP TRÌNH

Giảng viên hướng dẫn : Nguyễn Thị Thanh Huyền
Nhóm : 24
Thành viên : Thịnh Xuân Đông - 20195855
Phạm Đình Thông - 20195923
Lớp : Toán Tin 01 - K64

Hà Nội, tháng 06/2020

MỤC LỤC

| | |
|---|----|
| Phần mở đầu..... | 0 |
| Phần nội dung..... | 1 |
| I. Phát triển chương trình..... | 1 |
| Bước 0 :..... | 1 |
| Bước 1:..... | 5 |
| Bước 2 : Tạo kiểu dữ liệu và danh sách liên kết | 6 |
| Bước 3 : Chức năng nhập dữ liệu | 7 |
| Bước 4 : Chức năng xóa dữ liệu..... | 11 |
| Bước 5 : Chức năng sửa dữ liệu..... | 15 |
| Bước 6 : Truy vấn dữ liệu..... | 16 |
| Bước 7 : Vào ra file bin..... | 19 |
| Bước 8 : Chương trình chính..... | 22 |
| II . Giao diện chương trình..... | 24 |
| III. Mã nguồn..... | 36 |
| Phần kết luận..... | 81 |

PHẦN MỞ ĐẦU

Trong khuôn khổ của báo cáo bài tập lớn này, nhóm sinh viên chúng em đã vận dụng những kiến thức được học trong môn Kỹ thuật lập trình như kỹ thuật phát triển chương trình, các cấu trúc dữ liệu cơ bản và các thao tác xử lý file,... để giải quyết bài tập quản lý danh sách sinh viên. Cấu trúc bài báo cáo gồm 3 phần:

Phần 1: Mở đầu

Phần 2: Nội dung

- +Phát triển chương trình

- +Mã nguồn

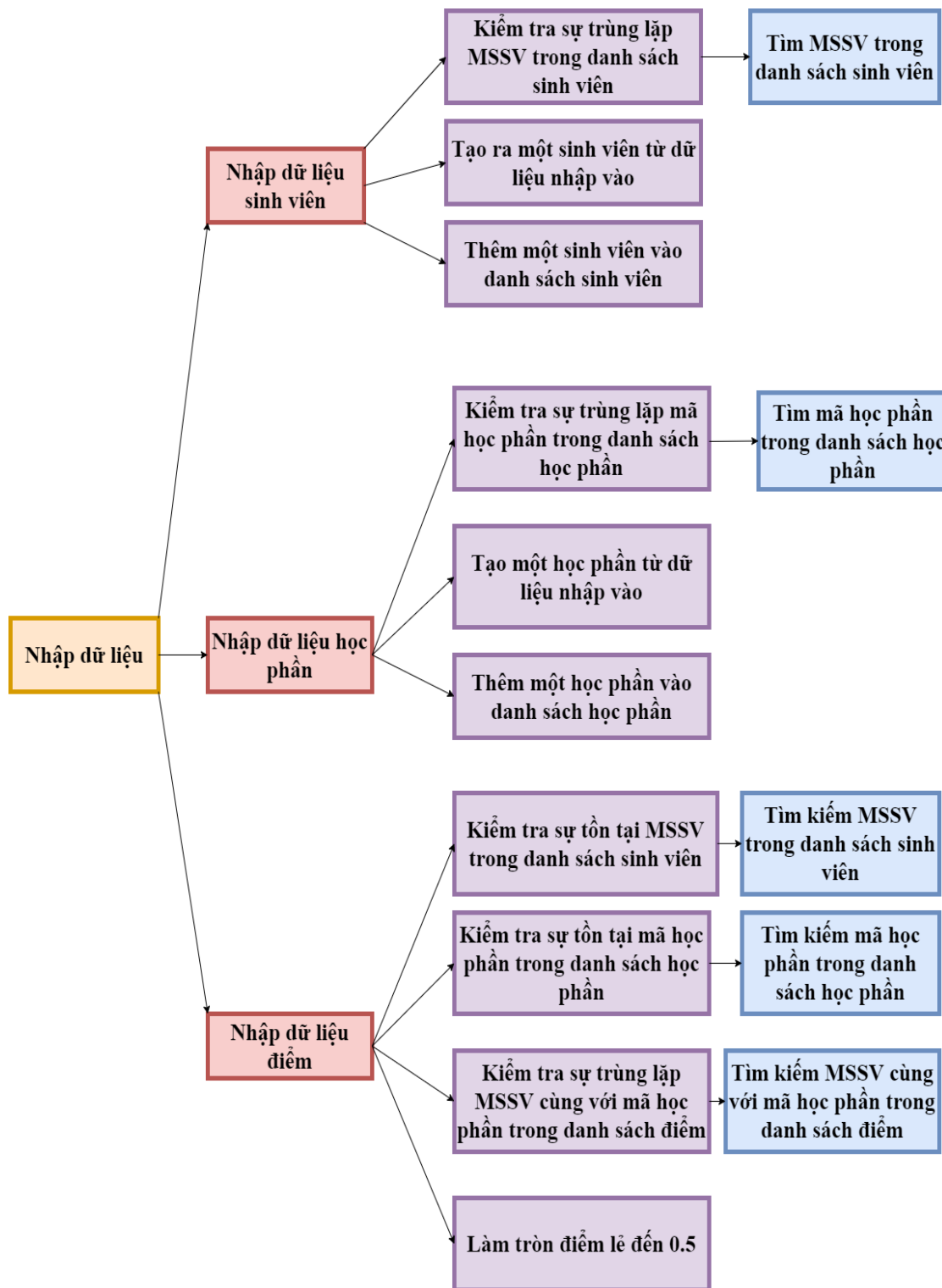
- +Giao diện chương trình

Phần 3: Kết luận

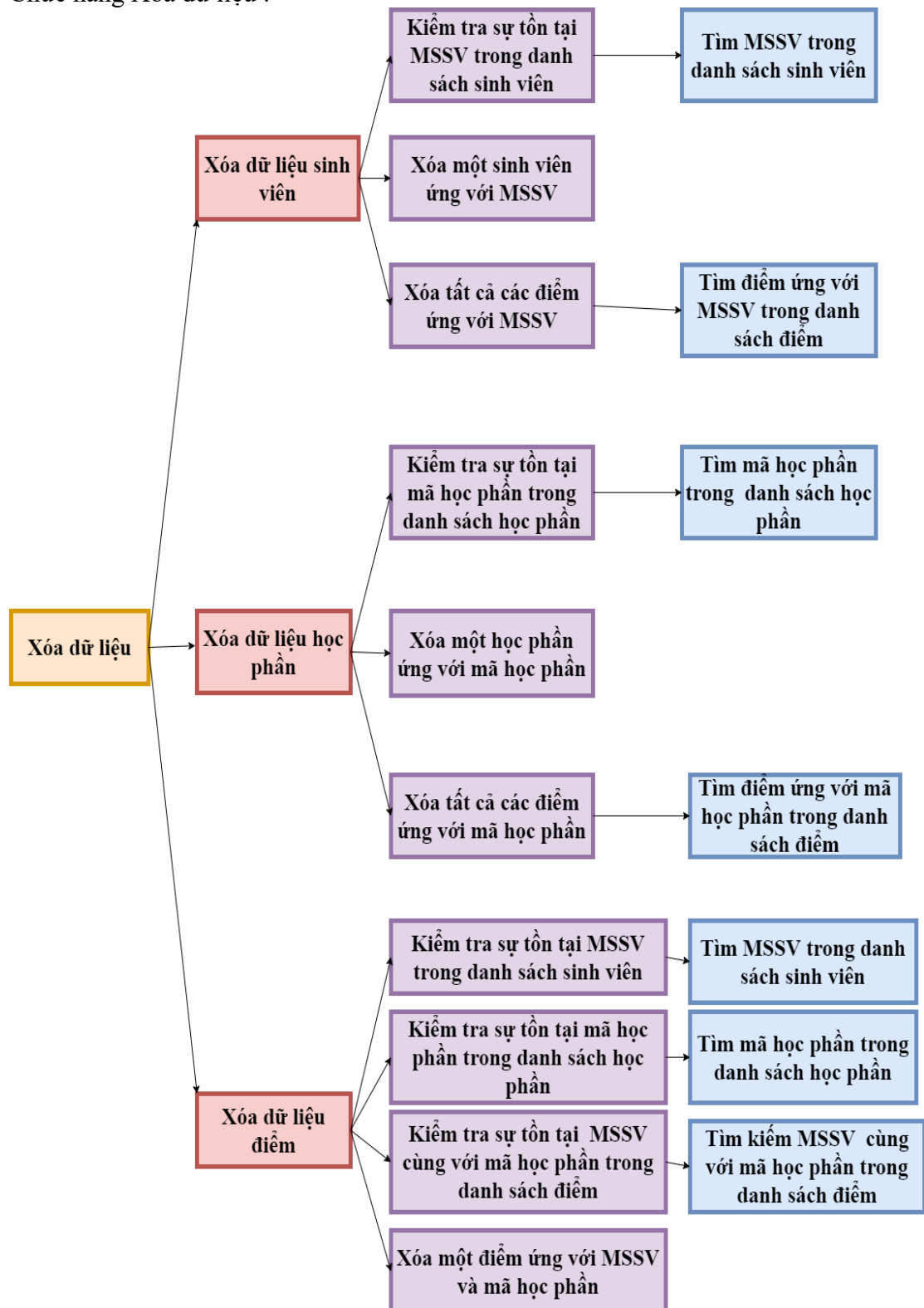
Phần nội dung : I. Phát triển chương trình

Bước 0 :

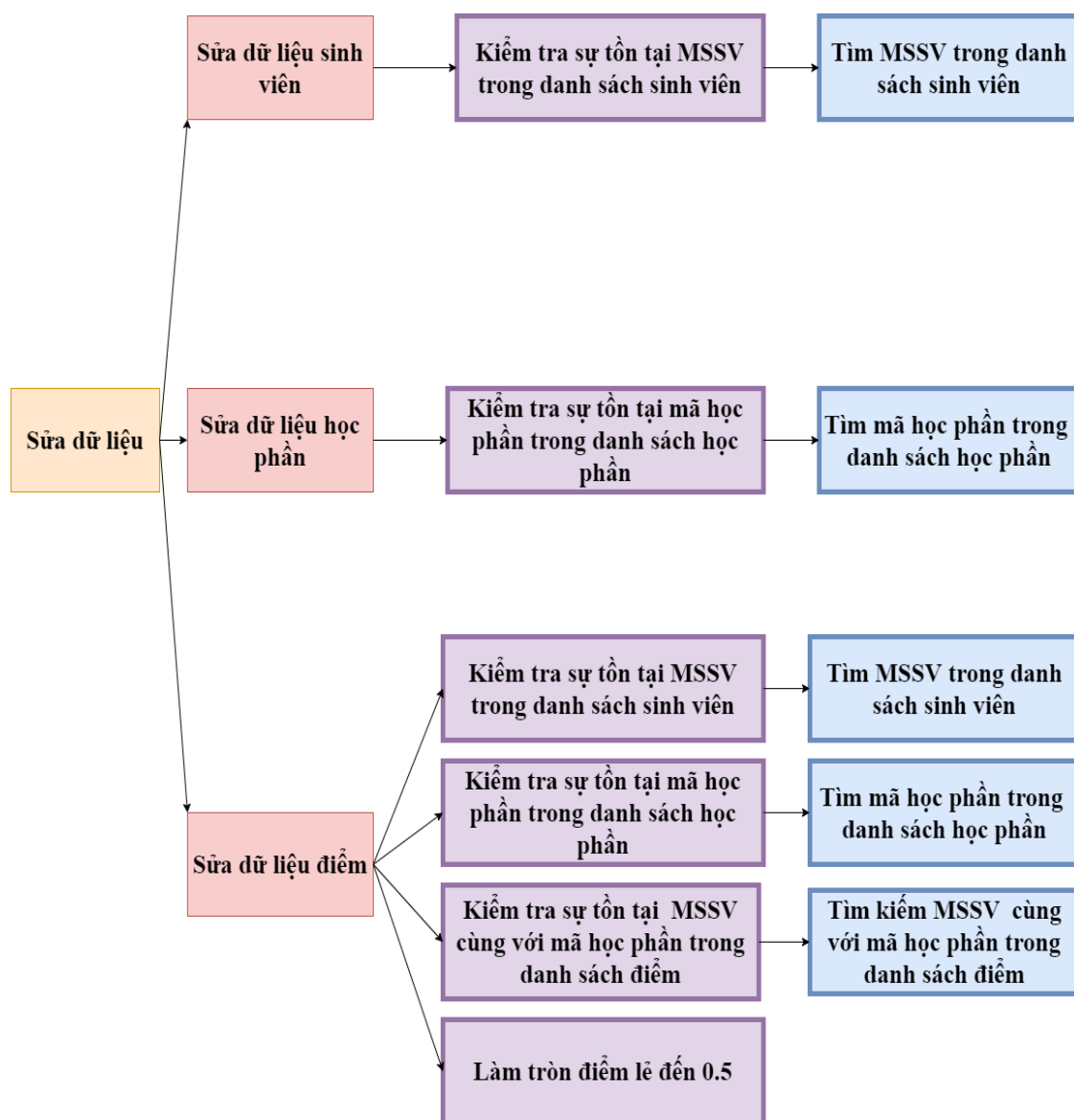
Chức năng Nhập dữ liệu :



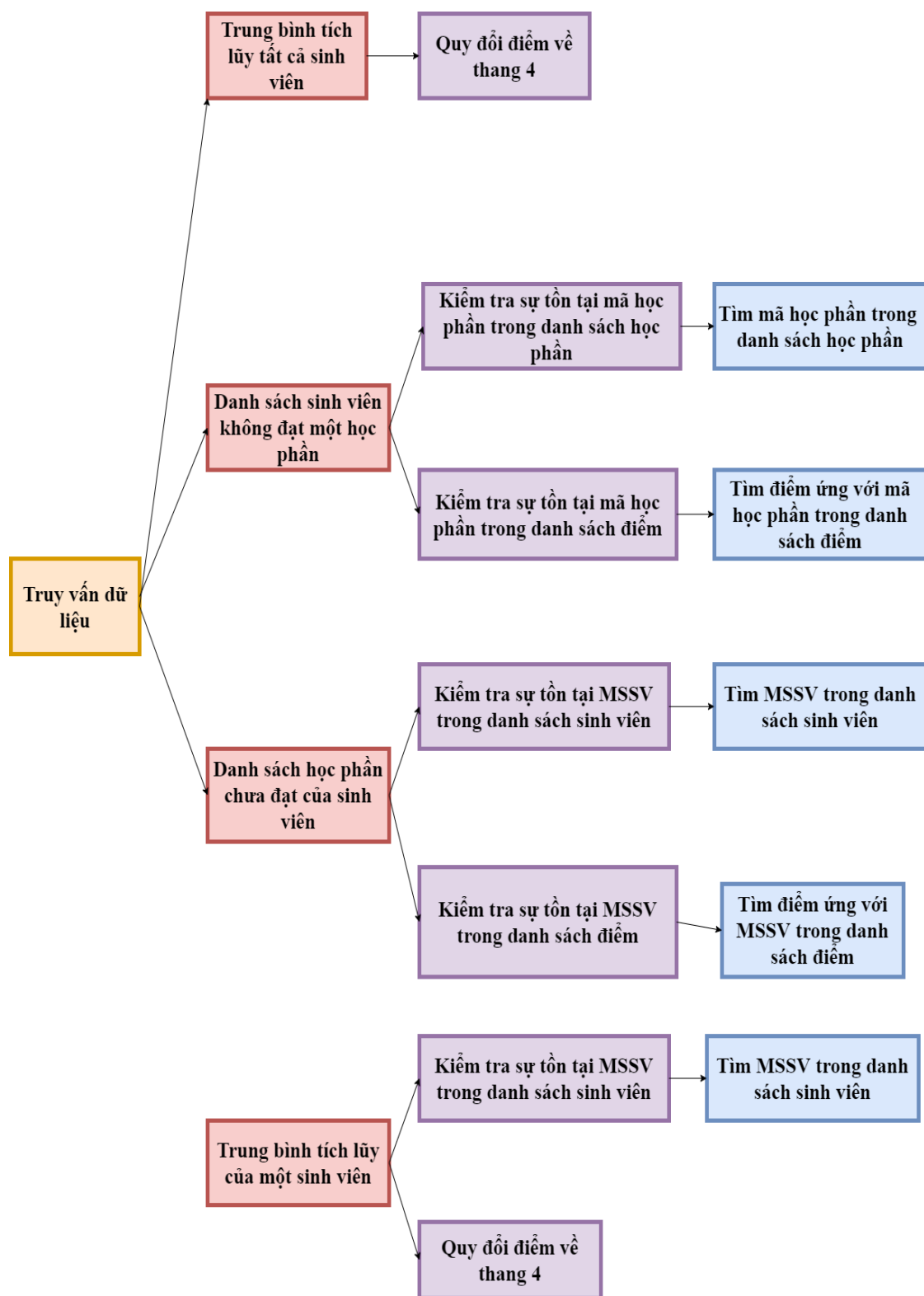
Chức năng Xóa dữ liệu :



Chức năng Sửa dữ liệu :



Chức năng Truy vấn dữ liệu :



Bước 1 :

Dữ liệu vào:

- Thông tin sinh viên (Lưu file **SINHVIEN.bin**)
 - + Mã số sinh viên.
 - + Họ tên.
 - + Giới tính.
 - + Ngày sinh (ngày sinh, tháng sinh, năm sinh).
- Thông tin học phần (Lưu file **HOCPHAN.bin**)
 - + Mã học phần.
 - + Tên học phần
 - + Số tín chỉ.
- Thông tin điểm (Lưu file **DIEM.bin**)
 - + Mã số sinh viên.
 - + Mã học phần.
 - + Điểm.
- Nhập thông tin để thêm, xóa, sửa và truy xuất dữ liệu

* **Điều kiện:** Thêm xóa sửa sao cho không bị lặp:

- + Không có 2 sinh viên cùng "MSSV" trong **SINHVIEN.bin**.
- + Không có 2 môn học cùng "Mã học phần" trong **HOCPHAN.bin**.
- + Không có hai bản ghi trong **DIEM.bin** có đồng thời "MSSV" và "Mã học phần" giống nhau.
- + Ngày sinh được nhập theo đúng định dạng ngày tháng.
- + Điểm thi học phần là một số thực làm tròn lẻ đến 0.5.
- + Cập nhật file **DIEM.bin** sau file **SINHVIEN.bin** và **HOCPHAN.bin**:
 - Mọi mã học phần trong **DIEM.bin** có trong file **HOCPHAN.bin**.
 - Mọi mã sinh viên trong **DIEM.bin** có trong file **SINHVIEN.bin**.

Dữ liệu ra:

- Hiện thị thông tin sinh viên, học phần, điểm thi ở file **SINHVIEN.bin**, **HOCPHAN.bin**, **DIEM.bin**
- Tính điểm trung bình tích lũy từng sinh viên và lưu vào file **CPA.bin** (mssv, điểm trung bình tích lũy).
- Hiện thị danh sách gồm "Mã số, Họ tên" Những sinh viên không đạt học phần được nhập vào.
- Hiện thị danh sách các học phần sinh viên chưa đạt khi mssv được nhập vào.
- Hiện thị điểm trung bình tích lũy của sinh viên khi mssv được nhập vào.

Điều kiện:

- + Điểm trung bình tích lũy được tính trên thang điểm 4 và lấy đến 2 chữ số sau dấu phẩy.
- + Điểm < 4 thì không đạt học phần.

Ý TƯỞNG THUẬT TOÁN:

- Khai báo 3 kiểu dữ liệu:
- + **SINHVIEN**(mssv, hoten, gioitinh, ngaysinh).
- + **HOCPHAN**(mahp, tenhp, sotc).
- + **Diem**(mssv, mahp, diem).
- Tạo danh sách liên kết:
- + Tạo danh sách liên kết **LISHsv** để lưu danh sách sinh viên.
- + Tạo danh sách liên kết **LISHhp** để lưu danh sách học phần.
- + Tạo danh sách liên kết **LISHd** để lưu điểm điểm.
- + Mỗi lần vào chương trình thì sẽ đọc hết dữ liệu từ 3 file **SINHVIEN.bin**, **HOCPHAN.bin**, **DIEM.bin** và lưu vào danh sách liên kết.
- + Các quá trình thêm xóa sửa đều được lưu vào danh sách liên kết, khi kết thúc quá trình thực hiện thì sẽ lưu vào các file tương ứng để bảo toàn dữ liệu.
- Quản lý chương trình bằng menu gồm các chức năng: **Hiển thị dữ liệu**, **Nhập dữ liệu**, **Xóa dữ liệu**, **Sửa dữ liệu**, **Truy xuất dữ liệu**, **Out**

Bước 2 : Tạo kiểu dữ liệu và danh sách liên kết

Tạo kiểu dữ liệu **SINHVIEN** và cấu trúc danh sách sinh viên **LISHsv** :

Kiểu dữ liệu **NGAYSINH** :

ngay : ngày sinh
thang : tháng sinh
nam : nam sinh

Kiểu dữ liệu **SINHVIEN** :

mssv : mã số sinh viên
hoten : họ và tên
gioitinh : giới tính
ngaysinh : có kiểu dữ liệu **NGAYSINH** để lưu ngày, tháng, năm sinh

Cấu trúc một nút sinh viên **NODEsv** :

data : dữ liệu kiểu **SINHVIEN**
con trỏ Next : trỏ đến **NODEsv** kế tiếp

Cấu trúc một danh sách sinh viên **LISHsv** :

con trỏ Head : trỏ đến **NODEsv** đầu
con trỏ Tail : trỏ đến **NODEsv** cuối

Tạo kiểu dữ liệu **HOCPHAN** và cấu trúc danh sách học phần **LISHhp** :

Kiểu dữ liệu **HOCPHAN** :

mahp : mã học phần

tenhp : tên học phần

sotc : số tín chỉ

Cấu trúc một nút học phần **NODEhp** :

data : dữ liệu kiểu **HOCPHAN**

con trỏ Next : trỏ đến **NODEhp** kế tiếp

Cấu trúc một danh sách học phần **LISHhp** :

con trỏ Head : trỏ đến **NODEhp** đầu

con trỏ Tail : trỏ đến **NODEhp** cuối

Tạo kiểu dữ liệu **DIEM** và cấu trúc danh sách điểm **LISHD** :

Kiểu dữ liệu **DIEM** :

mssv : mã số sinh viên

mahp : mã học phần

diem : điểm học phần

Cấu trúc một nút điểm **NODEd** :

data : dữ liệu kiểu **DIEM**

con trỏ Next : trỏ đến **NODEd** kế tiếp

Cấu trúc một danh sách điểm **LISHd** :

con trỏ Head : trỏ đến **NODEd** đầu

con trỏ Tail : trỏ đến **NODEd** cuối

Tạo kiểu dữ liệu **CPA**

Kiểu dữ liệu **CPA**

mssv : mã số sinh viên

cpa : điểm trung bình tích lũy

Bước 3 : Chức năng nhập dữ liệu

Hàm nhập dữ liệu Sinh viên vào danh sách sinh viên :

SINHVIEN InputSV (LISHsv lsv) :

- khai báo SINHVIEN info
- nhập vào mssv
- tìm kiếm mssv trong danh sách sinh viên : **TimSV (lsv,mssv)**
- vòng lặp while (tìm thấy một NODEsv ứng với mssv) :
 nhập lại mssv
 TimSV(lsv,mssv)
- nhập họ tên
- nhập giới tính
- nhập ngày, tháng, năm sinh
- trả về info

Hàm tìm sinh viên trong danh sách sinh viên :

NODEsv *TimSV (lsv, mssv) :

- vòng lặp for duyệt qua tất cả các NODEsv có trong danh sách sinh viên :
 nếu có 1 NODEsv ứng với mssv : trả về về NODEsv đó
- nếu không tìm được NODEsv nào ứng với mssv: trả về NULL

Hàm tạo ra một NODEsv :

NODEsv taoNODEsv (SINHVIEN info)

- khai báo biến con trỏ : NODEsv *p
- cấp phát vùng nhớ cho p
- p → data = info
- p → Next = NULL
- trả về p

Thủ tục thêm một NODEsv vào đầu danh sách sinh viên :

void themNODEsvvaodau (LISHsv &lsv , NODEsv *p)

- nếu (lsv.Head == NULL)
 lsv.Head = lsv.Tail = p
- ngược lại
 p → Next = lsv.Head
 lsv.Head = p

Hàm nhập dữ liệu Học phần vào danh sách học phần :

HOCPHAN InputHP (LISHhp lhp) :

- khai báo HOCPHAN info
- nhập vào mahp
- tìm kiếm mahp trong danh sách học phần : **TimHP (lhp,mahp)**
- vòng lặp while (tìm thấy một NODEhp ứng với mahp) :
nhập lại mahp
TimHP(lhp,mahp)
- nhập tên học phần
- nhập số tín chỉ
- trả về info

Hàm tìm học phần trong danh sách học phần :

NODEhp *TimHP (lhp, mahp) :

- vòng lặp for duyệt qua tất cả các NODEhp có trong danh sách học phần :
nếu có 1 NODEhp ứng với mahp : trả về về NODEhp đó
- nếu không tìm được NODEhp nào ứng với mahp: trả về NULL

Hàm tạo ra một NODEhp :

NODEhp taoNODEhp (HOCPHAN info)

- khai báo biến con trỏ : NODEhp *p
- cấp phát vùng nhớ cho p
- p → data = info
- p → Next = NULL
- trả về p

Thủ tục thêm một NODEhp vào đầu danh sách học phần :

void themNODEhpvaodau (LISHhp &lhp , NODEhp *p)

- nếu (lhp.Head == NULL)
lhp.Head = lhp.Tail = p
- ngược lại
p → Next = lhp.Head
lhp.Head = p

Hàm nhập dữ liệu điểm trong danh sách điểm

DIEM InputD (LISHsv lsv, LISHhp lhp , LISHd ld)

```
- khai báo DIEM info
NODEsv *p1
NODEhp *p2
NODEd *p
do {
- nhập info.mssv
- p1 = TimSV (lsv,info.mssv) để kiểm tra sự tồn tại của mssv trong lsv
- vòng lặp while ( p1 == NULL )
    nhập lại mssv
    p1 = TimSV (lsv,info.mssv)
- p2 = TimHP (lhp,info.mahp) để kiểm tra sự tồn tại của mahp trong lhp
- vòng lặp while ( p2 = NULL )
    nhập lại info.mahp
    p2 = TimHP (lhp,info.mahp)
- p = TimD (ld,info.mssv,info.mahp) để kiểm tra sự trùng lặp cặp
(mssv,mahp) trong ld
} while ( p != NULL )
- nhập điểm : info.diemhp
- vòng lặp while ( info.diemhp <0 || info.diemhp >10)
    nhập lại điểm : info.diemhp
- Làm tròn điểm lẻ đến 0.5 : Lamtron (info.diemhp)
- trả về info
```

Hàm tìm mã số sinh viên và mã học phần trong danh sách điểm

NODEd Timd (LISHd ld , int mssv , char mahp[])

```
- vòng lặp for duyệt lần lượt từng NODEd có trong ds điểm , biến con trỏ lặp k
nếu (( k→data.mssv == mssv ) && ( strcmp (k→data.mahp,mahp) == 0))
    trả về k
- nếu không có NODEd nào thỏa mãn thì trả về NULL
```

Hàm làm tròn điểm lẻ đến 0.5

float Lamtron (float a)

```
- khai báo b = a
- while ( a >1 )
    a = a-1
- if ( a ≥ 0 && a < 0. )
    b = b-a
else if(a>=0.25 && a<0.75)
    b=b-a+0.5
else b=b-a+1
- trả về b
```

Bước 4 : Chức năng xóa dữ liệu

Thủ tục xóa dữ liệu sinh viên

```
void XoaNODEsv ( LISHsv &lsv , LISHd &ld )
- khai báo mssv, NODEsv *j , NODEsv *p
- nếu danh sách sinh viên trống thì return
- nhập mssv
- p = TimSV (lsv,mssv )
- nếu ( p == NULL ) thì kết thúc
- nếu ( NODEsv đầu danh sách sinh viên ứng với mssv )
    sử dụng XoaNODEsvdau(lsv)
    return
- nếu ( NODEsv cuối danh sách sinh viên ứng với mssv )
    sử dụng XoaNODEsvcuoi(lsv)
    return
- vòng lặp for duyệt từ đầu đến cuối NODEsv có trong lsv, biến con trỏ lặp k
{
    nếu ( k ứng với mssv ) {
        j→Next = k→Next
        free(k)
        return
    }
    j = k
}
- while ( KiemtramssvtrongLISHd (ld,mssv) )
    sử dụng XoaNODEdkhietmssv (ld,mssv)
```

Thủ tục xóa NODEsv đầu danh sách sinh viên

```
void XoaNODEsvdau ( LISHsv &lsv )
- nếu danh sách trống thì return
- khai báo NODEsv *p
p = lsv.Head
lsv.Head = lsv.Head→Next
p→Next = NULL
free(p)
```

Thủ tục xóa NODEsv cuối danh sách sinh viên

```
void XoaNODEsvcuoi ( LISHsv &lsv )
- nếu danh sách trống thì return
- vòng lặp for duyệt lần lượt từ đầu đến cuối danh sách sinh viên , biến lặp con
trỏ k
{
    nếu ( k→Next == lsv.Tail ) {
        free (lsv.Tail)
        k→Next = NULL
        lsv.Tail = k
    }
}
```

Hàm kiểm tra mã số sinh viên trong danh sách điểm

bool KiemtramssvtrongLISHd (LISHd ld , int mssv)

- dùng for duyệt lần lượt từ đầu đến cuối danh sách điểm, biến lặp con trỏ k
 - {
 - nếu (k ứng với mssv)
 - return true
 - }
- nếu không tìm thấy k thỏa mãn thì return false

Thủ tục xóa NODEd trong danh sách điểm khi biết mssv

void XoaNODEdkhibietmssv (LISHd &ld , int mssv)

- khai báo NODEd *j
- nếu NODEd đầu danh sách điểm ứng với mssv
 sử dụng **XoaNODEddau (ld)**
- nếu NODEd cuối danh sách điểm ứng với mssv
 sử dụng **XoaNODEdcuoi (ld)**
- vòng lặp for duyệt từ đầu đến cuối danh sách điểm, biến lặp con trỏ k {
 - nếu (k ứng với mssv) {
 - j → Next = k → Next
 - free(k)
 - return
 - }
 - j = k
- }

Thủ tục xóa NODEd đầu danh sách điểm

void XoaNODEddau (LISHd &ld)

- nếu danh sách trống thì return
- khai báo NODEd *p
- p = ld.Head
- ld.Head = ld.Head→Next
- p→Next = NULL
- free(p)

Thủ tục xóa NODEd cuối danh sách điểm

```
void XoaNODEdcuoi ( LISHd &ld )
- nếu danh sách trống thì return
- vòng lặp for duyệt lần lượt từ đầu đến cuối danh sách điểm , biến lặp con trỏ k
{
    nếu ( k → Next == ld.Tail ) {
        free (ld.Tail)
        k → Next = NULL
        ld.Tail = k
    }
}
```

Thủ tục xóa dữ liệu học phần

```
void XoaNODEhp ( LISHhp &lhp , LISHd &ld )
- khai báo int x , char mahp[] , NODEsv *j , NODEsv *p
- nếu danh sách học phần trống thì return
- nhập mahp
- p = TimHP (lhp,mahp )
- nếu ( p == NULL ) thì kết thúc
- nếu ( NODEhp đầu danh sách học phần ứng với mahp )
    sử dụng XoaNODEhpdau(lhp)
    return
- nếu ( NODEhp cuối danh sách học phần ứng với mahp )
    sử dụng XoaNODEhpcuoi(lhp)
    return
- vòng lặp for duyệt từ đầu đến cuối NODEhp có trong lhp, biến con trỏ lặp k
{
    x = strcmp ( k→ data.mahp,mahp)
    nếu ( x == 0 ) {
        j→Next = k→Next
        free(k)
        return
    }
    j = k
}
- while ( KiemtramahptrongLISHd (ld,mahp) )
    sử dụng XoaNODEdkhietmahp (ld,mahp)
```

Thủ tục xóa NODEhp đầu danh sách học phần

```
void XoaNODEhpdau ( LISHhp &lhp )
- nếu danh sách trống thì return
- khai báo NODEhp *p
p = lhp.Head
lhp.Head = lhp.Head→Next
p→Next = NULL
free(p)
```


Thủ tục xóa NODEhp cuối danh sách học phần

```
void XoaNODEhpcuoi ( LISHp &lhp )  
- nếu danh sách trống thì return  
- vòng lặp for duyệt từ đầu đến cuối danh sách học phần, biến lặp con trỏ k  
{  
    nếu ( k→Next == lhp.Tail ) {  
        free (lhp.Tail)  
        k→Next = NULL  
        lhp.Tail = k  
    }  
}
```

Hàm kiểm tra mã học phần trong danh sách điểm

```
bool KiemtramahptrongLISHd ( LISHd ld , char mahp[] )  
- dùng for duyệt lần lượt từ đầu đến cuối danh sách điểm, biến lặp con trỏ k  
{  
    nếu ( strcmp ( k →data.mahp,mahp) == 0 )  
        return true  
}  
- nếu không tìm thấy k thỏa mãn thì return false
```

Thủ tục xóa NODEd khi biết mã học phần trong danh sách điểm

```
void XoaNODEdkhobietmahp ( LISHd &ld , char mahp[] )  
- khai báo NODEd *j  
- nếu NODEd đầu danh sách điểm ứng với mahp  
    sử dụng XoaNODEddau (ld)  
- nếu NODEd cuối danh sách điểm ứng với mssv  
    sử dụng XoaNODEdcuoi (ld)  
- vòng lặp for duyệt từ đầu đến cuối danh sách điểm, biến lặp con trỏ k {  
    nếu ( k ứng với mahp ) {  
        j → Next = k → Next  
        free(k)  
        return  
    }  
    j = k  
}
```

Thủ tục xóa dữ liệu điểm

```
void XoaNODEd ( LISHsv &lsv , LISHhp &lhv , LISHd &ld )
- khai báo int mssv , char mahp[] , NODEd *p , NODEd *j
- nếu danh sách điểm trống thì return
- nhập mssv
- nếu ( TimSV ( lsv, mssv ) == NULL )
    return
- nhập mahp
- nếu ( TimHP ( lhv, mahp ) == NULL )
    return
- p = TimD ( ld, mssv, mahp)
- nếu ( p == NULL )
    return
- nếu ( NODEd đầu danh sách điểm ứng với cặp (mssv, mahp) )
    XoaNODEddau (ld)
    return
- nếu ( NODEd cuối danh sách điểm ứng với cặp (mssv,mahp) )
    XoaNODEdcuoi (ld)
    return
- vòng lặp for duyệt từ đầu đến cuối danh sách điểm, biến lặp con trỏ k
    nếu ( k ứng với cặp (mssv,mahp) ) {
        j → Next = k → Next
        free(k)
        return
    }
```

Bước 5 : Chức năng sửa dữ liệu

Thủ tục sửa dữ liệu sinh viên trong danh sách sinh viên

```
void SuaSV ( LISHsv lsv )
- khai báo SINHVIEN suasv
- nhập mssv vào biến suasv.mssv
- if ( TimSV ( lsv,suasv.mssv ) == NULL )
    in ra màn hình : không tồn tại mssv trong danh sách sinh viên
else
    nhập họ tên mới vào suasv.hoten
    nhập giới tính mới vào suasv.gioitinh
    nhập ngày, tháng, năm sinh mới vào suasv.ngaysinh
    gán p → data = suasv
```

Thủ tục sửa dữ liệu học phần trong danh sách học phần

void SuaHP (LISHhp lhp)

- khai báo HOCPHAN suahp
- nhập mahp vào biến suahp.mahp
- if (**TimHP (lhp,suahp.mahp)** == NULL)
 in ra màn hình : không tồn tại mã học phần trong danh sách học phần
- else
 nhập tên môn học mới vào suahp.tenhp
 nhập số tín chỉ mới vào suahp.sotc
 gán p → data = suahp

Thủ tục sửa dữ liệu điểm trong danh sách điểm

void SuaD (LISHsv lsv , LISHhp lhp , LISHd ld)

- khai báo int mssv , char mahp[] , NODEd *p
- nhập mssv
- nếu (**TimSV (lsv, mssv)** == NULL)
 return
- nhập mahp
- nếu (**TimHP (lhp, mahp)** == NULL)
 return
- p = **TimD (ld, mssv, mahp)**
- nếu (p == NULL)
 return
- nhập điểm học phần vào biến p → data.diemhp
- while ((p → data.diemhp < 0) || (p → data.diemhp > 10))
 nhập lại điểm học phần vào biến p → data.diemhp
- gán p → data.diemhp = Lamtron(p → data.diemhp)

Bước 6 : Truy vấn dữ liệu

Thủ tục tính trung bình tích lũy của tất cả sinh viên có trong danh sách sinh viên và ghi vào file CPA.bin

```
void Tinhtrungbinhtichluy ( LISHsv lsv , LISHhp lhp , LISHd ld )
- khai báo NODEhp *k , float S , int Stc , FILE *fcpa , cpa c
- fcpa = fopen ("CPA.bin","wb")
- nếu mở không thành công thì return
- for ( NODEsv *i = lsv.Head ; i != NULL ; i = i → Next )
    gán S = 0
    gán Stc = 0
    for ( NODEd *j = ld.Head ; j != NULL ; j = j → Next ) {
        if ( i → data.mssv == j → data.mssv )
            gán k = TimHP(lhp,j → data.mahp)
            S += (k → data.sotc)*Quydoi diem(j → data.diemhp)
            Stc += k → data.sotc
        }
    if ( Stc == 0 )
        c.mssv = i → data.mssv
        c.cpa = 0
        in ra mssv, cpa từ c
        fwrite (&c, sizeof(cpa),1,fcpa)
    else
        c.mssv = i → data.mssv
        c.cpa = S/Stc
        in ra mssv, cpa từ c
        fwrite (&c, sizeof(cpa),1,fcpa)
- fclose (fcpa)
```

Hàm quy đổi điểm về thang 4

```
float Quydoi diem ( float x )
    nếu (x < 4)
        return 0
    nếu ((x ≥ 4) && (x < 5))
        return 1
    nếu ((x ≥ 5) && (x < 5.5))
        return 1.5
    nếu ((x ≥ 5.5) && (x < 6.5))
        return 2
    nếu ((x ≥ 6.5) && (x < 7))
        return 2.5
    nếu ((x ≥ 7) && (x < 8))
        return 3
    nếu ((x ≥ 8) && (x < 8.5))
        return 3.5
    nếu (x ≥ 8.5)
        return 4
```

Thủ tục truy vấn danh sách sinh viên không đạt một học phần

```
void Danhsachsvchuadathp ( LISHsv lsv , LISHhp lhp , LISHd ld )
- khai báo int count = 0 , NODEsv *p , char mahp[]
- nhập vào mahp
- nếu ( TimHP (lhp,mahp) == NULL )
    return
- nếu ( KiemtramahptrongLISHd(ld,mahp) == false )
    return
- vòng lặp for duyệt từ đầu đến cuối danh sách điểm, biến lặp con trỏ k
    nếu ( strcmp (k→data.mahp,mahp) == 0) && (k→data.diemhp < 4))
        count ++
        p = TimSV (lsv, k→data.mssv)
        in ra mssv,họ tên từ p
- nếu ( count == 0 )
    in ra : danh sách trống !!!
```

Thủ tục truy vấn danh sách học phần chưa đạt của một sinh viên

```
void Danhsachhpchuadatcuasv ( LISHsv lsv , LISHhp lhp , LISHd ld )
- khai báo int count = 0 , int mssv , NODEhp *p
- nhập mssv
- nếu ( TimSV(lsv,mssv) == NULL )
    return
- nếu ( KiemtramssvtrongLISHd(ld,mssv) == false )
    return
- for ( NODEd *k = ld.Head ; k != NULL ; k = k→Next )
    if ((k→data.mssv == mssv) && (k→data.diemhp < 4)) {
        count ++
        p = TimHP (lhp,k→data.mahp)
        in ra mã học phần, tên học phần từ p
    }
- nếu ( count == 0 )
    in ra : danh sách trống !
```

Thủ tục tính trung bình tích lũy của một sinh viên

```
void Trungbinhtichluysv ( LISHsv lsv , LISHhp lhp , LISHd ld )
- khai báo NODEhp *p , NODEsv *j , float S = 0 , int Stc = 0 , int mssv
- nhập mssv
- nếu ( TimSV(lsv,mssv) == NULL )
    return
- vòng lặp for duyệt từ đầu đến cuối danh sách điểm, biến lặp con trỏ k
    nếu ( k ứng với mssv )
        p = TimHP(lhp,k→data.mahp)
        S += (p→data.sotc)*Quydoidiem(k→data.diemhp)
        Stc += p→data.sotc
- if ( Stc == 0 )
    in ra : điểm trung bình tích lũy của sinh viên = Stc
else
    in ra : điểm trung bình tích lũy của sinh viên = S/Stc
```

Bước 7 : Vào ra file bin

Thủ tục đọc danh sách sinh viên từ file bin

```
void DocLISHsvtufilebin(LISHsv &lsv)
{
    FILE *fsv;
    SINHVIEN info;
    fsv = fopen ("SINHVIEN.bin","rb");
    if( fsv == NULL )
    {
        printf("Error!");
        exit(1);
    }
    while ( fread (&info,sizeof(SINHVIEN),1,fsv) > 0 )
    {
        themNODEsvvaodau(lsv,taoNODEsv(info));
    }
    fclose(fsv);
}
```

Thủ tục đọc danh sách học phần từ file bin

```
void DocLISHptufilebin ( LISHhp &lhp )
{
    FILE *fhp;
    HOCPHAN info;
    fhp = fopen ("HOCPHAN.bin","rb");
    if( fhp == NULL )
    {
        printf("Error!");
        exit(1);
    }
    while ( fread( &info, sizeof (HOCPHAN), 1, fhp) > 0 )
    {
        themNODEhpvaodau(lhp,taoNODEhp(info));
    }
    fclose(fhp);
}
```

Thủ tục đọc danh sách điểm từ file bin

```
void DocLISHdtufilebin( LISHd &ld )
{
    FILE *fd;
    DIEM info;
    fd = fopen ("DIEM.bin","rb");
    if( fd == NULL )
    {
        printf("Error!");
        exit(1);
    }
    while ( fread( &info, sizeof(DIEM), 1, fd) > 0 )
    {
        themNODEdvaodau(ld,taoNODEd(info));
    }
    fclose(fd);
}
```

Thủ tục ghi danh sách sinh viên vào file bin

```
void GhiLISHsvvaofilebin ( LISHsv lsv )
{
    FILE *fsv;
    fsv = fopen ("SINHVIEN.bin","wb");
    if( fsv == NULL )
    {
        printf("Error!");
        exit(1);
    }
    for ( NODEsv *k = lsv.Head ; k !=NULL ; k = k->Next )
    {
        fwrite (&k->data,sizeof(SINHVIEN),1,fsv);
    }
    fclose(fsv);
}
```

Thủ tục ghi danh sách học phần vào file bin

```
void GhiLISHhpvaofilebin ( LISHhp lhp )
{
    FILE *fhp;
    fhp = fopen ("HOCPHAN.bin","wb");
    if ( fhp == NULL )
    {
        printf("Error!");
        exit(1);
    }
    for ( NODEhp *k = lhp.Head ; k != NULL ; k = k->Next )
    {
        fwrite ( &k -> data , sizeof(HOCPHAN) , 1 , fhp );
    }
    fclose(fhp);
}
```


Thủ tục ghi danh sách điểm vào file bin

```
void GhiLISHdvaofilebin ( LISHd ld )
{
    FILE *fd;
    fd = fopen ("DIEM.bin","wb");
    if( fd == NULL )
    {
        printf("Error!");
        exit(1);
    }
    for ( NODEd *k = ld.Head ; k !=NULL ; k = k->Next )
    {
        fwrite (&k->data,sizeof(DIEM),1,fd);
    }
    fclose(fd);
}
```

Ý tưởng :

- Lúc mới vào chương trình con trỏ file sẽ đọc hết dữ liệu từ file **SINHVIEN.bin**, **HOCPHAN.bin**, **DIEM.bin**, khi kết thúc chương trình tất cả bộ nhớ đệm của danh sách sinh viên, học phần, điểm sẽ được ghi lại bởi lời gọi hàm fopen(“name_file”, wb) vào các file **SINHVIEN.bin**, **HOCPHAN.bin**, **DIEM.bin**
- Mỗi lần thực hiện chức năng trong menu sẽ được thông qua con trỏ f lưu vào file **NHATKY.txt**
- Sau khi truy xuất danh sách điểm trung bình tích lũy của sinh viên sẽ được thông qua con trỏ fcpa lưu vào file **CPA.bin**.

Bước 8 : Chương trình chính

```
int main () {
    taoLISHsv (lsv)
    taoLISHhp (lhp)
    taoLISHd (ld)
    DocLISHsvtufilebin(lsv)
    DocLISHhptufilebin(lhp)
    DocLISHdtufilebin(ld)
    menu chọn chức năng :
    do {
        nhập 1 : Hiển thị dữ liệu
        nhập 1: hiển thị danh sách sinh viên
        OutputSV (lsv)
        nhập 1 : quay tại menu chính hoặc 2 : kết thúc CT
        nhập 2: hiển thị danh sách học phần
        OutputHP (lhp)
        nhập 1 : quay tại menu chính hoặc 2 : kết thúc CT
        nhập 3 : hiển thị danh sách điểm
        OutputD (ld)
        nhập 1 : quay tại menu chính hoặc 2 : kết thúc CT
    } while (1);
}
```

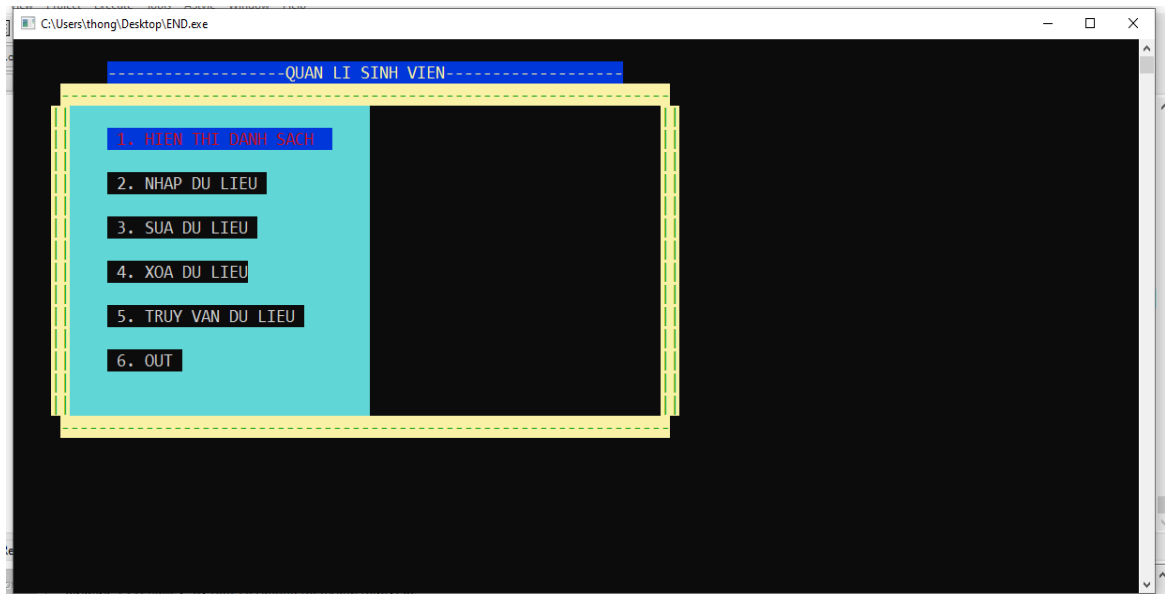
```

nhập 2 : Nhập dữ liệu
    nhập 1: Nhập dữ liệu sinh viên
        InputSV (lsv)
        nhập 1 : quay tại menu chính hoặc 2 : kết thúc CT
    nhập 2: Nhập dữ liệu học phần
        InputHP (lhp)
        nhập 1 : quay tại menu chính hoặc 2 : kết thúc CT
    nhập 3 : Nhập điểm
        InputD (lsv,lhp,ld)
        nhập 1 : quay tại menu chính hoặc 2 : kết thúc CT
nhập 3 : Sửa dữ liệu
    nhập 1: Sửa dữ liệu sinh viên
        SuaSV (lsv)
        nhập 1 : quay tại menu chính hoặc 2 : kết thúc CT
    nhập 2: Sửa dữ liệu học phần
        SửaHP (lhp)
        nhập 1 : quay tại menu chính hoặc 2 : kết thúc CT
    nhập 3 : Sửa điểm
        SuaD (lsv,lhp,ld)
        nhập 1 : quay tại menu chính hoặc 2 : kết thúc CT
nhập 4 : Xóa dữ liệu
    nhập 1 : Xóa dữ liệu sinh viên
        XoaNODEsv (lsv,ld)
        nhập 1 : quay lại menu chính hoặc 2 : kết thúc CT
    nhập 2 : Xóa dữ liệu học phần
        XoaNODEhp (lhp,ld)
        nhập 1 : quay lại menu chính hoặc 2 : kết thúc CT
    nhập 3 : Xóa điểm
        XoaNODEd (lsv,lhp,ld)
        nhập 1 : quay lại menu chính hoặc 2 : kết thúc CT
nhập 5 : Truy vấn dữ liệu
    nhập 1 : Trung bình tích lũy của tất cả sinh viên
        Tinhtrungbinhtichluy (lsv,lhp,ld)
        nhập 1 : quay lại menu chính hoặc 2 : kết thúc CT
    nhập 2 : Danh sách sinh viên chưa đạt một học phần
        Danhsachsvchuadathp (lsv,lhp,ld)
        nhập 1 : quay lại menu chính hoặc 2 : kết thúc CT
    nhập 3 : Danh sách học phần chưa đạt của một sinh viên
        Danhsachhpchuadatcuasv (lsv,lhp,ld)
        nhập 1 : quay lại menu chính hoặc 2 : kết thúc CT
    nhập 4 : Trung bình tích lũy của một sinh viên
        Trungbinhtichluysv (lsv,lhp,ld)
        nhập 1 : quay lại menu chính hoặc 2 : kết thúc CT
nhập 6 : Out
    GhiLISHsvvaofilebin (lsv)
    GhiLISHhpvaofilebin (lhp)
    GhiLISHdvaofilebin (ld)
    Ghitrungbinhtichluyvaofilebin (lsv, lhp, ld)
} while ( nhập sai cú pháp == true)

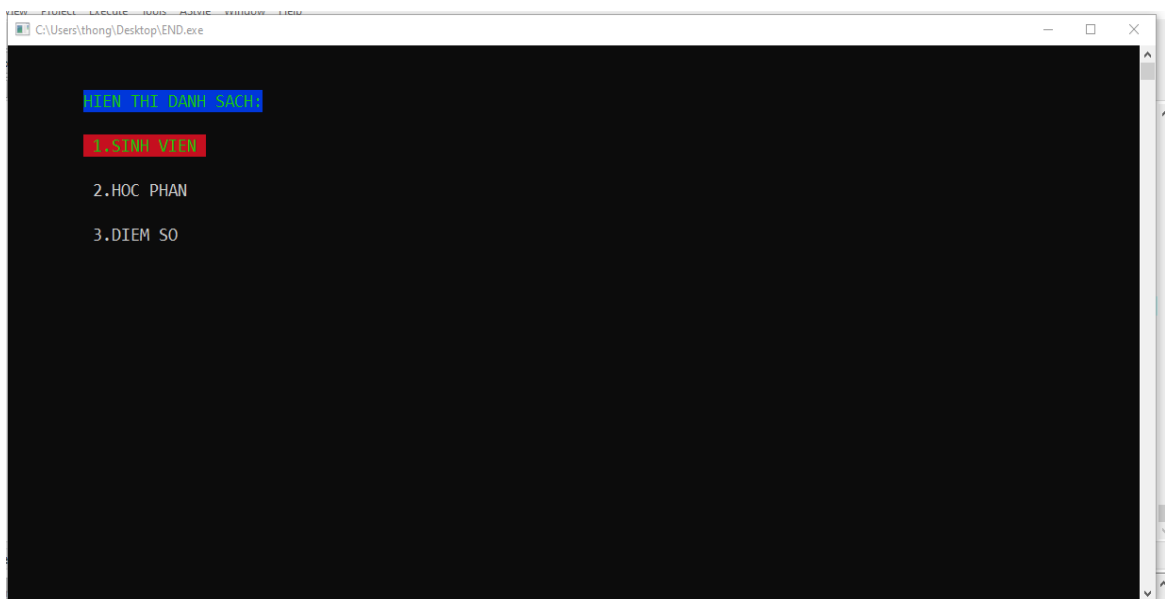
```

II. Giao diện chương trình

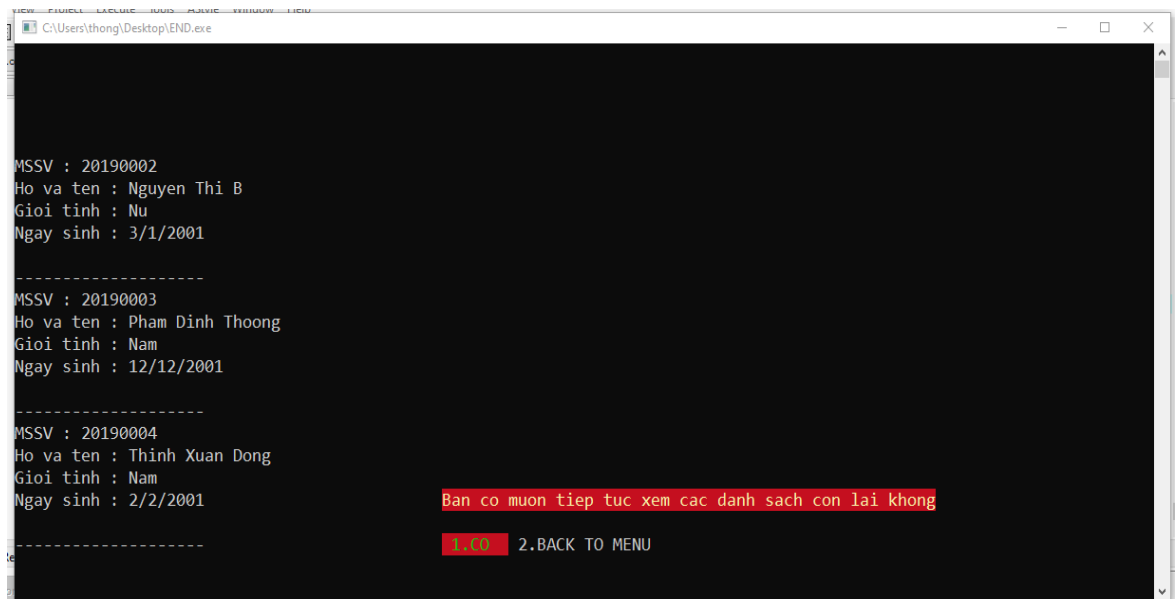
- Menu chính



- Menu con 1



- hiển thị danh sách sinh viên



A screenshot of a Windows command prompt window titled "C:\Users\thong\Desktop\END.exe". The window displays a menu for viewing student records. It lists three students with their IDs, names, genders, and birth dates, separated by dashed lines. At the bottom, there is a red prompt message and two options: "1.CO" and "2.BACK TO MENU".

```
C:\Users\thong\Desktop\END.exe

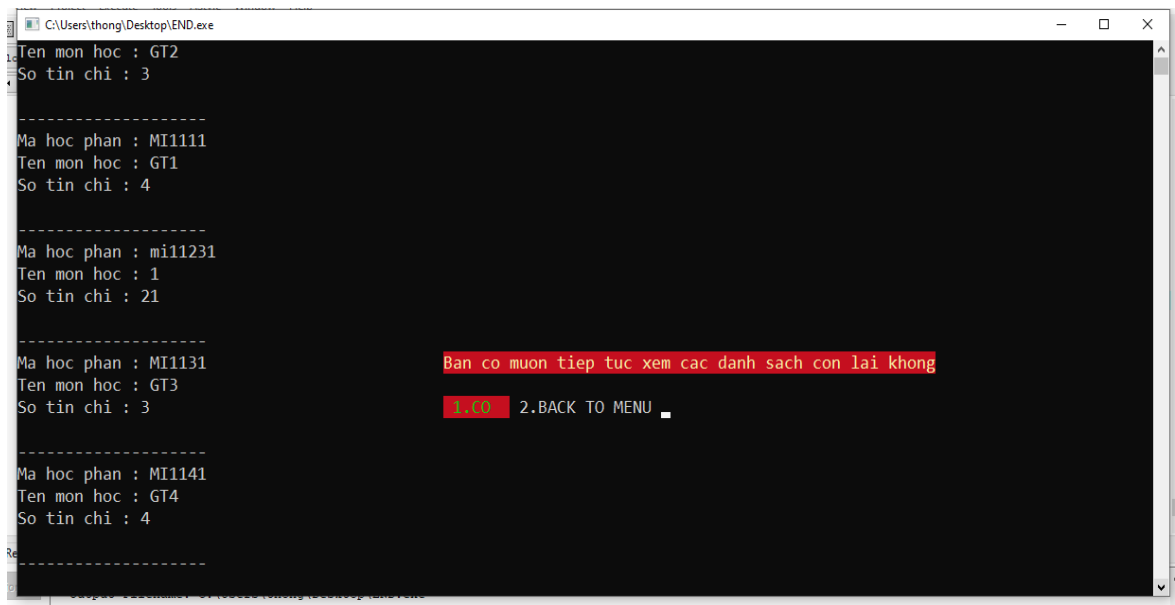
MSSV : 20190002
Ho va ten : Nguyen Thi B
Gioi tinh : Nu
Ngày sinh : 3/1/2001

-----
MSSV : 20190003
Ho va ten : Pham Dinh Thoang
Gioi tinh : Nam
Ngày sinh : 12/12/2001

-----
MSSV : 20190004
Ho va ten : Thinh Xuan Dong
Gioi tinh : Nam
Ngày sinh : 2/2/2001

-----
Ban co muon tiep tục xem các danh sách còn lại không
1.CO 2.BACK TO MENU
```

- hiển thị danh sách học phần



A screenshot of a Windows command prompt window titled "C:\Users\thong\Desktop\END.exe". The window displays a menu for viewing course records. It lists three courses with their IDs, names, and credit values, separated by dashed lines. At the bottom, there is a red prompt message and two options: "1.CO" and "2.BACK TO MENU".

```
C:\Users\thong\Desktop\END.exe

Ten môn học : GT2
Số tín chỉ : 3

-----
Mã học phần : MI1111
Tên môn học : GT1
Số tín chỉ : 4

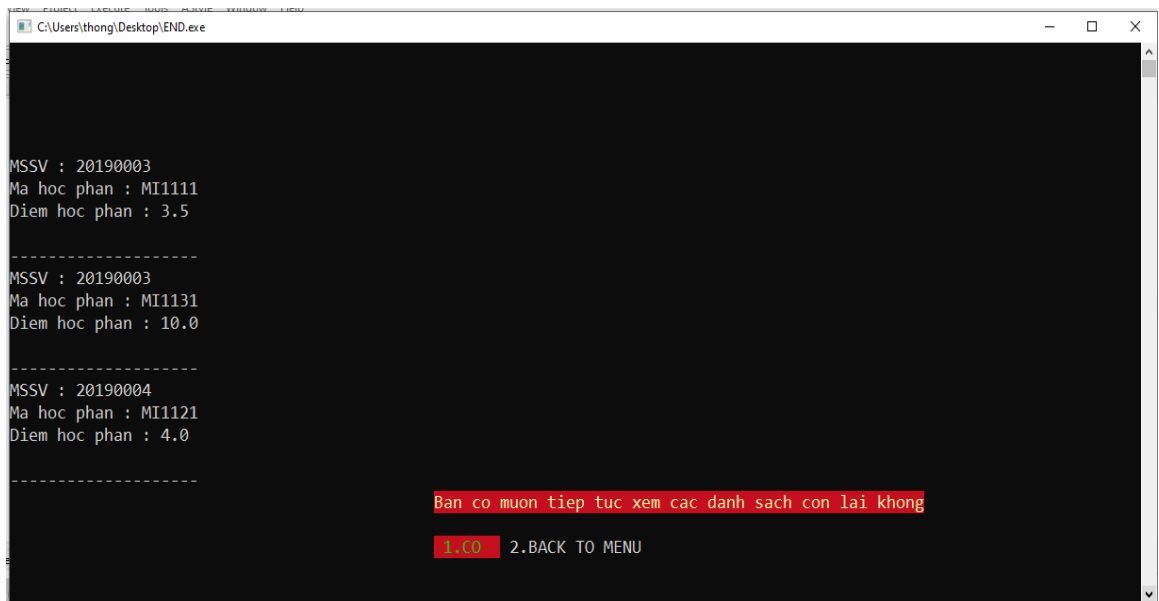
-----
Mã học phần : MI11231
Tên môn học : 1
Số tín chỉ : 21

-----
Mã học phần : MI1131
Tên môn học : GT3
Số tín chỉ : 3

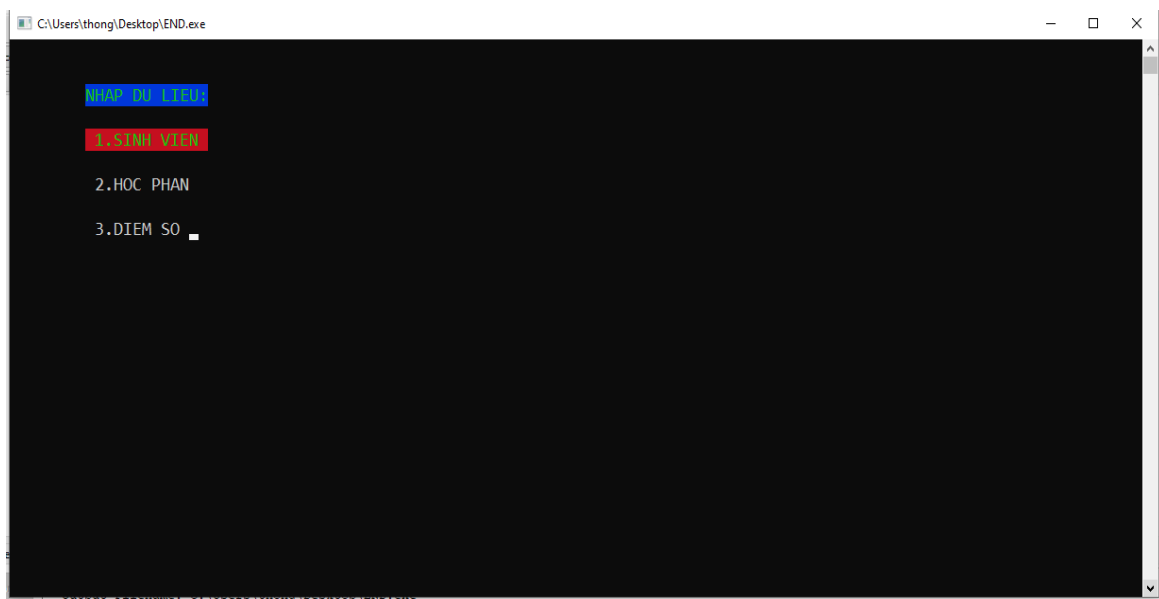
-----
Mã học phần : MI1141
Tên môn học : GT4
Số tín chỉ : 4

-----
Ban co muon tiep tục xem các danh sách còn lại không
1.CO 2.BACK TO MENU
```

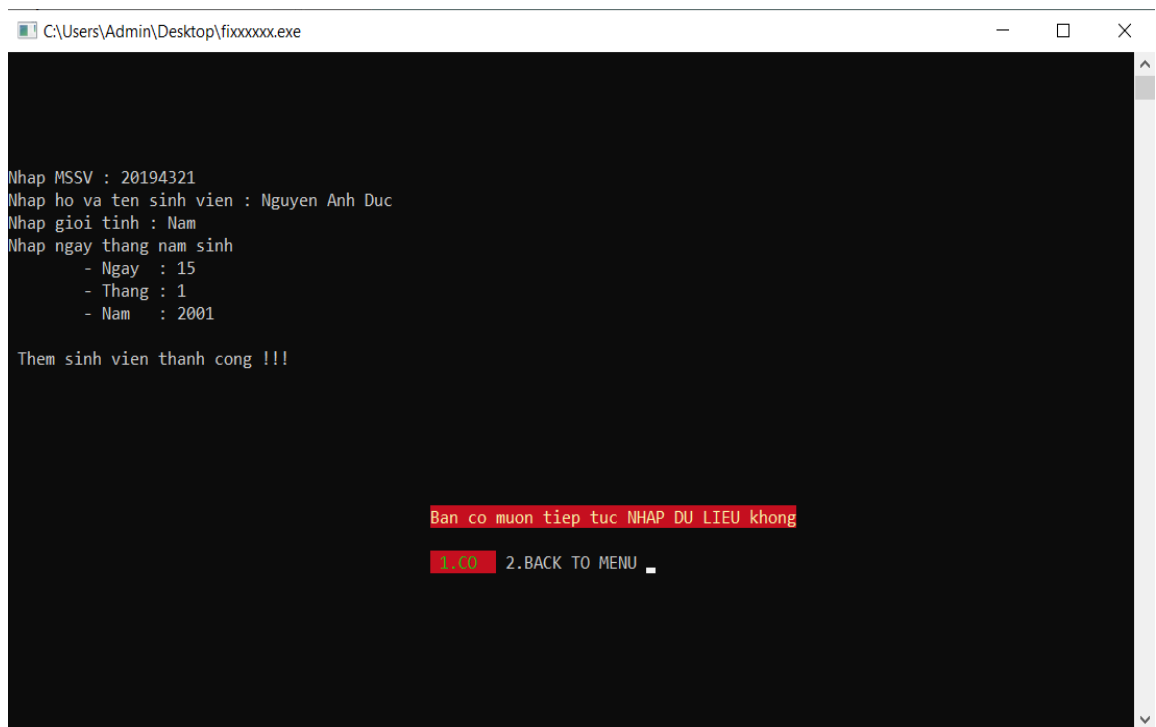
- Hiện thị danh sách điểm



- Menu con 2



- Nhập dữ liệu sinh viên



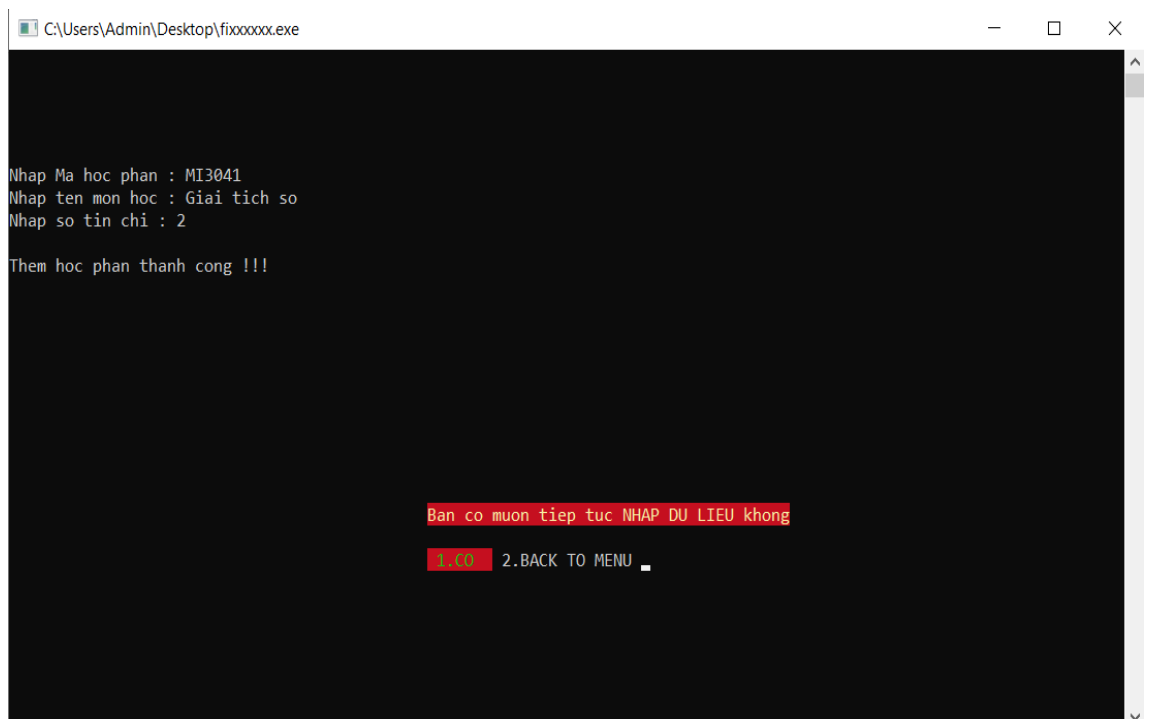
```
C:\Users\Admin\Desktop\fixxxxx.exe

Nhap MSSV : 20194321
Nhap ho va ten sinh vien : Nguyen Anh Duc
Nhap gioi tinh : Nam
Nhap ngay thang nam sinh
  - Ngay : 15
  - Thang : 1
  - Nam : 2001

Them sinh vien thanh cong !!!

Ban co muon tiep tục NHAP DU LIEU không
1.CO 2.BACK TO MENU
```

- Nhập dữ liệu học phần



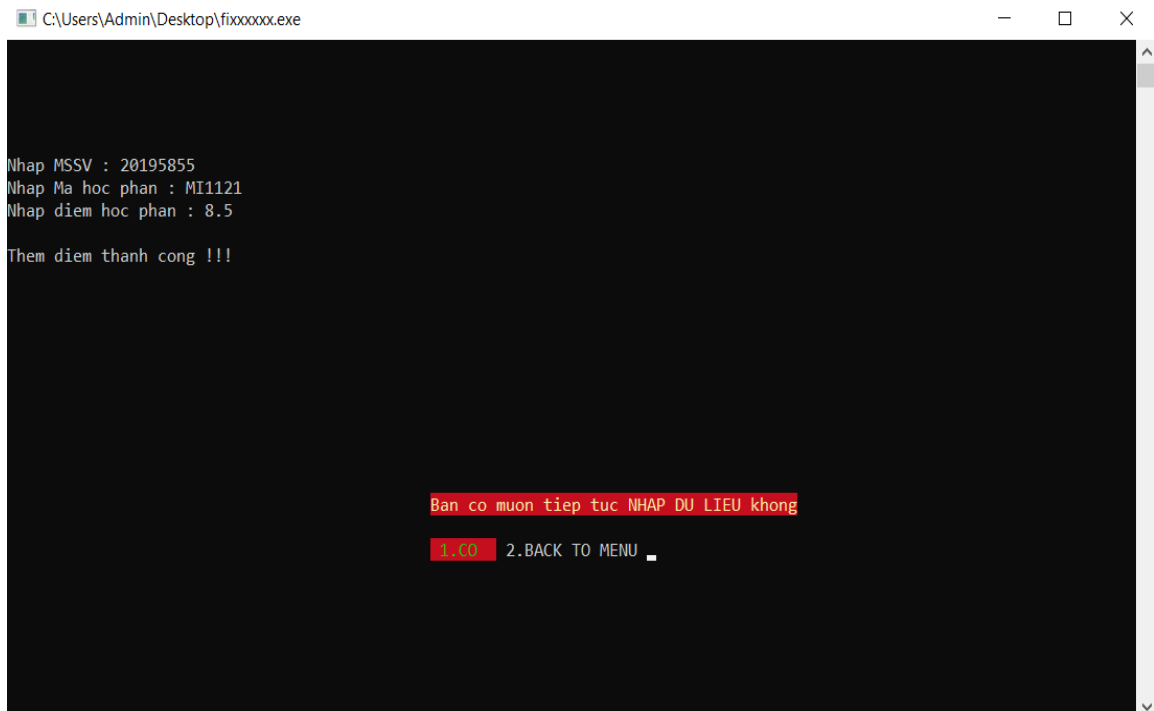
```
C:\Users\Admin\Desktop\fixxxxx.exe

Nhap Ma hoc phan : MI3041
Nhap ten mon hoc : Giai tích so
Nhap so tin chi : 2

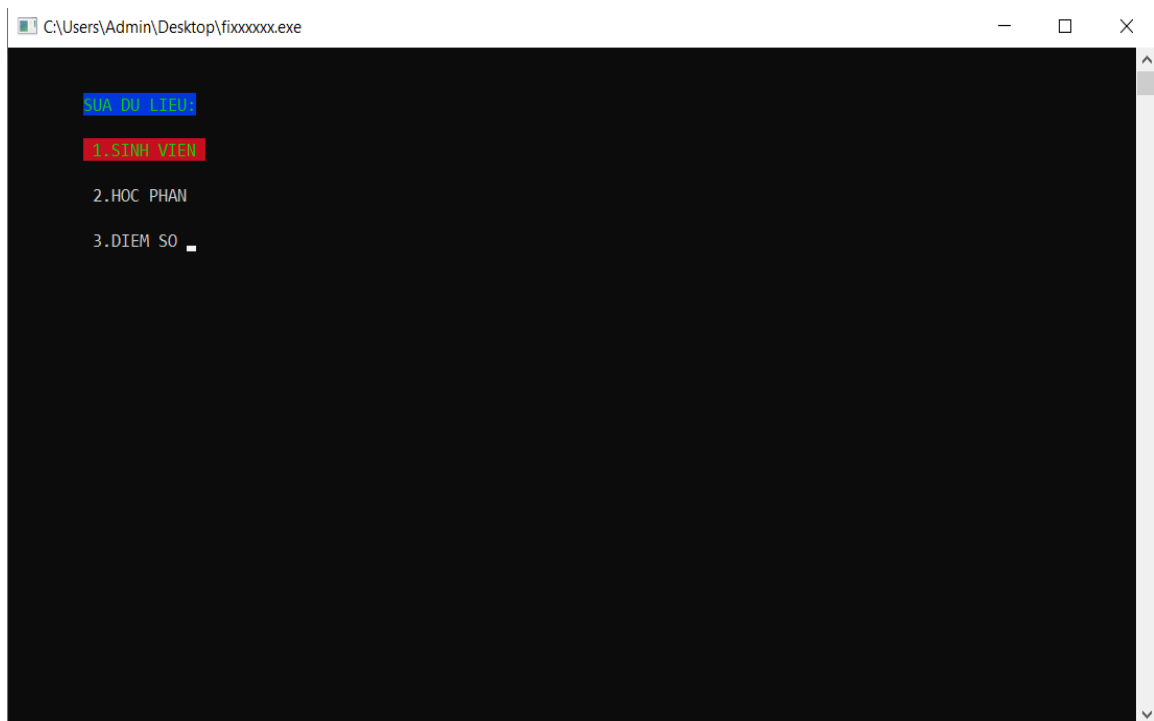
Them hoc phan thanh cong !!!

Ban co muon tiep tục NHAP DU LIEU không
1.CO 2.BACK TO MENU
```

- Nhập dữ liệu điểm



- Menu con 3



- Sửa dữ liệu sinh viên

```
C:\Users\Admin\Desktop\fixxxxxx.exe

Nhap MSSV : 20194321
Nhap ho va ten moi : Ba Nhat
Nhap gioi tinh moi : Nam
Nhap ngay thang nam sinh moi :
    - Ngay : 15
    - Thang : 1
    - Nam : 2001

Sua sinh vien thanh cong !!!

Ban co muon tiep tuc SUA DU LIEU khong
1.CO 2.BACK TO MENU
```

- Sửa dữ liệu học phần

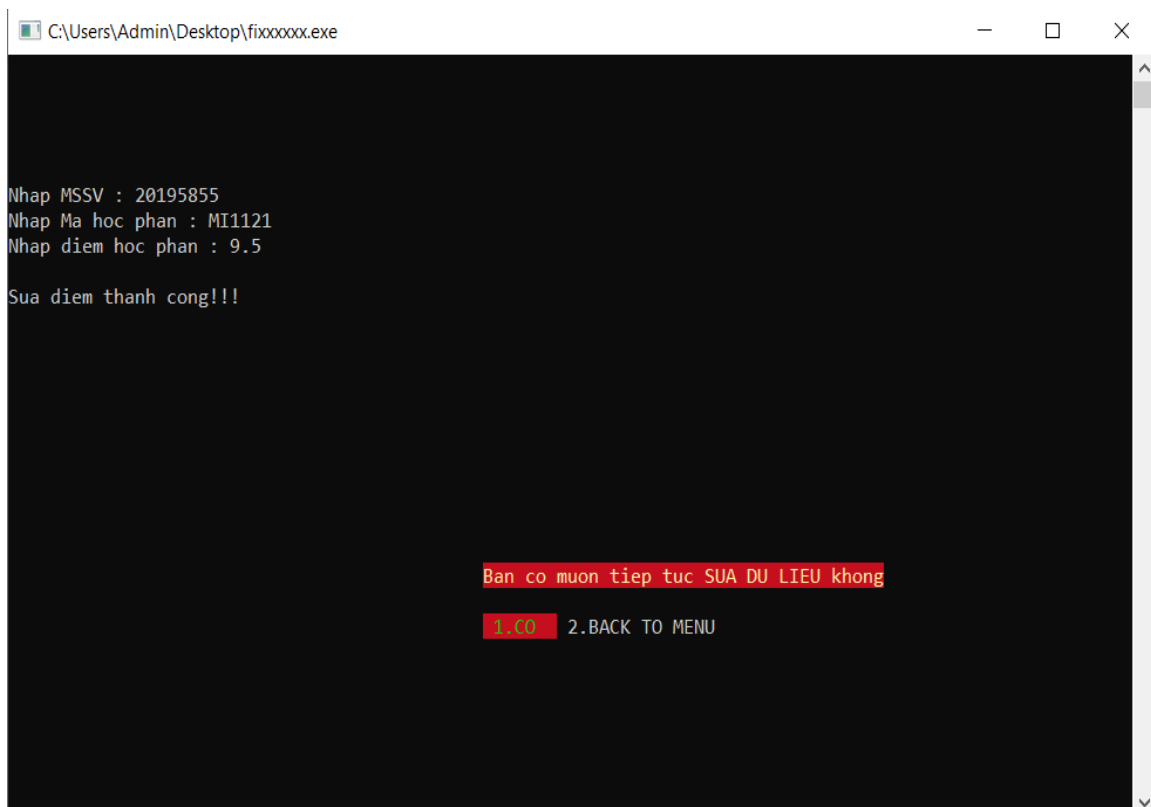
```
C:\Users\Admin\Desktop\fixxxxxx.exe

Nhap Ma hoc phan : MI3041
Nhap ten mon hoc : Giai tinh so
Nhap so tin chi : 2

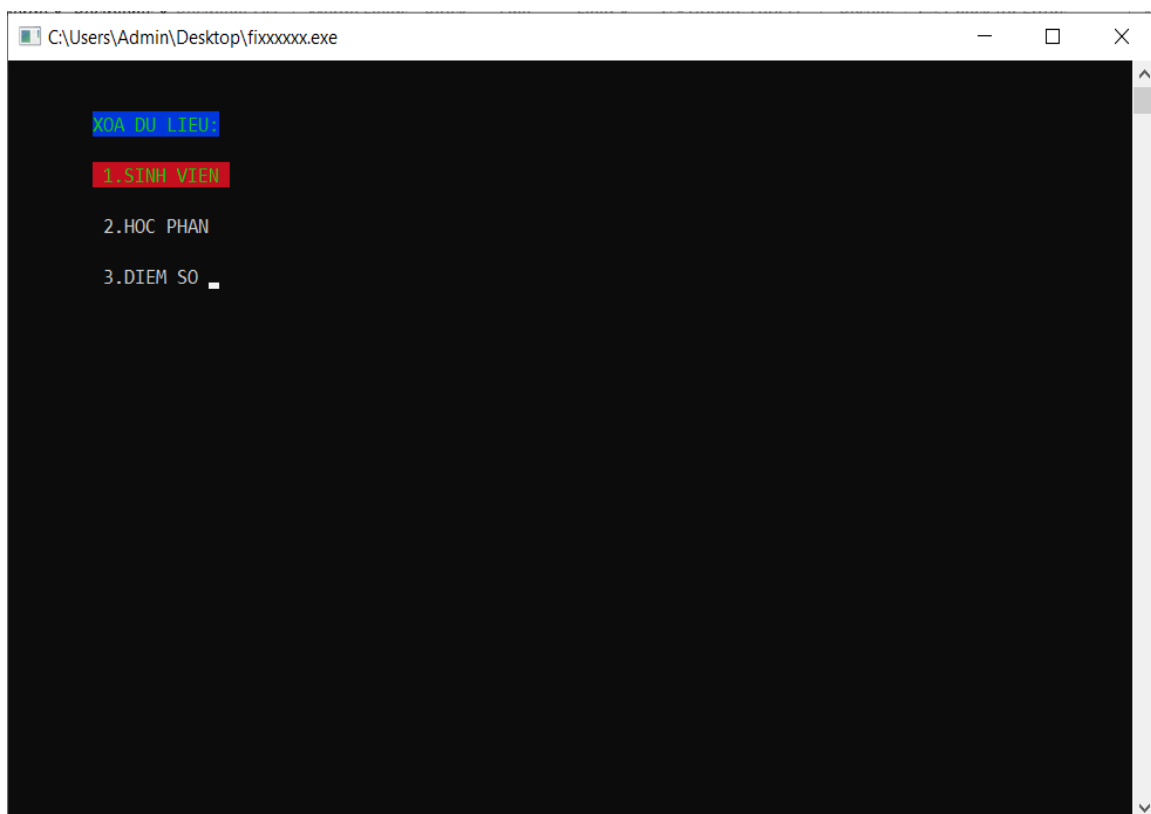
Sua hoc phan thanh cong!!!

Ban co muon tiep tuc SUA DU LIEU khong
1.CO 2.BACK TO MENU
```

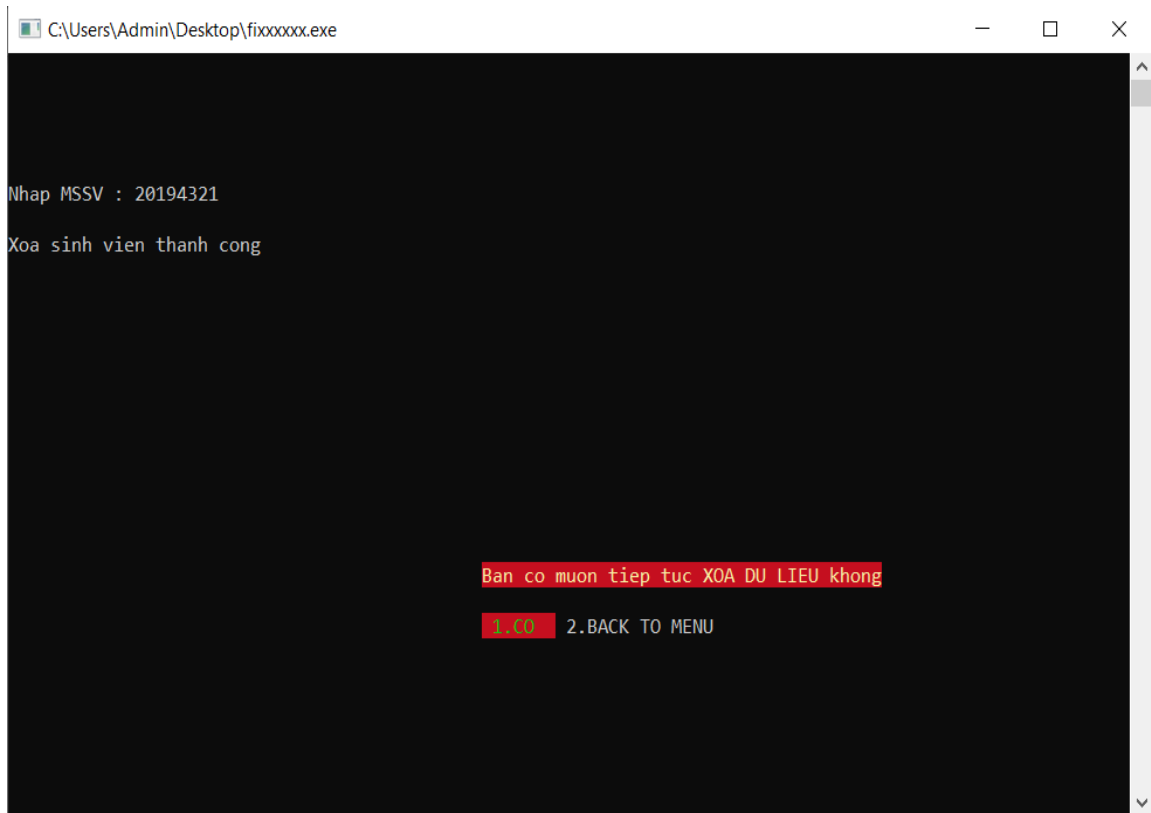

- Sửa dữ liệu điểm



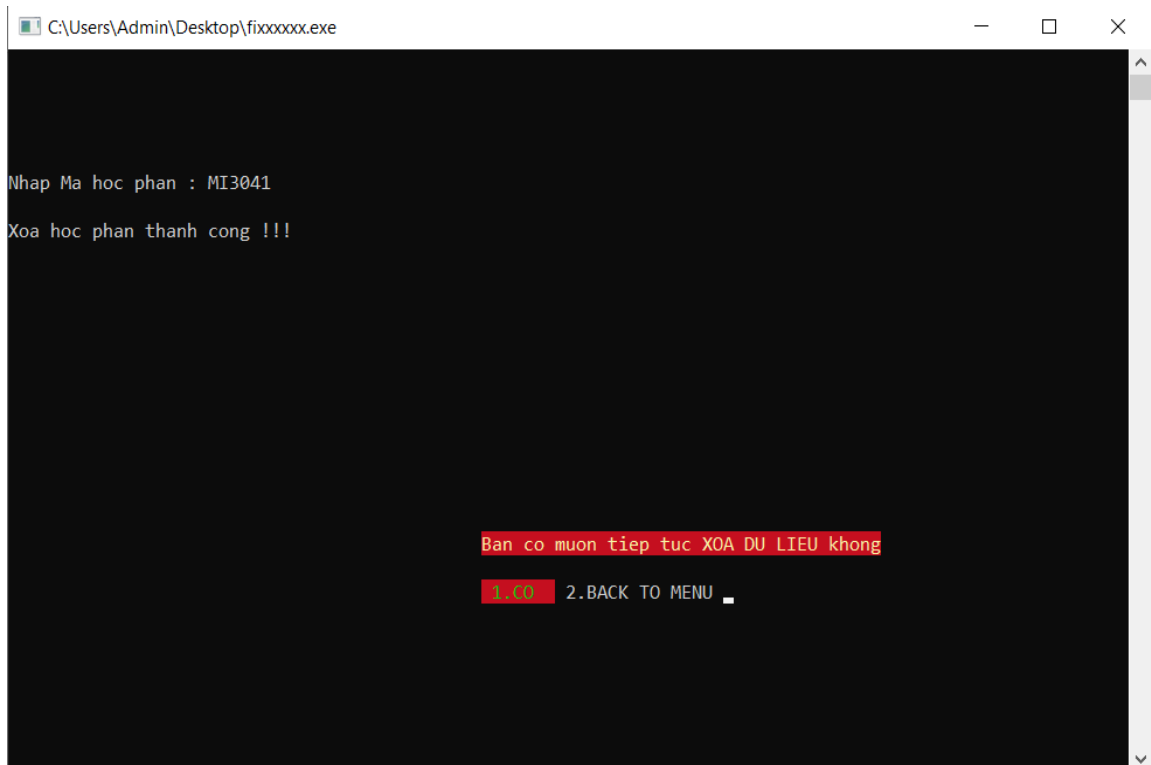
- Menu con 4



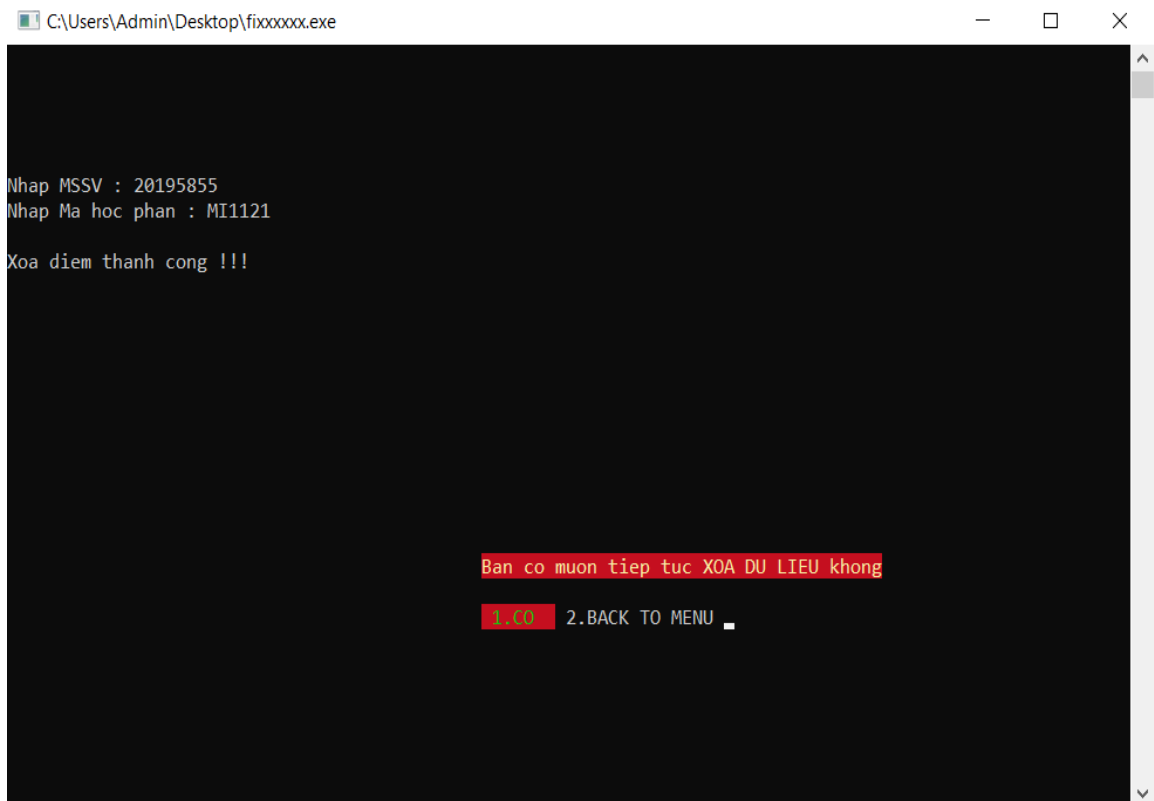
- Xóa dữ liệu sinh viên



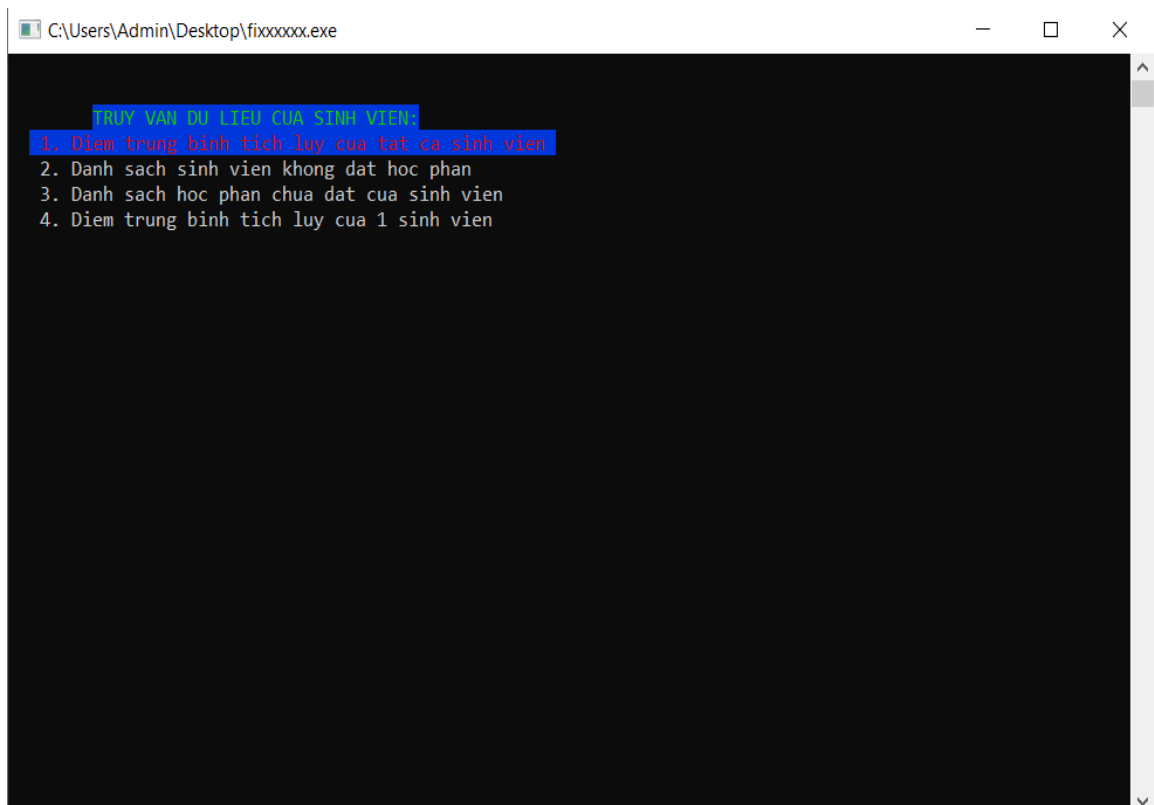
- Xóa dữ liệu học phần



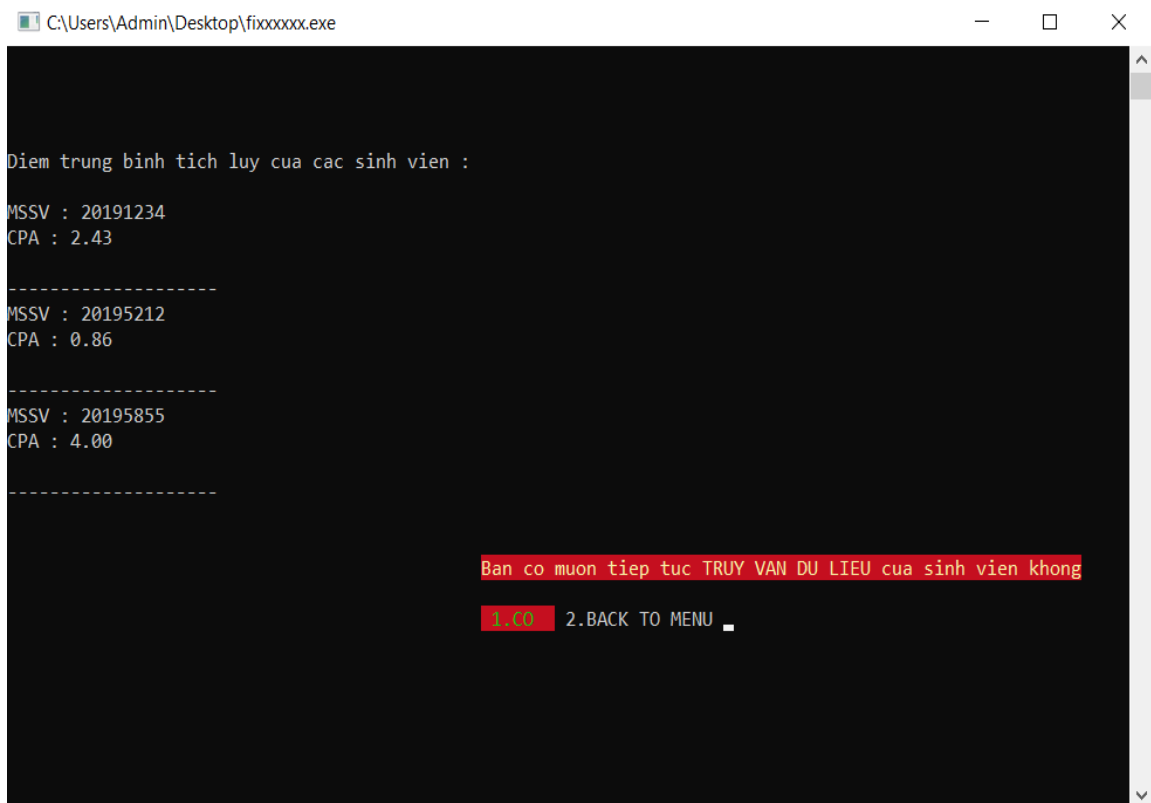
- Xóa dữ liệu điểm



- Menu con 5



- Điểm trung bình tích lũy của tất cả sinh viên



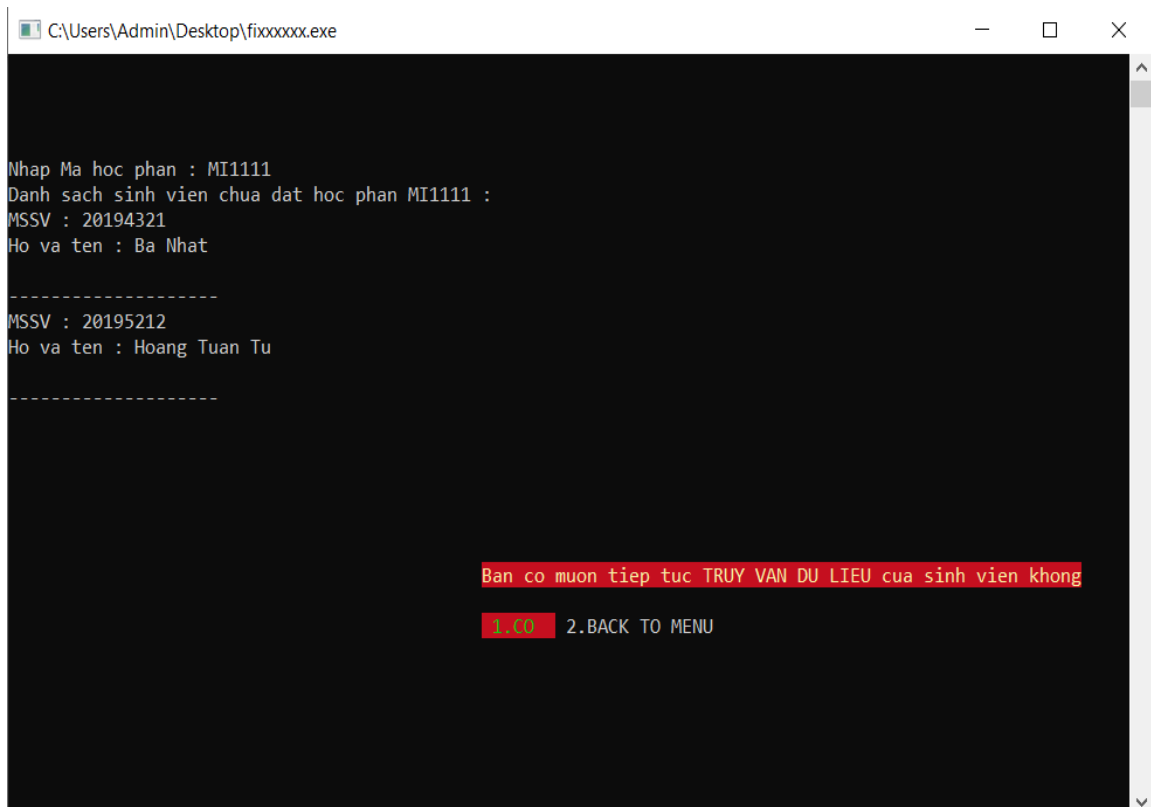
```
C:\Users\Admin\Desktop\fixxxxxx.exe

Diem trung binh tinh luy cua cac sinh vien :

MSSV : 20191234
CPA : 2.43
-----
MSSV : 20195212
CPA : 0.86
-----
MSSV : 20195855
CPA : 4.00
-----

Ban co muon tiep tuc TRUY VAN DU LIEU cua sinh vien khong
1.CO 2.BACK TO MENU
```

- Danh sách sinh viên chưa đạt học phần

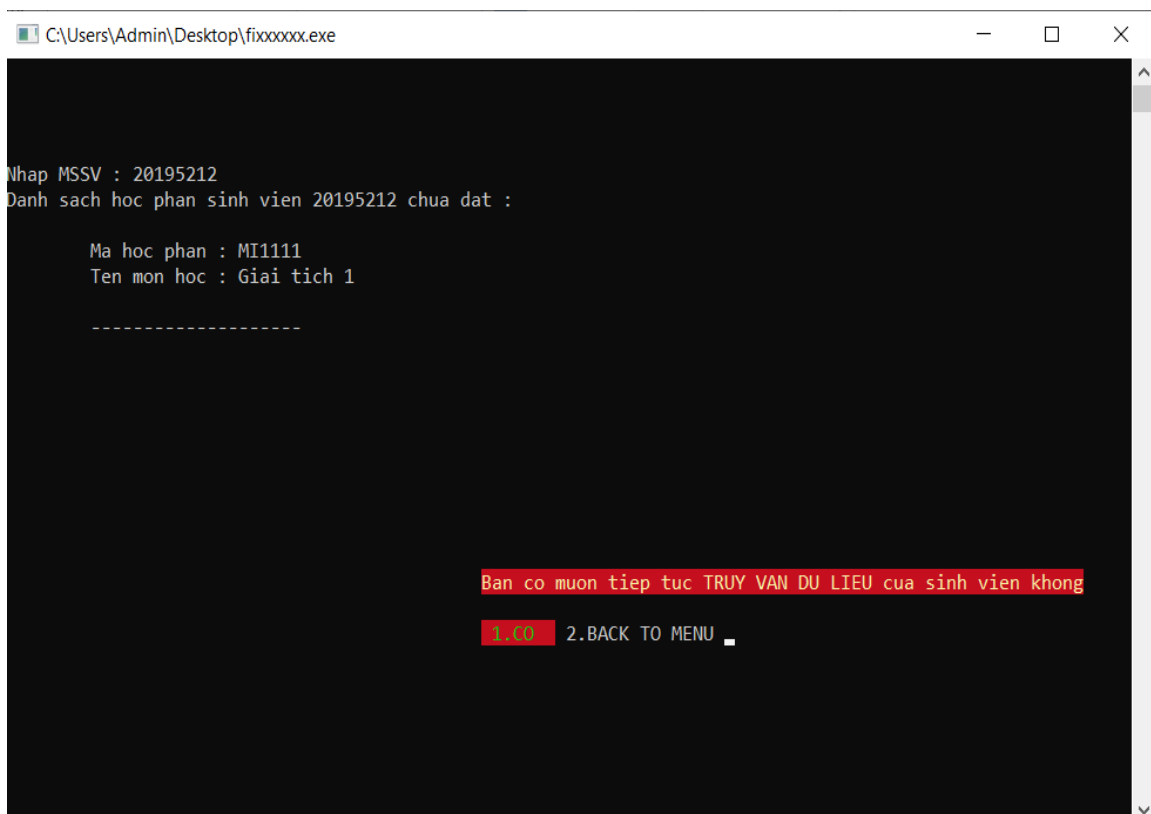


```
C:\Users\Admin\Desktop\fixxxxxx.exe

Nhap Ma hoc phan : MI1111
Danh sach sinh vien chua dat hoc phan MI1111 :
MSSV : 20194321
Ho va ten : Ba Nhat
-----
MSSV : 20195212
Ho va ten : Hoang Tuan Tu
-----

Ban co muon tiep tuc TRUY VAN DU LIEU cua sinh vien khong
1.CO 2.BACK TO MENU
```

- Danh sách học phần chưa đạt của một sinh viên



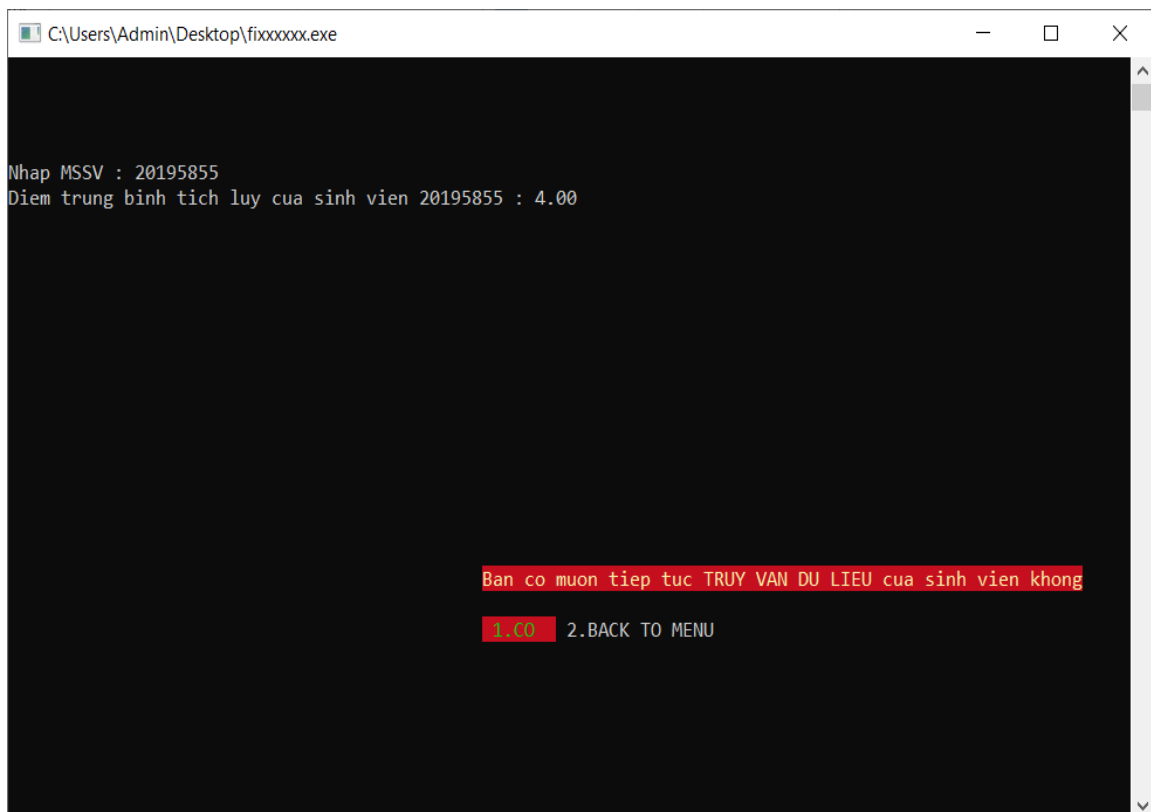
```
C:\Users\Admin\Desktop\fixxxxxx.exe

Nhap MSSV : 20195212
Danh sach hoc phan sinh vien 20195212 chua dat :

    Ma hoc phan : MI1111
    Ten mon hoc : Giai tich 1
    -----

Ban co muon tiep tuc TRUY VAN DU LIEU cua sinh vien khong
1.CO 2.BACK TO MENU
```

- Trung bình tích lũy của một sinh viên



```
C:\Users\Admin\Desktop\fixxxxxx.exe

Nhap MSSV : 20195855
Diem trung binh tich luy cua sinh vien 20195855 : 4.00

Ban co muon tiep tuc TRUY VAN DU LIEU cua sinh vien khong
1.CO 2.BACK TO MENU
```

- Kết quả được lưu trong file **NHATKY.text**

```
C:\Users\thong\Desktop\dochnatki.exe
Sinh vien da them :
MSSV      : 20199999
Ho va ten : Truong Trung Hieu
Gioi tinh : Nam
Ngay sinh : 5/3/2001
-----
Hoc phan da them :
Ma hoc phan : MI3041
Ten mon hoc : Giai tich so
So tin chi  : 2
-----
Diem hoc phan da them la :
MSSV      : 20199999
Ma hoc phan : MI1111
Diem      : 2.0
-----
Sinh vien 20199999 duoc sua thong tin moi :
MSSV      : 20199999
Ho va ten : Hieu Truong
Gioi tinh : Nam
Ngay sinh : 5/3/2001
-----
Hoc phan MI3041 duoc sua moi :
Ma hoc phan : MI3041
Ten mon hoc : GT so
So tin chi  : 2
-----
Diem hoc phan MI1111 cua sinh vien 0 duoc sua moi :
MSSV      : 20199999
Ma hoc phan : MI1111
Diem      : 2.5
-----

Gioi tinh : Nam
Ngay sinh : 5/3/2001
-----
Hoc phan bi xoa thong tin va toan bo diem lien quan :
Ma hoc phan : MI3041
Ten mon hoc : GT so
So tin chi  : 2
-----
Diem hoc phan da xoa :
MSSV      : 20191234
Ma hoc phan : MI1121
Diem      : 4.5
-----
MSSV : 20195855
CPA  : 4.00
-----
MSSV : 20195212
CPA  : 0.86
-----
MSSV : 20191234
CPA  : 3.50
-----
Sinh vien co MSSV      : 20195212 ,      ho va ten : Hoang Tuan Tu      chua dat hoc phan MI1111
Sinh vien 20195212 chua dat hoc phan MI1111
-----
Diem trung binh tich luy cua sinh vien 20195855 la : 4.00
-----
Process exited after 0.07542 seconds with return value 0
Press any key to continue . . .
```

III. Mã nguồn

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include <windows.h>
#include <conio.h>

void color(int color);
void gotoxy(int x, int y);
void drawframe();
void input();
void MENU();
void color(int color)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),color);
}

void gotoxy(int x, int y)
{
    COORD c;
    c.X=x;
    c.Y=y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),c);
}

void drawframe()
{
    int i;
    gotoxy(10,1);
    color(30);
    printf("-----QUAN LI SINH VIEN-----");
    color(10);
    for(i=0; i<65; i++){
        gotoxy(5+i,2);color(226);
        printf("-");
        gotoxy(5+i,17);
        printf("-");
    }
    for(i=0; i<=13; i++)
    {
        gotoxy(5,2);color(226);
        gotoxy(4,3+i);
        printf("||");
        gotoxy(69,3+i);
        printf("||");
    }
    for(i=3; i<= 16; i++)
    {
        gotoxy(6,i);color(176);
        printf(" ");
    }
}
```

```

    }
}

/*-----*/
struct Ngaysinh
{
    int ngay;
    int thang;
    int nam;
};
typedef struct Ngaysinh DATEE;

struct Sinhvien
{
    int mssv;
    char hoten[50];
    char gioitinh[10];
    DATEE ngaysinh;
};
typedef struct Sinhvien SINHVIEN;

struct nodesinhvien
{
    SINHVIEN data;
    struct nodesinhvien *Next;
};
typedef struct nodesinhvien NODESv;

struct danhsachSV
{
    NODESv *Head;
    NODESv *Tail;
};
typedef struct danhsachSV LISHsv;

struct Hocphan
{
    char mahp[30];
    char tenhp[50];
    int sotc;
};
typedef struct Hocphan HOCPHAN;

struct nodehocphan
{
    HOCPHAN data;
    struct nodehocphan *Next;
};
typedef struct nodehocphan NODEhp;

struct danhsachHP

```



```

    {
        NODEhp *Head;
        NODEhp *Tail;
    };
typedef struct danhsachHP LISHhp;

struct Diem
{
    char mahp[30];
    int mssv;
    float diemhp;
};
typedef struct Diem DIEM;

struct nodediem
{
    DIEM data;
    struct nodediem *Next;
};
typedef struct nodediem NODEd;

struct danhsachD
{
    NODEd *Head;
    NODEd *Tail;
};
typedef struct danhsachD LISHd;

struct trungbinhtichluy
{
    int mssv;
    float cpa;
};
typedef struct trungbinhtichluy cpa;

NODEsv *TimSV ( LISHsv lsv , int mssv )
{
    for ( NODEsv *k = lsv.Head ; k != NULL ; k = k->Next )
    {
        if ( k->data.mssv == mssv )
            return k;
    }
    return NULL;
}

NODEhp *TimHP ( LISHhp lhp , char mahp[] )
{
    int x;
    for ( NODEhp *k = lhp.Head ; k != NULL ; k = k->Next )
    {
        x = strcmp ( k->data.mahp , mahp );
        if ( x == 0 )

```

```

        return k;
    }
    return NULL;
}

NODEd *TimD ( LISHd ld , int mssv , char mahp[] )
{
    for ( NODEd *k = ld.Head ; k != NULL ; k = k->Next)
    {
        if (( strcmp(k->data.mahp,mahp) == 0) && (k->data.mssv ==
mssv ))
            return k;
    }
    return NULL;
}

void Nhatkythemsv ( SINHVIEN info )
{
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"Sinh vien da them :\n\n");
    fprintf(f,"MSSV      : %d\n",info.mssv);
    fprintf(f,"Ho va ten : %s\n",info.hoten);
    fprintf(f,"Gioi tinh : %s\n",info.gioitinh);
    fprintf(f,"Ngay sinh :
%d/%d/%d",info.ngaysinh.ngay,info.ngaysinh.thang,info.ngaysinh.nam);
    fprintf(f,"\n\n-----\n\n");
    fclose(f);
}

void Nhatkythemhp ( HOCPHAN info )
{
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"Hoc phan da them :\n\n");
    fprintf(f,"Ma hoc phan : %s\n",info.mahp);
    fprintf(f,"Ten mon hoc : %s\n",info.tenhp);
    fprintf(f,"So tin chi : %d",info.sotc);
    fprintf(f,"\n\n-----\n\n");
    fclose(f);
}

void Nhatkythemd ( DIEM info )
{
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"Diem hoc phan da them la :\n\n");
    fprintf(f,"MSSV      : %d\n",info.mssv);
    fprintf(f,"Ma hoc phan : %s\n",info.mahp);
    fprintf(f,"Diem      : %.1f",info.diemhp);
    fprintf(f,"\n\n-----\n\n");
    fclose(f);
}

void Nhatkyxoasv ( int mssv , LISHsv lsv)

```

```

{
    NODEsv *p = TimSV(lsv,mssv);
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"Sinh vien bi xoa thong tin va toan bo diem lien quan :\n\n");
    fprintf(f,"MSSV      : %d\n",p->data.mssv);
    fprintf(f,"Ho va ten : %s\n",p->data.hoten);
    fprintf(f,"Gioi tinh : %s\n",p->data.gioitinh);
    fprintf(f,"Ngay sinh : %d/%d/%d",p->data.ngaysinh.ngay,p-
>data.ngaysinh.thang,p->data.ngaysinh.nam);
    fprintf(f,"\n\n-----\n\n");
    fclose(f);
}

```

void Nhatkyxoahp (char mahp[30] , LISHhp lhp)

```

{
    NODEhp *p = TimHP(lhp,mahp);
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"Hoc phan bi xoa thong tin va toan bo diem lien quan :\n\n");
    fprintf(f,"Ma hoc phan : %s\n",p->data.mahp);
    fprintf(f,"Ten mon hoc : %s\n",p->data.tenhp);
    fprintf(f,"So tin chi  : %d",p->data.sotc);
    fprintf(f,"\n\n-----\n\n");
    fclose(f);
}

```

void Nhatkxoad (NODEd *p)

```

{
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"Diem hoc phan da xoa :\n\n");
    fprintf(f,"MSSV      : %d\n",p->data.mssv);
    fprintf(f,"Ma hoc phan : %s\n",p->data.mahp);
    fprintf(f,"Diem      : %.1f",p->data.diemhp);
    fprintf(f,"\n\n-----\n\n");
    fclose(f);
}

```

void Nhatkysuasv (NODEsv *p)

```

{
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"Sinh vien %d duoc sua thong tin moi :\n",p->data.mssv);
    fprintf(f,"MSSV      : %d\n",p->data.mssv);
    fprintf(f,"Ho va ten : %s\n",p->data.hoten);
    fprintf(f,"Gioi tinh : %s\n",p->data.gioitinh);
    fprintf(f,"Ngay sinh : %d/%d/%d",p->data.ngaysinh.ngay,p-
>data.ngaysinh.thang,p->data.ngaysinh.nam);
    fprintf(f,"\n\n-----\n\n");
    fclose(f);
}

```

void Nhatkysuahp (NODEhp *p)

```

{
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");

```

```

        fprintf(f,"Hoc phan %s duoc sua moi :\n",p->data.mahp);
        fprintf(f,"Ma hoc phan : %s\n",p->data.mahp);
        fprintf(f,"Ten mon hoc : %s\n",p->data.tenhp);
        fprintf(f,"So tin chi : %d",p->data.sotc);
        fprintf(f,"\n\n-----\n\n");
        fclose(f);
    }

void Nhatkysuad ( NODEd *p )
{
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"Diem hoc phan %s cua sinh vien %d duoc sua moi :\n",p->data.mahp,p-
>data.diemhp);
    fprintf(f,"MSSV      : %d\n",p->data.mssv);
    fprintf(f,"Ma hoc phan : %s\n",p->data.mahp);
    fprintf(f,"Diem      : %.1f",p->data.diemhp);
    fprintf(f,"\n\n-----\n\n");
    fclose(f);
}

void NhatkyTBtatcasv ( cpa c )
{
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"MSSV : %d\n",c.mssv);
    fprintf(f,"CPA : %.2f",c.cpa);
    fprintf(f,"\n\n-----\n\n");
    fclose(f);
}

void Nhatkysvchuadathp ( NODEsv *p , NODEd *k )
{
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"Sinh vien co MSSV      : %d ,\t",p->data.mssv);
    fprintf(f,"ho va ten : %s\t chua dat hoc phan %s",p->data.hoten,k->data.mahp);
    fprintf(f,"\n\n-----\n\n");
    fclose(f);
}

void Nhatkyhpchuadatcuasv ( NODEd *k )
{
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"Sinh vien %d chua dat hoc phan %s",k->data.mssv,k->data.mahp);

    fprintf(f,"\n\n-----\n\n");
    fclose(f);
}

void NhatkyTBCuasv ( int mssv , float cpa )
{
    FILE *f = fopen("C:\\Users\\Admin\\Desktop\\NHATKY.txt","a");
    fprintf(f,"Diem trung binh tich luy cua sinh vien %d la : %.2f",mssv,cpa);
    fprintf(f,"\n\n-----\n\n");
}

```

```

        fclose(f);
    }

void taoLISHhp ( LISHhp &lhp )
{
    lhp.Head = NULL;
    lhp.Tail = NULL;
}

NODEhp *taoNODEhp ( HOCPHAN info )
{
    NODEhp *p = (NODEhp *)malloc(sizeof(NODEhp));
    p->data = info;
    p->Next = NULL;
    return p;
}

void themNODEhpvaodau ( LISHhp &lhp , NODEhp *p )
{
    if ( lhp.Head == NULL )
        lhp.Head = lhp.Tail = p;
    else
    {
        p->Next = lhp.Head;
        lhp.Head = p;
    }
}

HOCPHAN InputHP ( LISHhp lhp )
{
    HOCPHAN info;
    printf("Nhap Ma hoc phan : "); fflush(stdin);
    scanf("%s", &info.mahp);
    NODEhp *p = TimHP(lhp,info.mahp);
    while ( p != NULL )
    {
        printf("Ma hoc phan da ton tai, moi nhap lai!!!\n");
        printf("Nhap Ma hoc phan : "); fflush(stdin);
        scanf("%s", &info.mahp);
        p = TimHP(lhp,info.mahp);
    }
    printf("Nhap ten mon hoc : "); fflush (stdin);
    gets(info.tenhp);
    printf("Nhap so tin chi : ");
    scanf("%d", &info.sotc);
    printf("\nThem hoc phan thanh cong !!!");
    Nhatkythemhp(info);
    return info;
}

void XoaNODEhpdau ( LISHhp &lhp )

```

```

    {
        if ( lhp.Head == NULL )
            return;
        NODEhp *p = lhp.Head;
        lhp.Head = lhp.Head->Next;
        p->Next = NULL;
        free(p);
    }

void XoaNODEhpcuoi ( LISHhp &lhp )
{
    if ( lhp.Head == NULL )
        return;
    for ( NODEhp *k = lhp.Head ; k != NULL ; k = k->Next )
    {
        if ( k->Next == lhp.Tail )
        {
            free (lhp.Tail);
            k->Next = NULL;
            lhp.Tail = k;
        }
    }
}

void SuaHP ( LISHhp lhp )
{
    HOCPHAN suahp;
    printf("Nhap Ma hoc phan : "); fflush(stdin);
    gets(suahp.mahp);
    NODEhp *p = TimHP(lhp,suahp.mahp);
    if ( p == NULL )
        printf("Ma hoc phan khong ton tai !\n");
    else
    {
        printf("Nhap ten mon hoc : "); fflush(stdin); gets(suahp.tenhp);
        printf("Nhap so tin chi : "); scanf("%d",&suahp.sotc);
        p->data = suahp;
        Nhatkysuahp(p);
        printf("\nSua hoc phan thanh cong!!!");
    }
}

void OutputHP( LISHhp lhp)
{
    if ( lhp.Head == NULL )
    {
        printf("Danh sach trong !!!");
        return;
    }
    for ( NODEhp *k = lhp.Head ; k != NULL ; k = k->Next )
    {
        printf("Ma hoc phan : %s\n",k->data.mahp);
    }
}

```

```

        printf("Ten mon hoc : %s\n",k->data.tenhp);
        printf("So tin chi : %d\n",k->data.sotc);
        printf("\n-----\n");
    }
}

void GhiLISHhpvaofilebin ( LISHhp lhp )
{
    FILE *fhp;
    fhp = fopen ("HOCPHAN.bin","wb");
    if ( fhp == NULL )
    {
        printf("Error!");
        exit(1);
    }
    for ( NODEhp *k = lhp.Head ; k != NULL ; k = k->Next )
    {
        fwrite ( &k -> data , sizeof(HOCPHAN) , 1 , fhp );
    }
    fclose(fhp);
}

void DocLISHhptufilebin ( LISHhp &lhp )
{
    FILE *fhp;
    HOCPHAN info;
    fhp = fopen ("HOCPHAN.bin","rb");
    if( fhp == NULL )
    {
        printf("Error!");
        exit(1);
    }
    while ( fread( &info, sizeof (HOCPHAN), 1, fhp) > 0 )
    {
        themNODEhpvaodau(lhp,taoNODEhp(info));
    }
    fclose(fhp);
}

void taoLISHsv ( LISHsv &lsv )
{
    lsv.Head = NULL;
    lsv.Tail = NULL;
}

NODEsv *taoNODEsv ( SINHVIEN info ) {
    NODEsv *p = (NODEsv *)malloc(sizeof(NODEsv));
    p->data = info;
    p->Next = NULL;
    return p;
}

```

```

void themNODESvvaodau ( LISHsv &lsv , NODESv *p )
{
    if ( lsv.Head == NULL )
        lsv.Head = lsv.Tail = p;
    else
    {
        p->Next = lsv.Head;
        lsv.Head = p;
    }
}

SINHVIEN InputSV ( LISHsv lsv )
{
    SINHVIEN info;
    printf("Nhap MSSV : "); fflush(stdin); scanf("%d", &info.mssv);
    NODESv *p = TimSV(lsv,info.mssv);
    while( p != NULL )
    {
        printf("Ma so sinh vien da ton tai ! Moi ban nhap lai !!!\n");
        printf("Nhap MSSV : "); fflush(stdin); scanf("%d", &info.mssv);
        p = TimSV(lsv,info.mssv);
    }
    printf("Nhap ho va ten sinh vien : "); fflush(stdin); gets(info.hoten);
    printf("Nhap gioi tinh : "); gets(info.gioitinh);
    printf("Nhap ngay thang nam sinh\n");
    printf("\t- Ngay : "); fflush(stdin); scanf("%d", &info.ngaysinh.ngay);
    printf("\t- Thang : "); fflush(stdin); scanf("%d", &info.ngaysinh.thang);
    printf("\t- Nam : "); fflush(stdin); scanf("%d", &info.ngaysinh.nam);
    printf("\n Them sinh vien thanh cong !!!");
    Nhatkythemsv(info);
    return info;
}

void XoaNODESvdau ( LISHsv &lsv )
{
    if ( lsv.Head == NULL )
        return;
    NODESv *p = lsv.Head;
    lsv.Head = lsv.Head->Next;
    p->Next = NULL;
    free(p);
}

void XoaNODESvcuoi ( LISHsv &lsv )
{
    if ( lsv.Head == NULL )
        return;
    for ( NODESv *k = lsv.Head ; k != NULL ; k = k->Next )
    {
        if ( k->Next == lsv.Tail )
            {

```



```

        free (lsv.Tail);
        k->Next = NULL;
        lsv.Tail = k;
    }
}

void SuaSV ( LISHsv lsv )
{
    SINHVIEN suasv;
    printf("Nhap MSSV : "); fflush(stdin); scanf("%d", &suasv.mssv);
    NODEsv *p = TimSV(lsv,suasv.mssv);
    if ( p == NULL )
        printf("MSSV khong ton tai !\n");
    else
    {
        printf("Nhap ho va ten moi : "); fflush(stdin); gets(suasv.hoten);
        printf("Nhap gioi tinh moi : "); gets(suasv.gioitinh);
        printf("Nhap ngay thang nam sinh moi :\n");
        printf("\t- Ngay : "); fflush(stdin); scanf("%d",
&suasv.ngaysinh.ngay);
        printf("\t- Thang : "); fflush(stdin); scanf("%d",
&suasv.ngaysinh.thang);
        printf("\t- Nam : "); fflush(stdin); scanf("%d",
&suasv.ngaysinh.nam);
        p->data = suasv;
        Nhatkysuasv(p);
        printf("\nSua sinh vien thanh cong !!!");
    }
}

void OutputSV ( LISHsv lsv )
{
    if ( lsv.Head == NULL )
    {
        printf("Danh sach trong !!!");
        return;
    }
    for ( NODEsv *k = lsv.Head ; k != NULL ; k = k->Next )
    {
        printf("MSSV : %d\n",k->data.mssv);
        printf("Ho va ten : %s\n",k->data.hoten);
        printf("Gioi tinh : %s\n",k->data.gioitinh);
        printf("Ngay sinh : %d/%d/%d\n",k->data.ngaysinh.ngay,k-
>data.ngaysinh.thang,k->data.ngaysinh.nam);
        printf("\n-----\n");
    }
}

void GhiLISHsvvaofilebin ( LISHsv lsv )
{
    FILE *fsv;

```

```

        fsv = fopen ("SINHVIEN.bin","wb");
        if( fsv == NULL )
        {
            printf("Error!");
            exit(1);
        }
        for ( NODEsv *k = lsv.Head ; k !=NULL ; k = k->Next )
        {
            fwrite (&k->data,sizeof(SINHVIEN),1,fsv);
        }
        fclose(fsv);
    }

void DocLISHsvtufilebin(LISHsv &lsv)
{
    FILE *fsv;
    SINHVIEN info;
    fsv = fopen ("SINHVIEN.bin","rb");
    if( fsv == NULL )
    {
        printf("Error!");
        exit(1);
    }
    while ( fread (&info,sizeof(SINHVIEN),1,fsv) > 0 )
    {
        themNODEsvvaodau(lsv,taoNODEsv(info));
    }
    fclose(fsv);
}

void taoLISHd ( LISHd &ld )
{
    ld.Head = NULL;
    ld.Tail = NULL;
}

NODEd *taoNODEd ( DIEM info )
{
    NODEd *p = (NODEd *)malloc(sizeof(NODEd));
    p->data = info;
    p->Next = NULL;
    return p;
}

void themNODEdvaodau ( LISHd &ld , NODEd *p )
{
    if ( ld.Head == NULL )
        ld.Head = ld.Tail = p;
    else
    {
        p->Next = ld.Head;
        ld.Head = p;
    }
}

```

```

    }
}

float Lamtron ( float a )
{
    float b = a;
    while (a>1)
        a-=1;
    if(a>=0 && a<0.25)
        b=b-a;
    else if(a>=0.25 && a<0.75)
        b=b-a+0.5;
    else b=b-a+1;
    return b;
}

bool KiemtramssvtrongLISHd ( LISHd ld , int mssv )
{
    for ( NODEd *k = ld.Head ; k != NULL ; k = k->Next )
    {
        if ( k->data.mssv == mssv )
            return true;
    }
    return false;
}

bool KiemtramahttrongLISHd ( LISHd ld , char mahp[] )
{
    for ( NODEd *k = ld.Head ; k != NULL ; k = k->Next )
    {
        if ( strcmp(k->data.mahp,mahp) == 0 )
            return true;
    }
    return false;
}

DIEM InputD ( LISHsv lsv , LISHhp lhp , LISHd ld )
{
    DIEM info;
    NODEsv *p1;
    NODEhp *p2;
    NODEd *p;
    do
    {
        printf("Nhap MSSV : "); fflush(stdin); scanf("%d",&info.mssv);
        p1 = TimSV(lsv,info.mssv);
        while ( p1 == NULL )
        {
            printf("MSSV khong ton tai ! Moi nhap lai !\n");
            printf("Nhap MSSV : "); fflush(stdin);
            scanf("%d",&info.mssv);
            p1 = TimSV(lsv,info.mssv);
        }
    }
}

```

```

        printf("Nhap Ma hoc phan : "); fflush(stdin); gets(info.mahp);
        p2 = TimHP(lhp,info.mahp);
        while ( p2 == NULL )
        {
            printf("Ma hoc phan khong ton tai ! Moi nhap lai
!\n");
            printf("Nhap Ma hoc phan : "); fflush(stdin);
            gets(info.mahp);
            p2 = TimHP(lhp,info.mahp);
        }
        p = TimD (ld,info.mssv,info.mahp);
        if ( p != NULL )
        {
            printf("Sinh vien da co diem voi hoc phan nay !!!
Moi nhap lai !!!\n");
        }
    }
    while ( p != NULL );
    printf("Nhap diem hoc phan : "); fflush(stdin); scanf("%f",&info.diemhp);
    while ((info.diemhp < 0 ) || (info.diemhp > 10 ))
    {
        printf("Loi nhap diem ! Moi nhap lai !\n");
        printf("Nhap diem hoc phan : "); fflush(stdin);
        scanf("%f",&info.diemhp);
    }
    info.diemhp = Lamtron(info.diemhp);
    printf("\nThem diem thanh cong !!!");
    Nhatkythemd(info);
    return info;
}

void XoaNODEddau( LISHd &ld )
{
    if ( ld.Head == NULL )
        return;
    NODEd *p = ld.Head;
    ld.Head = ld.Head->Next;
    p->Next = NULL;
    free(p);
}

void XoaNODEdcuoi ( LISHd &ld )
{
    if ( ld.Head == NULL )
        return;
    for ( NODEd *k = ld.Head ; k != NULL ; k = k->Next )
    {
        if ( k->Next == ld.Tail )
        {
            free (ld.Tail);
            k->Next = NULL;
            ld.Tail = k;

```

```

    }
}

void XoaNODEd ( LISHsv &lsv , LISHhp &lhp , LISHd &ld )
{
    int mssv;
    char mahp[30];
    NODEd *j;
    NODEd *p;
    if ( ld.Head == NULL )
    {
        printf("Danh sach diem hoc phan rong !!!\n");
        return;
    }
    printf("Nhap MSSV : "); fflush(stdin); scanf("%d",&mssv);
    if ( TimSV(lsv,mssv) == NULL )
    {
        printf("Khong ton tai MSSV!!!\n");
        return;
    }
    printf("Nhap Ma hoc phan : "); fflush (stdin); gets(mahp);
    if ( TimHP(lhp,mahp) == NULL )
    {
        printf("Khong ton tai Ma hoc phan !!!\n");
        return;
    }
    p = TimD(ld,mssv,mahp);
    if ( p == NULL )
    {
        printf("Khong co MSSV voi Ma hoc phan nay!\n");
        return;
    }
    Nhatkyxoad(p);
    if((ld.Head->data.mssv == p->data.mssv) && (strcmp(ld.Head->data.mahp,p-
>data.mahp) == 0 ))
    {
        XoaNODEddau(ld);
        printf("\nXoa diem thanh cong !!!");
        return;
    }
    if(( ld.Tail->data.mssv == p->data.mssv) && (strcmp (ld.Tail->data.mahp,p-
>data.mahp) == 0 ))
    {
        XoaNODEdcuoi(ld);
        printf("\nXoa diem thanh cong !!!");
        return;
    }
    for( NODEd *k = ld.Head ; k != NULL ; k = k->Next )
    {
        if ((k->data.mssv == p->data.mssv) && (strcmp(k->data.mahp,p-
>data.mahp) == 0))

```

```

        {
            j->Next = k->Next;
            free(k);
            printf("\nXoa diem thanh cong !!!");
            return;
        }
        j = k;
    }
}

void SuaD ( LISHsv lsv , LISHhp lhp , LISHd ld )
{
    int mssv;
    char mahp[30];
    NODEd *p;
    printf("Nhap MSSV : "); fflush(stdin); scanf("%d",&mssv);
    if ( TimSV(lsv,mssv) == NULL )
    {
        printf("Khong ton tai MSSV!!!\n");
        return;
    }
    printf("Nhap Ma hoc phan : "); fflush (stdin); gets(mahp);
    if ( TimHP(lhp,mahp) == NULL )
    {
        printf("Khong ton tai Ma hoc phan !!!\n");
        return;
    }
    p = TimD(ld,mssv,mahp);
    if ( p == NULL )
    {
        printf("Khong co MSSV voi Ma hoc phan nay!\n");
        return;
    }
    printf("Nhap diem hoc phan : "); fflush (stdin); scanf("%f",&p->data.diemhp);
    while ((p->data.diemhp < 0) || (p->data.diemhp > 10))
    {
        printf("Loi nhap diem ! Moi nhap lai !\n");
        printf("Nhap diem hoc phan : "); fflush(stdin); scanf("%f",&p-
>data.diemhp);
    }
    p->data.diemhp = Lamtron(p->data.diemhp);
    Nhatkysuad(p);
    printf("\nSua diem thanh cong!!!");
}

void OutputD ( LISHd ld )
{
    if ( ld.Head == NULL )
    {
        printf("Danh sach trong !!!");
        return;
    }
}

```

```

        for( NODEd *k = ld.Head ; k != NULL ; k = k->Next )
        {
            printf("MSSV : %d\n",k->data.mssv);
            printf("Ma hoc phan : %s\n",k->data.mahp);
            printf("Diem hoc phan : %.1f\n",k->data.diemhp);
            printf("\n-----\n");
        }
    }

void GhiLISHdvaofilebin ( LISHd ld )
{
    FILE *fd;
    fd = fopen ("DIEM.bin","wb");
    if( fd == NULL )
    {
        printf("Error!");
        exit(1);
    }
    for ( NODEd *k = ld.Head ; k !=NULL ; k = k->Next )
    {
        fwrite (&k->data,sizeof(DIEM),1,fd);
    }
    fclose(fd);
}

void DocLISHdtufilebin( LISHd &ld )
{
    FILE *fd;
    DIEM info;
    fd = fopen ("DIEM.bin","rb");
    if( fd == NULL )
    {
        printf("Error!");
        exit(1);
    }
    while ( fread( &info, sizeof(DIEM), 1, fd) > 0 )
    {
        themNODEdvaodau(ld,taoNODEd(info));
    }
    fclose(fd);
}

void XoaNODEdkhibietmssv ( LISHd &ld , int mssv )
{
    NODEd *j;
    if( ld.Head->data.mssv == mssv )
        XoaNODEddau(ld);
    if( ld.Tail->data.mssv == mssv )
        XoaNODEdcuoi(ld);
    for( NODEd *k = ld.Head ; k != NULL ; k = k->Next )
    {
        if ( k->data.mssv == mssv )

```

```

        {
            j->Next = k->Next;
            free(k);
            return;
        }
        j = k;
    }
}

void XoaNODEdkhibietmahp ( LISHd &ld , char mahp[] )
{
    NODEd *j;
    if( strcmp(ld.Head->data.mahp,mahp) == 0 )
        XoaNODEddau(ld);
    if (strcmp(ld.Tail->data.mahp,mahp) == 0 )
        XoaNODEdcuoi(ld);
    for( NODEd *k = ld.Head ; k != NULL ; k = k->Next )
    {
        if (strcmp(k->data.mahp,mahp) == 0 )
        {
            j->Next = k->Next;
            free(k);
            return;
        }
        j = k;
    }
}

void XoaNODEsv ( LISHsv &lsv , LISHd &ld )
{
    int mssv;
    NODEsv *j = (NODEsv *)malloc(sizeof(NODEsv));
    if (lsv.Head == NULL) {
        printf("Danh sach sinh vien rong !!!\n");
        return;
    }
    printf("Nhap MSSV : "); fflush(stdin); scanf("%d",&mssv);
    NODEsv *p = TimSV(lsv,mssv);
    if ( p == NULL )
    {
        printf("MSSV khong ton tai!\n");
        return;
    }
    Nhatkyxoasv(mssv,lsv);
    while ( KiemtramssvtrongLISHd(ld,mssv) )
        XoaNODEdkhibietmssv(ld,mssv);
    if ( lsv.Head->data.mssv == mssv )
    {
        XoaNODEsvdau(lsv);
        printf("\nXoa sinh vien thanh cong");
        return;
    }
}

```



```

if ( lsv.Tail->data.mssv == mssv )
{
    XoaNODEsvcuoi(lsv);
    printf("\nXoa sinh vien thanh cong");
    return;
}
for( NODEsv *k = lsv.Head ; k != NULL ; k = k->Next )
{
    if ( k->data.mssv == mssv )
    {
        j->Next = k->Next;
        free(k);
        printf("\nXoa sinh vien thanh cong");
        return;
    }
    j = k;
}
}

void XoaNODEhp ( LISHhp &lhp , LISHd &ld )
{
    int x;
    char mahp[30];
    NODEhp *p;
    NODEhp *j = (NODEhp *)malloc(sizeof(NODEhp));
    if ( lhp.Head == NULL )
    {
        printf("Danh sach hoc phan rong !!!\n");
        return;
    }
    printf("Nhap Ma hoc phan : "); fflush(stdin); gets(mahp);
    p = TimHP(lhp,mahp);
    if( p == NULL )
    {
        printf("Khong ton tai ma hoc phan!\n");
        return;
    }
    Nhatkyxoahp(mahp,lhp);
    while( KiemtramahptrongLISHd(ld,mahp) )
        XoaNODEdkhibietmahp(ld,mahp);
    if ( strcmp(lhp.Head->data.mahp,mahp) == 0 )
    {
        XoaNODEhpdau(lhp);
        printf("\nXoa hoc phan thanh cong !!!");
        return;
    }
    if( strcmp(lhp.Tail->data.mahp,mahp) == 0 )
    {
        XoaNODEhpcuoi(lhp);
        printf("\nXoa hoc phan thanh cong !!!");
        return;
    }
}

```

```

for( NODEhp *k = lhp.Head ; k != NULL ; k = k->Next )
{
    x = strcmp(k->data.mahp,mahp);
    if ( x == 0 )
    {
        j->Next = k->Next;
        free(k);
        printf("\nXoa hoc phan thanh cong !!!");
        return;
    }
    j = k;
}

float Quydoidiem ( float x )
{
    if (x < 4)
    {
        return 0;
    }
    if ((x >= 4) && (x < 5))
    {
        return 1;
    }
    if ((x >= 5) && (x < 5.5))
    {
        return 1.5;
    }
    if ((x >= 5.5) && (x < 6.5))
    {
        return 2;
    }
    if ((x >= 6.5) && (x < 7))
    {
        return 2.5;
    }
    if ((x >= 7) && (x < 8))
    {
        return 3;
    }
    if ((x >= 8) && (x < 8.5))
    {
        return 3.5;
    }
    if (x >= 8.5)
    {
        return 4;
    }
}

void Tinhtrungbinhtichluy( LISHsv lsv , LISHhp lhp , LISHd ld )
{

```

```

    NODEhp *k;
    float S;
    int Stc;
    FILE *fcpa;
    cpa c;
    fcpa = fopen ("CPA.bin","wb");
    if ( fcpa == NULL )
    {
        printf("Error!");
        return;
    }
    printf("Diem trung binh tich luy cua cac sinh vien :\n\n");
    for ( NODEsv *i = lsv.Head ; i != NULL ; i = i->Next )
    {
        S = 0;
        Stc = 0;
        for ( NODEd *j = ld.Head ; j != NULL ; j = j->Next )
        {
            if( i->data.mssv == j->data.mssv )
            {
                k = TimHP(lhp,j->data.mahp);
                S += (k->data.sotc)*Quydoidiem(j-
>data.diemhp);

                Stc += k->data.sotc;

            }
        }
        if( Stc == 0 )
        {
            c.mssv = i->data.mssv;
            c.cpa = Stc;
            printf("MSSV : %d\n",i->data.mssv);
            printf("CPA : %.2f\n",Stc);
            printf("\n-----\n");
            fwrite(&c, sizeof(cpa),1,fcpa);
            NhatkyTBtatcasv(c);
        }
        else
        {
            c.mssv = i->data.mssv;
            c.cpa = S/Stc;
            printf("MSSV : %d\n",i->data.mssv);
            printf("CPA : %.2f\n",S/Stc);
            printf("\n-----\n");
            fwrite(&c,sizeof(cpa),1,fcpa);
            NhatkyTBtatcasv(c);
        }
    }
    fclose(fcpa);
}

void Danhsachsvchuadathp( LISHsv lsv , LISHhp lhp , LISHd ld )

```

```

{
    int count = 0;
    NODEsv *p;
    NODEhp *j;
    char mahp[30];
    printf("Nhap Ma hoc phan : "); fflush(stdin); gets(mahp);
    j = TimHP(lhp,mahp);
    if ( j == NULL )
    {
        printf("Khong ton tai Ma hoc phan !!!\n");
        return;
    }
    if ( KiemtramahptrongLISHd(ld,mahp) == false )
    {
        printf("Khong co sinh vien hoc mon nay!!!\n");
        return;
    }
    printf("Danh sach sinh vien chua dat hoc phan %s :\n",mahp);
    for ( NODEd *k = ld.Head ; k != NULL ; k = k->Next )
    {
        if ((strcmp(k->data.mahp,mahp) == 0) && (k->data.diemhp < 4))
        {
            count++;
            p = TimSV(lsv,k->data.mssv);
            printf("MSSV : %d\n",p->data.mssv);
            printf("Ho va ten : %s\n",p->data.hoten);
            printf("\n-----\n");
            Nhatkysvchuadathp(p,k);
        }
    }
    if ( count == 0 )
        printf("Khong co sinh vien nao chua dat hoc phan nay!!!\n");
}

void Danhsachhpchuadatcuasv ( LISHsv lsv , LISHhp lhp , LISHd ld )
{
    int count = 0;
    int mssv;
    NODEhp *p;
    NODEsv *j;
    printf("Nhap MSSV : "); fflush(stdin); scanf("%d", &mssv);
    j = TimSV(lsv,mssv);
    if( j == NULL )
    {
        printf("Khong ton tai MSSV!!!\n");
        return;
    }
    if ( KiemtramssvtrongLISHd(ld,mssv) == false )
    {
        printf("Sinh vien %d chua hoc mon nao !!!\n",mssv);
        return;
    }
}

```

```

printf("Danh sach hoc phan sinh vien %d chua dat : \n",mssv);
for ( NODEd *k = ld.Head ; k != NULL ; k = k->Next )
{
    if ((k->data.mssv == mssv) && (k->data.diemhp < 4))
    {
        count++;
        p = TimHP(lhp,k->data.mahp);
        printf("\n\tMa hoc phan : %s\n",k->data.mahp);
        printf("\tTen mon hoc : %s\n",p->data.tenhp);
        printf("\n\t-----\n");
        Nhatkyhpchuadatcuasv(k);
    }
}
if ( count == 0 )
    printf("Danh sach trong!!!");
}

void Trungbinhtichluysv ( LISHsv lsv , LISHhp lhp , LISHd ld )
{
    NODEhp *p;
    NODEsv *j;
    float S = 0;
    int Stc = 0;
    int mssv;
    printf("Nhap MSSV : "); fflush(stdin); scanf("%d",&mssv);
    j = TimSV(lsv,mssv);
    if( j == NULL )
    {
        printf("Khong ton tai MSSV!!!\n");
        return;
    }
    for ( NODEd *k = ld.Head ; k != NULL ; k = k->Next )
    {
        if ( k->data.mssv == mssv )
        {
            p = TimHP(lhp,k->data.mahp);
            S += (p->data.sotc)*Quydoidiem(k->data.diemhp);
            Stc += p->data.sotc;
        }
    }
    if ( Stc == 0 )
    {
        printf("Diem trung binh tinh luy cua sinh vien %d :
%.2f\n",mssv,Stc);
        NhatkyTBcuasv(mssv,Stc);
    }
    else
    {
        printf("Diem trung binh tinh luy cua sinh vien %d :
%.2f\n",mssv,S/Stc);
        NhatkyTBcuasv(mssv,Stc);
    }
}

```



```

LISHd ld;
LISHsv lsv;
taoLISHd(ld);
taoLISHhp(lhp);
taoLISHsv(lsv);
DocLISHsvtufilebin(lsv);
DocLISHhptufilebin(lhp);
DocLISHdtufilebin(ld);
int Set[] = {7,7,7,7,7,7}; // default colors
int counter = 1;
char key;

while (1)
{
    MENUPOP:
        drawframe();
        gotoxy(0,0);
        color(90);
        if (counter == 1)
        {
            Set[0] = 20;
        }
        gotoxy(10, 4);
        color(Set[0]);
        printf(" 1. HIEN THI DANH SACH ");

        gotoxy(10, 6);
        color(Set[1]);
        printf(" 2. NHAP DU LIEU ");

        gotoxy(10, 8);
        color(Set[2]);
        printf(" 3. SUA DU LIEU ");

        gotoxy(10, 10);
        color(Set[3]);
        printf(" 4. XOA DU LIEU");

        gotoxy(10, 12);
        color(Set[4]);
        printf(" 5. TRUY VAN DU LIEU ");

        gotoxy(10, 14);
        color(Set[5]);
        printf(" 6. OUT ");

        key = getch();
        if (key == 72 && (counter >= 2 && counter <= 6))
        {
            counter--;
        }
        else if (key == 80 && (counter >= 1 && counter <= 5))

```

```

        {
            counter++;
        }
    else if (key == 72 && (counter <= 1))
    {
        counter = 6;
    }
    else if (key == 80 && (counter >= 6))
    {
        counter = 1;
    }

if (key == '\r')
{
    if (counter == 6)
    {
        gotoxy(0, 17);
        break;
    }
    system("cls");
    gotoxy(0, 5);

    /*1.XEM DANH SACH-----*/

    if (counter == 1)
    {
        while (1)
        {
            int Set[] = {7,7,7}; // default colors
            int counter = 1, i = 0;
            char key;
            while (1)
            {
                gotoxy(8,2);
                color(26);

printf("HIEN THI DANH SACH:");

                                if (counter == 1)
                                    Set[0] = 74;

                                gotoxy(8, 4);
                                color(Set[0]);
                                printf(" 1.SINH

VIEN ");

                                gotoxy(8, 6);
                                color(Set[1]);
                                printf(" 2.HOC

PHAN ");

```



```

gotoxy(8, 8);
color(Set[2]);
printf(" 3.DIEM SO

");

(counter >= 2 && counter <= 3))

&& (counter >= 1 && counter <= 2))

&& (counter <= 1))

&& (counter >= 3))

{
(counter == 3)

{
gotoxy(0, 17);
color(7);
}

system("cls");
gotoxy(0, 5);

(counter == 1)

{
OutputSV(lsv);
break;
}

}

```

```

gotoxy(8, 8);
color(Set[2]);
printf(" 3.DIEM SO

```

```

key = getch();
if (key == 72 &&

```

```

counter--;

```

```

else if (key == 80

```

```

counter++;

```

```

else if (key == 72

```

```

counter = 3;
else if (key == 80

```

```

counter = 1;

```

```

if (key == '\r')

```

```

if

```

```

if

```

```

if

```

```

(counter == 2)
    {
        OutputHP(lhp);
        break;
    }
    if
(counter == 3)
    {
        OutputD(ld);
        break;
    }

    key =
    if (key
    == '\r')
        system("cls");
    }

    Set[0] = 7;
    Set[1] = 7;
    Set[2] = 7;
    if (counter == 1)
        Set[0] = 74;

    if (counter == 2)
        Set[1] = 74;

    if (counter == 3)
        Set[2] = 74;

    }
    gotoxy(45, 20);
    color(78);
    printf("Ban co muon tiep
    Tuc xem cac danh sach con lai khong");

    int Set1[] = {7,7}; //

    default colors

    int counter1 = 1;
    char key1;
    while (1)
        {
            if (counter1 ==

```

1)

```

    Set1[0] = 74;
    gotoxy(45, 22);
    color(Set1[0]);
    printf(" 1.CO ");
    gotoxy(52, 22);
    color(Set1[1]);
    printf(" 2.BACK

TO MENU ");

    key1 = getch();

if (key1 == 75 &&

    counter1--;

    else if (key1

        counter1++;

    else if (key1 == 75

        counter1 = 2;
    else if (key1 == 77

        counter1 = 1;
        if

(key1 == '\r')

    {

        if (counter1 == 2)

            color(7);

            system("cls");

            gotoxy(0, 5);

            if (counter1 == 1)

                {

                    system("cls");

                    break;

                }

            if (counter1 == 2)

                goto MENUPOP;
```

```

key1 = getch();

if (key1 == '\r')

    system("cls");

}

Set1[0] = 7;
Set1[1] = 7;
if (counter1 == 1)
    Set1[0] = 74;

if (counter1 == 2)
    Set1[1] = 74;

}

}

}

/*-----NHAP DU LIEU-----
----- */

if (counter == 2)
{
    while (1)
    {
        int Set[] = {7,7,7}; // default colors
        int counter = 1, i = 0;
        char key;
        while (1)
        {
            gotoxy(8,2);
            color(26);

printf("NHAP DU LIEU:");

if (counter == 1)
    Set[0] = 74;
gotoxy(8, 4);
color(Set[0]);
printf(" 1.SINH

VIEN ");

gotoxy(8, 6);
color(Set[1]);
printf(" 2.HOC

PHAN ");

gotoxy(8, 8);
color(Set[2]);
printf(" 3.DIEM SO

```

```

");

(counter >= 2 && counter <= 3))

== 80 && (counter >= 1 && counter <= 2))

== 72 && (counter <= 1))

&& (counter >= 3))

{
    (counter == 3)
        {
            gotoxy(0, 17);
            color(7);
        }
        system("cls");
        gotoxy(0, 5);
        (counter == 1)
            {
                NODEsv * p = taoNODEsv(InputSV(lsv));
                themNODEsvvaodau(lsv, p);

                break;
            }
        (counter == 2)
            {
                NODEhp * p = taoNODEhp(InputHP(lhp));

```

```

key = getch();
if (key == 72 &&

        counter--;
        else if (key

        counter++;
        else if (key

        counter = 3;
    else if (key == 80

        counter = 1;

    if (key == '\r')

```

```

        if

```

```

        if

```

```

        if

```

```

        themNODEhpvaodau(lhp, p);

        break;

    }

    if
(counter == 3)

    {

        NODEd * p = taoNODEd(InputD(lsv, lhp, ld));

        themNODEdvaodau(ld, p);

        break;

    }

    key =
getch();

    if (key
== '\r')

        system("cls");
    }

    Set[0] = 7;
    Set[1] = 7;
    Set[2] = 7;
    if (counter == 1)
        Set[0] = 74;

    if (counter == 2)
        Set[1] = 74;

    if (counter == 3)
        Set[2] = 74;

    }

    gotoxy(45, 20);
    color(78);
    printf("Ban co muon tiep tuc

NHAP DU LIEU khong");

int Set2[] = {7,7}; // default colors
int counter2 = 1;
char key2;
while (1)

    {
        if (counter2 == 1)
            Set2[0] = 74;
        gotoxy(45, 22);
        color(Set2[0]);

```

| | |
|---------------------|---------------------|
| | printf(" 1.CO "); |
| | gotoxy(52, 22); |
| | color(Set2[1]); |
| | printf(" 2.BACK TO |
| MENU "); | |
| | key2 = getch(); |
| (counter2 == 2)) | if (key2 == 75 && |
| | counter2--; |
| | else if (key2 == 77 |
| && (counter2 == 1)) | counter2++; |
| | else if (key2 == 75 |
| && (counter2 <= 1)) | counter2 = 2; |
| | else if (key2 == 77 |
| && (counter2 >= 2)) | counter2 = 1; |
| | if (key2 == '\r') |
| { | |
| | if |
| (counter2 == 2) | |
| color(7); | |
| system("cls"); | |
| gotoxy(0, 5); | |
| | if |
| (counter2 == 1) | |
| { | |
| system("cls"); | |
| break; | |
| } | |
| | if |
| (counter2 == 2) | |
| | goto |
| MENUPOP; | |
| | key2 = |
| getch(); | |
| | if |
| (key2 == '\r') | |
| system("cls"); | |

```

}

Set2[0] = 7;
Set2[1] = 7;
if (counter2 ==

1)

Set2[0] = 74;

if (counter2 ==

2)

Set2[1] = 74;
}
}
}

/*-----SUA DU LIEU-----
----- */

if (counter == 3)
{
while (1)
{
int Set[] = {7,7,7}; // default colors
int counter = 1, i = 0;
char key;
while (1)
{
gotoxy(8,2);
color(26);

printf("SUA DU LIEU:");

if (counter == 1)
Set[0] = 74;
gotoxy(8, 4);
color(Set[0]);
printf(" 1.SINH

VIEN ");

gotoxy(8, 6);
color(Set[1]);
printf(" 2.HOC

PHAN ");

gotoxy(8, 8);
color(Set[2]);
printf(" 3.DIEM SO

");

key = getch();
if (key == 72 &&

(counter >= 2 && counter <= 3))

counter--;
else if

(key == 80 && (counter >= 1 && counter <= 2))

counter++;
else if

```



```
(key == 72 && (counter <= 1))
```

```
&& (counter >= 3))
```

```
{
```

```
(counter == 3)
```

```
{
```

```
gotoxy(0, 17);
```

```
color(7);
```

```
}
```

```
system("cls");
```

```
gotoxy(0, 5);
```

```
(counter == 1)
```

```
{
```

```
SuaSV(lsv);
```

```
break;
```

```
}
```

```
(counter == 2)
```

```
{
```

```
SuaHP(lhp);
```

```
break;
```

```
}
```

```
(counter == 3)
```

```
{
```

```
SuaD(lsv,lhp,ld);
```

```
break;
```

```
counter = 3;  
else if (key == 80
```

```
counter = 1;
```

```
if (key == '\r')
```

```
if
```

```
if
```

```
if
```

```
if
```

```

    }

    getch();

    == '\r')

    system("cls");

```

```

SUA DU LIEU khong");

colors

```

```

MENU ");

```

```

(counter3 == 2))

```

```

&& (counter3 == 1))

```

```

&& (counter3 <= 1))

```

```

key =

if (key

```

```

    }

    Set[0] = 7;
    Set[1] = 7;
    Set[2] = 7;
    if (counter == 1)
        Set[0] = 74;
    if (counter == 2)
        Set[1] = 74;
    if (counter == 3)
        Set[2] = 74;
}

```

```

gotoxy(45, 20);
color(78);
printf("Ban co muon tiep tuc

```

```

int Set3[] = {7,7}; // default

```

```

int counter3 = 1;
char key3;
while (1)
{
    if (counter3 == 1)
        Set3[0] = 74;
    gotoxy(45, 22);
    color(Set3[0]);
    printf(" 1.CO ");
    gotoxy(52, 22);
    color(Set3[1]);
    printf(" 2.BACK TO

```

```

    key3 = getch();
    if (key3 == 75 &&

```

```

        counter3--;

```

```

    else if (key3 == 77

```

```

        counter3++;

```

```

    else if (key3 == 75

```

```

        counter3 = 2;
    else if (key3 == 77

```

```
&& (counter3 >= 2))
```

```
        counter3 = 1;  
if (key3 == '\r')
```

```
    {  
(counter3 == 2)
```

```
if
```

```
    color(7);
```

```
    system("cls");
```

```
    gotoxy(0, 5);
```

```
if
```

```
(counter3 == 1)
```

```
    {
```

```
        system("cls");
```

```
    break;
```

```
    }
```

```
if
```

```
(counter3 == 2)
```

```
goto
```

```
MENUPOP;
```

```
key3 =
```

```
getch();
```

```
if
```

```
(key3 == '\r')
```

```
system("cls");
```

```
    }
```

```
Set3[0] = 7;  
Set3[1] = 7;  
if (counter3 == 1)  
    Set3[0] = 74;  
if (counter3 == 2)  
    Set3[1] = 74;
```

```
    }
```

```
    }
```

```
    }
```

```
/*-----XOA DU LIEU-----*/
```

```
----- */
```

```
if (counter == 4)
```

```
{
```

```
while (1)
```

```

{
int Set[] = {7,7,7}; // default colors
int counter = 1, i = 0;
char key;
while (1)
{
gotoxy(8,2);
color(26);

printf("XOA DU LIEU:");

VIEN ");

PHAN ");

");

(counter >= 2 && counter <= 3))

&& (counter >= 1 && counter <= 2))

&& (counter <= 1))

&& (counter >= 3))

{
(counter == 3)

{
gotoxy(0, 17);
color(7);

if (counter == 1)
Set[0] = 74;
gotoxy(8, 4);
color(Set[0]);
printf(" 1.SINH

gotoxy(8, 6);
color(Set[1]);
printf(" 2.HOC

gotoxy(8, 8);
color(Set[2]);
printf(" 3.DIEM SO

key = getch();
if (key == 72 &&

counter--;

else if (key == 80

counter++;

else if (key == 72

counter = 3;
else if (key == 80

counter = 1;

if (key == '\r')
if

```

```

}

system("cls");

gotoxy(0, 5);

(counter == 1)

{

XoaNODEsv(lsv, ld);

break;

}

(counter == 2)

{

XoaNODEhp(lhp, ld);

break;

}

(counter == 3)

{

XoaNODEd(lsv,lhp,ld);

break;

}

getch();

== 'r')

system("cls");

```

```

Set[0] = 7;
Set[1] = 7;
Set[2] = 7;
if (counter == 1)
    Set[0] = 74;

if (counter == 2)
    Set[1] = 74;
if (counter == 3)

```

XOA DU LIEU khong");

colors

MENU ");

(counter5 == 2))

&& (counter5 == 1))

&& (counter5 <= 1))

&& (counter5 >= 2))

{
(counter5 == 2)

color(7);
system("cls");

gotoxy(0, 5);

(counter5 == 1)

{

```
Set[2] = 74;
    }
gotoxy(45, 20);
color(78);
printf("Ban co muon tiep tuc

int Set5[] = {7,7}; // default

int counter5 = 1;
char key5;
while (1)
    {
        if (counter5 == 1)
            Set5[0] = 74;

        gotoxy(45, 22);
        color(Set5[0]);
        printf(" 1.CO ");
        gotoxy(52, 22);
        color(Set5[1]);
        printf(" 2.BACK TO

        key5 = getch();
        if (key5 == 75 &&

            counter5--;

        else if (key5 == 77

            counter5++;
        else if (key5 == 75

            counter5 = 2;
        else if (key5 == 77

            counter5 = 1;
        if (key5 == '\r')
```

if

if

```

system("cls");

break;

}

(counter5 == 2)
MENUPOP;
getch();
(key5 == '\r')
system("cls");

if
goto
key5 =
if

}

Set5[0] = 7;
Set5[1] = 7;
if (counter5 == 1)
    Set5[0] = 74;

if (counter5 == 2)
    Set5[1] = 74;

}

}

}

if (counter == 5)
{
while (1)
{
int Set[] = {7,7,7,7}; // default colors
int counter = 1, i = 0;
char key;
while (1)
{
gotoxy(8,2);
color(26);

printf("TRUY VAN DU LIEU CUA SINH VIEN:");

if (counter == 1)
    Set[0] = 20;
gotoxy(2, 3);
color(Set[0]);
printf(" 1. Diem trung

binh tinh luy cua tat ca sinh vien ");

gotoxy(2, 4);

```

sinh vien khong dat hoc phan ");

hoc phan chua dat cua sinh vien ");

binh tinh luy cua 1 sinh vien");

(counter >= 2 && counter <= 4))

&& (counter >= 1 && counter <= 3))

&& (counter <= 1))

&& (counter >= 4))

{

(counter == 4)

{

gotoxy(0, 17);

color(7);

}

system("cls");

gotoxy(0, 4);

(counter == 1)

{

color(Set[1]);
printf(" 2. Danh sach

gotoxy(2, 5);
color(Set[2]);
printf(" 3. Danh sach

gotoxy(2, 6);
color(Set[3]);
printf(" 4. Diem trung

gotoxy(2, 10);
key = getch();

if (key == 72 &&

counter--;

else if (key == 80

counter++;

else if (key == 72

counter = 4;

else if (key == 80

counter = 1;

if (key == '\r')

if

if


```

    Tinhtrungbinhtichluy(lsv, lhp, ld);
break;
    }
    if
(counter == 2)
    {
Danhsachsvchuadathp(lsv, lhp, ld);
break;
    }
    if
(counter == 3)
    {
Danhsachhpcchuadatcuasv(lsv, lhp, ld);
break;
    }
    if
(counter == 4)
    {
Trungbinhtichluysv(lsv, lhp, ld);
break;
    }
    key =
getch();
    if (key
== '\r')
system("cls");
    }

Set[0] = 7;
Set[1] = 7;
Set[2] = 7;
Set[3] = 7;
if (counter == 1)
    Set[0] = 20;
if (counter == 2)
    Set[1] = 20;

```

```

        if (counter == 3)
            Set[2] = 20;
        if (counter == 4)
            Set[3] = 20;

    }
    gotoxy(45, 20);
    color(78);
    printf("Ban co muon tiep tuc

TRUY VAN DU LIEU cua sinh vien khong");

colors

int Set5[] = {7,7}; // default

int counter5 = 1;
char key5;
while (1)
    {
        if (counter5 == 1)
            Set5[0] = 74;

        gotoxy(45, 22);
        color(Set5[0]);
        printf(" 1.CO ");
        gotoxy(52, 22);
        color(Set5[1]);
        printf(" 2.BACK TO

MENU ");

        (counter5 == 2))

        && (counter5 == 1))

        && (counter5 <= 1))

        && (counter5 >= 2))

        {
            if
(counter5 == 2)

            color(7);

            system("cls");

```

```

gotoxy(0, 5);

(counter5 == 1)
    {
        system("cls");
        break;
    }

(counter5 == 2)
    MENUPOP;
    getch();
    (key5 == '\r')
        system("cls");

Set5[0] = 7;
Set5[1] = 7;
if (counter5 == 1)
    Set5[0] = 74;
if (counter5 == 2)
    Set5[1] = 74;
}

}

}

Set[0] = 7;
Set[1] = 7;
Set[2] = 7;
Set[3] = 7;
Set[4] = 7;
Set[5] = 7;
if (counter == 1)
    Set[0] = 20;
if (counter == 2)
    Set[1] = 20;
if (counter == 3)
    Set[2] = 20;
if (counter == 4)
    Set[3] = 20;
if (counter == 5)
    Set[4] = 20;
if (counter == 6)
    Set[5] = 20;

```

```

        GhiLISHsvvaofilebin(lsv);
        GhiLISHhpvaofilebin(lhp);
        GhiLISHdvaofilebin(ld);
        Ghitrungbinhtichluyvaofilebin(lsv, lhp, ld);
    }
}

```

PHẦN KẾT LUẬN

* Đánh giá kết quả đạt được

- Về cấu trúc dữ liệu:

+ Do danh sách sinh viên, học phần, điểm thi được cài đặt dưới dạng danh sách liên kết nên dễ dàng cho việc thêm, xóa, sửa.

- Về thuật toán:

+ Thuật toán có độ phức tạp là $O(n1.n2.n3)$ (trong đó $n1$ là số sinh viên, $n2$ là số học phần, $n3$ là số điểm thi), vì vậy thời gian thực hiện chương trình sẽ khá chậm khi có khối lượng dữ liệu lớn.

+ Thuật toán đáp ứng được những yêu cầu đặt ra của bài toán.