

## LAB 04. DANH SÁCH LIÊN KẾT

### Bài 1:

Điểm giữa kỳ môn Cấu trúc dữ liệu và Thuật giải 1 chiếm tối đa 4 điểm trong điểm tổng kết cuối cùng môn học ( tương đương 40% ), là tổng của 3 cột điểm thành phần sau :

- Điểm chuyên cần : chiếm tối đa 0.5 đ ( tương đương 12,5 % điểm giữa kỳ)
- Điểm bài lab : tối đa 1.5 đ ( tương đương 37,5% điểm giữa kỳ )
- Điểm kiểm tra kiến thức một phần của môn học : tối đa 2 đ ( tương đương 50% điểm giữa kỳ)

Viết chương trình tính và xuất điểm giữa kỳ của môn học cho các sinh viên đã đăng ký tham gia môn học. Yêu cầu chương trình là xuất ra bảng điểm giữa kỳ, thông tin của mỗi sinh viên bao gồm: Mã sinh viên, Họ, tên, năm sinh, điểm giữa kỳ.

- Lý lịch sinh viên cho trong tập tin : “LLSV.text”

Mã SV	Họ	Tên	Năm Sinh	Nguyên quán
2410249	Nguyen	Canh	1986	Lam_Dong
2410271	Hoang	Dat	1985	Khanh_Hoa
2410272	Nguyen	Dat	1986	Ninh_Thuan
2410263	Khen	Di	1984	Lam_Dong
2410258	Kieu	Duyen	1986	Binh_Thuan
2413022	To	Duyen	1985	Lam_Dong
2413031	Do	Huong	1985	Binh_Dinh
2410295	Nguyen	Huy	1986	Quang_Ngai
2410302	Nguyen	Khanh	1984	Da_Nang
2413041	Vo	Kiet	1983	Binh_Thuan
2410316	Lam	Le	1986	Lam_Dong
2410312	K'	Luyt	1986	Lam_Dong
2413047	Nguyen	Mai	1985	Ninh_Thuan
2410327	Dang	Nga	1986	Binh_Thuan
2410336	Nguyen	Ngan	1985	Binh_Thuan
2410349	Vo	Phong	1986	Binh_Dinh
2410387	Vo	Tam	1984	Quang_Ngai
2410401	Nguyen	Thao	1985	Binh_Thuan
2410415	Mong	Vu	1986	Lam_Dong
2410420	Luu	Y	1986	Da_Nang

- Các cột điểm thành phần của điểm giữa kỳ cho trong tập tin “DiemGK.txt” :

Mã SV	Điểm Chuyên cần	Điểm bài lab	Điểm kiểm tra
2410249	0	1.5	1.5
2410271	0.5	1.5	2
2410272	0	1.5	1
2410263	0.5	1.5	1
2410258	0.5	1.5	1
2413022	0.5	1.5	1.5
2413031	0	1.5	1
2410295	0.5	1	1.5
2410302	0.5	1.5	1.5
2413041	0.5	1.5	1.5
2410316	0.5	1.5	1.5
2410312	0.5	1.5	2
2413047	0.5	1.5	1

2410327	0.5	1.5	1
2410336	0	1.5	1.5
2410349	0.5	1.5	1
2410387	0	1.5	2
2410401	0.5	1.5	1.5
2410415	0	1.5	1
2410420	0.5	1.5	2

---

Thực hiện :

Tạo dự án Win32 Console Application mới. Đặt tên là Lab04\_BaiLT\_01\_MaSV, với các tập tin rỗng **thuvien.h** (trong Header Files), **Program.cpp** (trong Source File). Không tạo tập tin menu.h, vì không có tùy chọn chức năng.

Tạo các tập tin dữ liệu theo yêu cầu chương trình.

Bổ sung dần vào các tập tin cho đến khi hoàn chỉnh, mỗi lần bổ sung chức năng cần chạy ngay chương trình để kiểm tra.

- Nội dung tập tin “**thuvien.h**”:

//Định nghĩa kiểu dữ liệu list: LL

```
struct LLSinhVien
{
    char maSV[8];
    char ho[12];
    char ten[12];
    int namSinh;
    char nguyenQuan[15];
};
```

```
struct tagNodeLL
{
    LLSinhVien info;
    tagNodeLL *pNext;
};
```

```
typedef tagNodeLL NODELL;
```

```
struct LL
{
    NODELL *pHead;
    NODELL *pTail;
};
```

---

//Định nghĩa kiểu dữ liệu list: BANGDIEM

```
struct DiemTP
```

```
{
    char maSV[8];
    double dcc;
    double dblab;
    double dkt;
    double dgk;
};
```

```
struct tagNodeDiem
```

```
{
    DiemTP info;
    tagNodeDiem *pNext;
};
```

```
typedef tagNodeDiem NODEBD;
```

```

struct BD
{
    NODEBD *pHead;
    NODEBD *pTail;
};
//=====
NODELL* GetNode_LL(LLSinhVien x);
void CreatList_LL(LL &l);
void InsertTail_LL(LL &l, LLSinhVien x);
void TapTin_List_LL(char *filename, LL &l);
//=====
void TapTin_List_BD(char *filename, BD &l);
void InsertTail_BD(BD &l, DiemTP x);
void CreatList_BD(BD &l);
NODEBD* GetNode_BD(DiemTP x);
//=====
void XuatBangDiem(LL l, BD g);
void TieuDe();
//=====
//Tao nut moi LL
NODELL* GetNode_LL(LLSinhVien x)
{
    NODELL *p;
    p = new NODELL;
    if (p != NULL)
    {
        p->info = x;
        p->pNext = NULL;
    }
    return p;
}
//Khoi tao DSLK don ly lich rong
void CreatList_LL(LL &l)
{
    l.pHead = l.pTail = NULL;
}
//Chen x vao cuoi ds
void InsertTail_LL(LL &l, LLSinhVien x)
{
    NODELL* new_ele = GetNode_LL(x);
    if (new_ele == NULL)
    {
        cout << "\nKhong du bo nho";
        system("CLS");
        return;
    }
    if (l.pHead == NULL)
    {
        l.pHead = new_ele; l.pTail = l.pHead;
    }
    else
    {

```

```

        l.pTail->pNext = new_ele;
        l.pTail = new_ele;
    }
}
//Chuyen du lieu tap tin diem thanh phan sang List l
void TapTin_List_LL(char *filename, LL &l)
{
    LLSinhVien x;
    ifstream in(filename);
    if (!in)
    {
        cout << "\nLoi mo file !\n";
        system("PAUSE");
        return;
    }
    CreatList_LL(l);
    char maSV[8];
    char ho[12];
    char ten[12];
    int namSinh;
    char nguyenQuan[15];
    in >> maSV; strcpy_s(x.maSV, maSV);
    in >> ho; strcpy_s(x.ho, ho);
    in >> ten; strcpy_s(x.ten, ten);
    in >> namSinh; x.namSinh = namSinh;
    in >> nguyenQuan; strcpy_s(x.nguyenQuan, nguyenQuan);
    InsertTail_LL(l, x);
    while (!in.eof())
    {
        in >> maSV; strcpy_s(x.maSV, maSV);
        in >> ho; strcpy_s(x.ho, ho);
        in >> ten; strcpy_s(x.ten, ten);
        in >> namSinh; x.namSinh = namSinh;
        in >> nguyenQuan; strcpy_s(x.nguyenQuan, nguyenQuan);
        InsertTail_LL(l, x);
    }
    in.close();
}
//=====
//Bang diem
//Tao nut moi BD
NODEBD* GetNode_BD(DiemTP x)
{
    NODEBD *p;
    p = new NODEBD;
    if (p != NULL)
    {
        p->info = x;
        p->pNext = NULL;
    }
    return p;
}

```

```

//Khoi tao DSLK don ly lich rong
void CreatList_BD(BD &l)
{
    l.pHead = l.pTail = NULL;
}
//Chen x vao cuoi ds
void InsertTail_BD(BD &l, DiemTP x)
{
    NODEBD* new_ele = GetNode_BD(x);
    if (new_ele == NULL)
    {
        cout << "\nKhong du bo nho";
        system("CLS");
        return;
    }
    if (l.pHead == NULL)
    {
        l.pHead = new_ele; l.pTail = l.pHead;
    }
    else
    {
        l.pTail->pNext = new_ele;
        l.pTail = new_ele;
    }
}
//Chuyen du lieu tap tin diem thanh phan sang List BD
void TapTin_List_BD(char *filename, BD &l)
{
    DiemTP x;
    ifstream in(filename);
    if (!in)
    {
        cout << "\nLoi mo file !\n";
        system("PAUSE");
        return;
    }
    CreatList_BD(l);
    char maSV[8];
    double dcc;
    double dblab;
    double dkt;
    double dgk;
    in >> maSV; strcpy_s(x.maSV, maSV);
    in >> dcc; x.dcc = dcc;
    in >> dblab; x.dblab = dblab;
    in >> dkt; x.dkt = dkt;
    dgk = dcc + dblab + dkt; x.dgk = dgk;
    InsertTail_BD(l, x);
    while (!in.eof())
    {
        in >> maSV; strcpy_s(x.maSV, maSV);
        in >> dcc; x.dcc = dcc;
    }
}

```

```

        in >> dblab; x.dblab = dblab;
        in >> dkt; x.dgk = dkt;
        dgk = dcc + dblab + dkt; x.dgk = dgk;
        InsertTail_BD(l, x);
    }
    in.close();
}
// Xuat ket qua bang diem
//Xuat tieu de
void TieuDe()
{
    cout << "\n=====\\n";
    cout << setiosflags(ios::left);
    cout << setw(10) << "Ma SV"
        << setw(15) << "Ho"
        << setw(15) << "Ten"
        << setw(10) << "Nam Sinh"
        << setw(15) << "Diem GK";
    cout << "\n=====\\n";
}
void XuatBangDiem(LL l, BD g)
{
    TieuDe();
    NODELL *p = l.pHead;
    NODEBD *q=g.pHead;
    while (p != NULL && q != NULL)
    {
        cout << endl;
        cout << setiosflags(ios::left)
            << setw(10) << p->info.maSV
            << setw(15) << p->info.ho
            << setw(15) << p->info.ten
            << setw(10) << p->info.namSinh
            << setw(15) << setprecision(2) << setiosflags(ios::fixed) << q->info.dgk;
        p = p->pNext;
        q = q->pNext;
    }
    cout << "\n=====\\n";
}

```

- Nội dung tập tin “Program.cpp”:

```

#include <iostream>
#include <fstream>
#include <string.h>
#include <iomanip>
using namespace std;
#include "Thuvien.h"
void ChayChuongTrinh();
int main()
{
    ChayChuongTrinh();
}

```

```

        return l;
    }
    void ChayChuongTrinh()
    {
        LL l;
        BD g;
        TapTin_List_LL("LLSV.txt", l);
        TapTin_List_BD("DiemGK.txt", g);
        XuatBangDiem(l,g);
        cout << endl;
        system("PAUSE");
    }
}

```

## Bài 2:

Giả sử có một danh sách nhân viên, mỗi nhân viên được lưu trữ các thông tin:

- Mã nhân viên, //chuỗi có 7 ký tự, không có ký tự trắng
- Họ của nhân viên, // chuỗi có không quá 10 ký tự
- Tên lót nhân viên, // một chuỗi có không quá 10 ký tự
- Tên nhân viên, // chuỗi có không quá 10 ký tự
- Địa chỉ, // chuỗi có không quá 15 ký tự
- Năm sinh, //số nguyên dương 4 ký số
- Lương, //số thực dương

- Tập tin dữ liệu “text.txt “ lưu trữ danh sách nhân viên :

Mã NV	Họ	Tên lót	Tên	Địa Chỉ	NSinh	Lương
LD12045	Nguyen	Tuan	Vo	Lam_Dong	1980	25000000
LD13210	Ly	Van	Hoa	Ninh_Thuan	1985	30000000
LD13452	Tran	Ngoc	Ninh	Khanh_Hoa	1974	10000000
LD14432	Nguyen	_	Vo	Phu_Yen	1985	12000000
LD15332	Le	Thi	Lieu	Binh_Dinh	1974	10000000
LD22032	Van	Thi	Hoa	Lam_Dong	1984	10000000
LD22052	Vo	Ngoc	Hoa	Lam_Dong	1984	70000000
LD22140	Tran	Vuong	Vo	Binh_Dinh	1990	12000000
LD22145	Le	Thi	Vo	Khanh_Hoa	1986	12000000
LD23045	Tran	Trong	Hieu	Ha_Noi	1991	25000000
LD24042	Ly	Van	Hoa	Ha_Noi	1983	30000000
LD30432	Nguyen	_	Vo	Lam_Dong	1975	12000000

Viết chương trình tùy chọn thực hiện các thao sau trên danh sách nhân viên:

1. Tách danh sách nhân viên thành 2 danh sách, danh sách đầu gồm những nhân viên có lương  $\leq x$ , danh sách sau gồm những nhân viên còn lại.
2. Tách danh sách nhân viên theo cách luân phiên từng nhân viên theo thứ tự vào 2 danh sách.
3. Đảo ngược danh sách.
4. Sắp danh sách tăng dần theo tên, tên trùng thì tăng dần theo họ; tên và họ trùng thì tăng dần theo tên lót.

Thực hiện :

Tạo dự án Win32 Console Application mới. Đặt tên là Lab04\_BaiLT\_02\_MaSV, với các tập tin rỗng **thuvien.h**, **menu.h** (trong Header Files), **Program.cpp** (trong Source File).

Tạo tập tin dữ liệu “text.txt” theo yêu cầu chương trình.

Bổ sung dần vào các tập tin cho đến khi hoàn chỉnh, mỗi lần bổ sung chức năng mới cần chạy ngay chương trình để kiểm tra.

- Nội dung tập tin “thuvien.h”:

//Định nghĩa kiểu dữ liệu : LIST

struct nhanvien

{

char maNV[8];  
char hoNV[10];  
char tenLot[10];  
char ten[10];  
char diachi[12];  
int namSinh;  
double luong;

};

typedef nhanvien data;

struct tagNode

{

data info;  
tagNode \*pNext;

};

typedef tagNode NODE;

struct LIST

{

NODE \*pHead;  
NODE \*pTail;

};

//=====

NODE\* GetNode(data x);

void CreatList(LIST &l);

int IsEmpty(LIST l);

void InsertHead(LIST &l, data x);

void InsertTail(LIST &l, data x);

//=====

void TapTin\_List(char \*filename, LIST &l);

void Xuat\_DS NV(LIST l);

void Xuat\_NV(data p);

void TieuDe();

//=====

void Tach\_Luong\_x(LIST l, double x);

void Tach\_LuanPhien(LIST l);

void DaoNguoc\_DS(LIST l);

void Remove\_Ten(LIST &l, char ten[10]);

int RemoveNode\_First(LIST &l, char ten[10]);

void List\_SapTang\_Ten\_Ho\_TLot(LIST &l);

void List\_Selection\_Sort(LIST &l);

void Hoanvi(data &x, data &y);

//=====

//Tạo nút mới

NODE\* GetNode(data x)

{

NODE \*p;  
p = new NODE;  
if (p != NULL)  
{



```

        p->info = x;
        p->pNext = NULL;
    }
    return p;
}
//Khoi tao DSLK don rong
void CreatList(LIST &l)
{
    l.pHead = l.pTail = NULL;
}
//Kiem tra danh sach co rong
int IsEmpty(LIST l)
{
    if (l.pHead == NULL)
        return 1;
    return 0;
}
//Chen x vao dau ds
void InsertHead(LIST &l, data x)
{
    NODE* new_ele = GetNode(x);
    if (new_ele == NULL)
    {
        cout << "\nKhong du bo nho";
        system("PAUSE");
        return;
    }
    if (l.pHead == NULL) //DS rong
    {
        l.pHead = new_ele;
        l.pTail = l.pHead;
    }
    else
    {
        new_ele->pNext = l.pHead; //chen vao dau DS
        l.pHead = new_ele;
    }
}
//Chen x vao cuoi ds
void InsertTail(LIST &l, data x)
{
    NODE* new_ele = GetNode(x);
    if (new_ele == NULL)
    {
        cout << "\nKhong du bo nho";
        system("CLS");
        return;
    }
    if (l.pHead == NULL)
    {
        l.pHead = new_ele; l.pTail = l.pHead;
    }
}

```

```

        else
        {
            l.pTail->pNext = new_ele;
            l.pTail = new_ele;
        }
    }
}
//=====
//Chuyen du lieu tap tin dang list
void TapTin_List(char *filename, LIST &l)
{
    data x;
    ifstream in(filename);
    if (!in)
    {
        cout << "\nLoi mo file !\n";
        system("PAUSE");
        return;
    }
    CreatList(l);
    char maNV[8];
    char hoNV[10];
    char tenLot[10];
    char ten[10];
    char diachi[12];
    int namSinh;
    double luong;
    in >> maNV; strcpy_s(x.maNV, maNV);
    in >> hoNV; strcpy_s(x.hoNV, hoNV);
    in >> tenLot; strcpy_s(x.tenLot, tenLot);
    in >> ten; strcpy_s(x.ten, ten);
    in >> diachi; strcpy_s(x.diachi, diachi);
    in >> namSinh; x.namSinh = namSinh;
    in >> luong; x.luong = luong;
    InsertTail(l, x);
    while (!in.eof())
    {
        in >> maNV; strcpy_s(x.maNV, maNV);
        in >> hoNV; strcpy_s(x.hoNV, hoNV);
        in >> tenLot; strcpy_s(x.tenLot, tenLot);
        in >> ten; strcpy_s(x.ten, ten);
        in >> diachi; strcpy_s(x.diachi, diachi);
        in >> namSinh; x.namSinh = namSinh;
        in >> luong; x.luong = luong;
        InsertTail(l, x);
    }
    in.close();
}
//Xuat tieu de
void TieuDe()
{
    cout
    <<"\n=====n";
}

```

```

        cout << setiosflags(ios::left);
        cout << setw(10) << "Ma NV"
            << setw(12) << "Ho"
            << setw(12) << "Ten lot"
            << setw(12) << "Ten"
            << setw(14) << "Dia Chi"
            << setw(6) << "NSinh"
            << setw(12) << "Luong";
        cout<<"\n===== \n";
    }
    //Xuat 1 nhan vien
    void Xuat_NV(data p)
    {
        cout << setiosflags(ios::left)
            << setw(10) << p.maNV
            << setw(12) << p.hoNV
            << setw(12) << p.tenLot
            << setw(12) << p.ten
            << setw(14) << p.diachi
            << setw(6) << p.namSinh
            << setw(12) << setprecision(2) << setiosflags(ios::fixed) << p.luong;
    }
    //Xuat danh sach nhan vien
    void Xuat_DS NV(LIST l)
    {
        NODE *p;
        if (IsEmpty(l))
        {
            cout << "\nDS rong!\n";
            return;
        }
        Tieude();
        p = l.pHead;
        while (p != NULL)
        {
            Xuat_NV(p->info);
            cout << "\n";
            p = p->pNext;
        }
        Cout<<"\n===== \n";
    }

    void Tach_Luong_x(LIST l, double x)
    {
        NODE *p;
        LIST l1, l2;
        p = l.pHead;
        if (p == NULL)
        {
            cout << "\nDS l rong";
            system("PAUSE");
            return;
        }
    }

```

```

    }
    CreatList(l1);
    CreatList(l2);
    while (p != NULL)
    {
        if (p->info.luong <= x)
            InsertTail(l1, p->info);
        else
            InsertTail(l2, p->info);
        p = p->pNext;
    }
    cout << "\n- Danh sach l1 (luong <= " << x << "):\n";
    Xuat_DS NV(l1);
    cout << "\n- Danh sach l2 (luong > " << x << "):\n";
    Xuat_DS NV(l2);
    cout << endl;
}
void Tach_LuanPhien(LIST l)
{
    NODE *p;
    LIST l1, l2;
    p = l.pHead;
    if (p == NULL)
    {
        cout << "\nDS l rong";
        system("PAUSE");
        return;
    }
    int k = 1; //khoa
    CreatList(l1);
    CreatList(l2);
    while (p != NULL)
    {
        if (k == 1)
            InsertTail(l1, p->info);
        else
            InsertTail(l2, p->info);
        p = p->pNext;
        k = 3 - k;
    }
    cout << "\n- Danh sach l1:\n";
    Xuat_DS NV(l1);
    cout << "\n- Danh sach l2 :\n";
    Xuat_DS NV(l2);
    cout << endl;
}
void DaoNguoc_DS(LIST l)
{
    NODE *p;
    LIST l1;
    p = l.pHead;
    if (p == NULL)

```

```

    {
        cout << "\nDS l rong";
        system("PAUSE");
        return;
    }
CreatList(l1);
while (p != NULL)
{
    InsertHead(l1, p->info);
    p = p->pNext;
}
cout << "\n- Danh sach dao nguoc cua l:\n";
Xuat_DSNV(l1);
}
//Sap tang theo ten
void List_Selection_Sort(LIST &l)
{
    NODE *min, *p, *q;
    p = l.pHead;
    while (p != l.pTail)
    {
        min = p;
        q = p->pNext;
        while (q != NULL)
        {
            if (_strcmpi(q->info.ten, min->info.ten) < 0)
                min = q;
            q = q->pNext;
        }
        Hoanvi(min->info, p->info);
        p = p->pNext;
    }
}
void List_SapTang_Ten_Ho_TLot(LIST &l)
{
    List_Selection_Sort(l);
    NODE *p, *q;
    for (p = l.pHead; p != l.pTail; p = p->pNext)
        for (q = p->pNext; q != NULL; q = q->pNext)
            if (_strcmpi(p->info.ten, q->info.ten) == 0)
                if (_strcmpi(p->info.hoNV, q->info.hoNV) > 0)
                    Hoanvi(q->info, p->info);
    for (p = l.pHead; p != l.pTail; p = p->pNext)
        for (q = p->pNext; q != NULL; q = q->pNext)
            if (_strcmpi(p->info.ten, q->info.ten) == 0 && _strcmpi(p->info.hoNV, q->info.hoNV) == 0)
                if (_strcmpi(p->info.tenLot, q->info.tenLot) > 0)
                    Hoanvi(q->info, p->info);
}

```

- Nội dung các tập tin menu.h, program.cpp : Tự viết.