Trường: **ĐH CNTP TP.HCM** Khoa: **Công nghệ thông tin** Bộ môn: **Công nghệ phần mềm.** MH: **TH Cấu trúc dữ liệu & giải thuật**	**BÀI 2. DANH SÁCH LIÊN KẾT ĐƠN** (tiếp theo)	

A. MỤC TIÊU:

- Cài đặt được danh sách liên kết đơn để lưu dữ liệu là các cấu trúc do người dùng định nghĩa
- Xử lý tính toán trên danh sách liên kết đơn đã xây dựng

B. DỤNG CỤ - THIẾT BỊ THÍ NGHIỆM CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

I. Tóm tắt lý thuyết

1. Một số phương thức trên danh sách liên kết đơn (tiếp theo)

a. Xóa nút đầu danh sách

- **Bước 1:** Nếu danh sách rỗng thì: Không thực hiện gì cả.
- **Bước 2:** Khi (head != NULL) thì: thực hiện các công việc sau:
  - + **Bước 2.1:** p = head;
  - + **Bước 2.2:** head = head→next;
  - + **Bước 2.3:** Nếu (head == NULL) thì: tail = NULL;
  - + **Bước 2.4:** Lưu lại thông tin nút bị xóa;
  - + **Bước 2.5:** delete p; *//Hủy nút do p trỏ đến (con trỏ p)*

b. Xóa nút cuối danh sách

- **Bước 1:** Nếu (tail == NULL) thì: không thực hiện gì cả.
- **Bước 2:** Khi (tail != NULL) thì: thực hiện các công việc sau:
  - + **Bước 2.1:** Gán q = head; và p = tail;
  - + **Bước 2.2:** Nếu (p == q) thì: head = tail = NULL;
  - + **Bước 2.3:** Ngược lại: Tìm đến nút kế trước nút tail.
  - + **Bước 2.4:** tail = q;
  - + **Bước 2.5:** tail→next = NULL;
  - + **Bước 2.6:** Lưu lại thông tin nút bị xóa;

+ **Bước 2.7:** delete p; `//Hủy nút do p trỏ đến`

c. Xóa một nút sau nút q của danh sách

- **Bước 1:** Nếu (q == NULL hoặc q→next == NULL) thì: không thực hiện.
- **Bước 2:** Khi (q != NULL và q→next != NULL) thì: thực hiện các công việc sau:
  + **Bước 2.1:** p = q→next;
  + **Bước 2.2:** q→next = p→next;
  + **Bước 2.3:** Nếu (p == tail) thì: tail = q;
  + **Bước 2.4:** Lưu lại thông tin nút bị xóa;
  + **Bước 2.5:** delete p; *//Hủy nút do p trỏ đến*

d. Xóa nút có giá trị x của danh sách

- **Bước 1:** Nếu danh sách rỗng thì: Không thực hiện gì cả.
- **Bước 2:** Tìm phần tử p có khóa bằng x, và q là phần tử đứng kế trước p.
- **Bước 3:** Nếu (p == NULL) thì: Không có nút chứa x nên không thực hiện.
- **Bước 4:** *//Ngược lại (p != NULL) → nghĩa là tồn tại nút p chứa khóa x*
  + Nếu (p == head) thì: *//p là nút đầu danh sách → xóa đầu*

    Deletehead(**sl**, x);
  + Ngược lại:

    DeleteAfter(**sl**, q, x); *//Xóa nút p chứa x kế sau nút q*

e. Xóa toàn bộ danh sách

- **Bước 1:** Nếu danh sách rỗng thì: Không thực hiện gì cả.
- **Bước 2:** Lặp lại trong khi (danh sách còn phần tử) thì thực hiện lần lượt những việc sau:
  + **Bước 2.1:** Gán p = head;
  + **Bước 2.2:** Gán head = head→next; *//Cập nhật con trỏ head*
  + **Bước 2.3:** delete p; *//Hủy nút do p trỏ đến*
- **Bước 3:** Gán tail = NULL; *//bảo toàn tính nhất quán khi danh sách rỗng*

f. Sắp xếp danh sách (tăng dần trên trường khóa)

– Do danh sách liên kết đơn chỉ có một chiều đi từ phần tử đầu tới phần tử kế tiếp nên chỉ áp dụng được các thuật toán sắp xếp thỏa tính chất như vậy như: InterchangeSort, SelectionSort, …

– Có thể thay thế a[i] trong mảng bằng i->info, a[i+1] thay thế bằng (i->next)->info, i++ thì thay bằng i=i->next
– Các bước của thuật toán áp dụng giống như trong mảng.

2. Tạo danh sách liên kết đơn chứa thông tin cấu trúc (struct) và đọc dữ liệu struct vào dslk đơn từ file.

Xét cấu trúc BaiHat chứa thông tin bài hát gồm

– Tên bài hát là một chuỗi tối đa 50 ký tự
– Tác giả là một chuỗi tối đa 40 ký tự
– Ca sĩ là một chuỗi tối đa 40 ký tự
– Thời lượng là một số nguyên (đơn vi là giây)

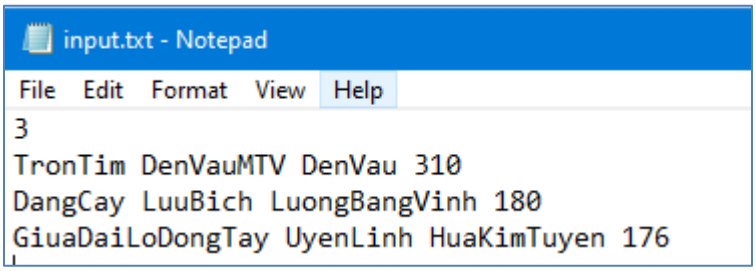
a. Xây dựng danh sách liên kết đơn chứa.

```
struct BaiHat
{
    char tenBH[51];
    char tenTG[41];
    char tenCS[41];
    int thoiLuong;
};

struct Node
{
    BaiHat info;
    SNode *next;
};
struct SList
{
    Node *head;
    Node *tail;
};

SList dsbh;
```

b. Tạo dslk đơn list_BH chứa dữ liệu đọc từ file txt.




```
void docList(SList &dsbh, char* filename)
{
    int n=0; // số lượng bài hát trong ds
    ifstream in;
    in.open(filename);
    if(in)
    {
```

```
            in>>n;
            for(int i=1; i<=n; i++)
            {
                    Node*p = new SNode;
                    in >> p->info.tenBH;
                    in >> p->info.tenCS;
                    in >> p->info.tenTG;
                    in >> p->info.thoiLuong;

                    addHead(dsbh,p); // thêm bài hát node q vào ds
            }
    }
    in.close();
}
```

II. Bài tập ở lớp

Bài 1. Xây dựng cấu trúc danh sách liên kết đơn lưu các số nguyên và cài đặt thao thác

a. Nhập danh sách liên kết đơn từ file text chưa các số nguyên, các số cách nhau bằng một khoảng trăng (không cho biết số lượng phần tử): VD: 2 3 4 5 12 13

b. Xóa phần tử đầu

c. Xóa phần tử sau một phần tử đã cho

d. Xóa các phần tử có giá trị bằng x

e. Xóa toàn bộ danh sách.

f. Xóa các phần tử chẵn trong danh sách liên kết.

Bài 2. Thông tin bài hát gồm

– Tên bài hát là một chuỗi tối đa 50 ký tự

– Tác giả là một chuỗi tối đa 40 ký tự

– Ca sĩ là một chuỗi tối đa 40 ký tự

– Thời lượng là một số nguyên (đơn vi là giây)

Sử dụng danh sách liên kết đơn lưu trữ và quản lý một playlist nhạc với các chức năng như sau:

a. Đọc danh sách list nhạc từ file định dạng txt

b. In danh sách các bài hát ra màn hình

c. Để nghe hết tất cả bài hát trong playlist phải cần bao nhiêu thời gian.

d. Thêm một bài nhạc mới vào đầu playlist/cuối playlist

e. Xóa một bài nhạc khỏi danh sách

f. Kiểm tra xem bai hat có tên X có trong playlist không?

g. Săp xếp các bài hat trong playlist theo thứ tự từ điển của tên bài hát

h. Sắp xếp các bài hát thứ tự giảm dần của tên ca sĩ

i. Đưa một bài hát trong playlist lên đầu

Bài 3. Hãy định nghĩa một DSLK đơn ***sl*** để quản lý điểm của một lớp học trong một học kỳ,

biết rằng thông tin của mỗi sinh viên trong lớp học gồm có:

- Mã số sinh viên (maSV) – chuỗi tối đa 10 ký tự,
- Họ đệm (hoDem) – chuỗi tối đa 25 ký tự,
- Tên sinh viên (tenSV) – chuỗi tối đa 8 ký tự,
- Năm sinh (namSinh) – số nguyên dương
- Điểm kết quả học tập (diemKQ).

Mỗi học kỳ sinh viên phải học nhiều môn học lý thuyết gồm các thông tin:

- Mã môn học, tên môn học, số tín chỉ
- Điểm tiểu luận (DTL)
- Điểm cuối kỳ (DCK),
- Điểm môn học: được xác định bằng 30% DTL + 70% DCK

Lưu ý: **Điểm kết quả học tập** của sinh viên là trung bình cộng điểm các môn học sinh viên đã đăng ký với trọng số là số tín chỉ.

Hãy viết một chương trình quản lý sinh viên theo mô tả với các chức năng như sau:

a. Cho biết họ tên và điểm kết quả học tập của sinh viên có mã số là **x**.
b. Cho biết các thông tin về sinh viên có tên là **x**.
c. Sắp xếp DSSV theo chiều tăng dần theo MaSV.
d. Sắp xếp DSSV theo chiều tăng dần của tên sinh viên.
e. Thêm một sinh viên sao cho vẫn giữ nguyên thứ tự tăng dần của mã số sinh viên (*có kiểm tra trùng khóa*).
f. Xóa sinh viên có MaSV = **x**.
g. Xóa tất cả các sinh viên có tên là **x**.
h. Tạo danh sách mới từ danh sách đã cho sao cho danh sách mới giảm dần theo điểm kết quả học tập.
i. In danh sách các sinh viên được xếp loại khá (sinh viên xếp loại khá nếu thỏa điều kiện: 7.0 <= điểm kết quả học tập >=8.5)
j. Cho biết sinh viên có điểm kết quả học tập cao nhất.
k. Cho biết sinh viên có điểm kết quả học tập thấp nhất.
l. Cho biết sinh viên có điểm kết quả học tập thấp nhất trong số các sinh viên xếp loại giỏi.
m. Cho biết sinh viên có điểm kết quả học tập gần **x** nhất (***x** là số thực*).

III. Bài tập về nhà

Bài 4. Hãy định nghĩa DSLK đơn ***sl*** để lưu trữ một đa thức (*gồm nhiều đơn thức*) có dạng:

$$\mathbf{p(x) = a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x^1 + a_n}$$

và cài đặt các hàm sau:

a. Sắp xếp đa thức theo chiều giảm dần của bậc (*số mũ*).

b. Rút gọn đa thức (sau khi đã thực hiện câu **a**).

c. Thêm một đơn thức mới vào sao cho vẫn giữ nguyên tính thứ tự của bậc.

d. Xóa một đơn thức khi biết bậc.

e. Tính giá trị của đa thức khi biết giá trị của **x**.

f. Thực hiện việc cộng/trừ/nhận 2 đa thức P và Q để tạo ra một đa thức mới.

Bài 5. Viết chương trình cộng, trừ, nhân, chia 2 số lớn (*Biết mỗi số lớn được lưu trữ bởi 1 DSLK đơn, trong đó mỗi nút của danh sách chỉ chứa một giá trị số* $\in [0, 9]$).

--- HẾT ---