

Bài thực hành:

**“Cấu trúc dữ liệu Danh sách liên kết đơn:
Tổ chức, các thao tác, ứng dụng”**
(Thời lượng: 4 tiết)

- I. Mục tiêu**
- II. Ôn tập**
- III. Luyện tập**
- IV. Bài tập**

I. Mục tiêu:

- Tìm hiểu cách tổ chức kiểu DSLK đơn
- Thực hiện thành thạo các thao tác cơ bản trên DSLK đơn.

II. Ôn tập:

- A. Cách tổ chức kiểu dữ liệu DSLK đơn
- B. Cài đặt các thao tác thường dùng trên DSLK đơn
- C. Tổ chức thư viện DSLK đơn

A. Tổ chức kiểu dữ liệu DSLK đơn:

- DSLK đơn liên kết các phần tử (là các biến động) của DS theo một chiều đi từ phần tử đầu đến phần tử cuối, trong đó phần tử trước chứa thành phần liên kết (là con trỏ) chứa địa chỉ của phần tử kế sau. DSLK đơn được quản lý bằng 2 con trỏ: một chứa địa chỉ của phần tử đầu, một chứa địa chỉ phần tử cuối.

Ta sẽ đặt tên Kiểu DSLK đơn (khi cài đặt) là: **LIST**

- Phần tử (biến động) của DS là một cấu trúc có 2 thành phần: Một thành phần chứa dữ liệu của phần tử, thành phần còn lại là con trỏ, được dùng để chứa địa chỉ của cấu trúc cùng kiểu (Cấu trúc tự trỏ)

Ta sẽ đặt tên Kiểu phần tử (khi cài đặt) là: **NODE**

- Thành phần dữ liệu của phần tử (nút) của DS có kiểu dữ liệu là các kiểu đơn (ký tự, nguyên, thực) hay là các kiểu có cấu trúc tự đặt . . .

Đặt tên Kiểu của thành phần dữ liệu của phần tử trong DS là: **Data**

(Data thay đổi tùy theo các ứng dụng)

Data → NODE → LIST

B. Cài đặt kiểu dữ liệu DSLK đơn:

- Kiểu của thành phần dữ liệu trong DS:
(Giả sử là kiểu int)

typedef int Data;

- Kiểu Phần tử (Nút) của DS:

➤ Định nghĩa ban đầu

```
struct tagNode
{
    Data          Info;
    tagNode*      pNext;
};
```

➤ Đổi lại tên:

typedef tagNode NODE;

- Kiểu DSLK đơn:

struct LIST

{

NODE* pHead; //Con trỏ lưu địa chỉ phần tử đầu DSLK

NODE* pTail; //Con trỏ lưu địa chỉ phần tử cuối DSLK

};

C. Các thao tác thường dùng trên DSLK đơn (trên các nút, trên danh sách)
(Giả sử ta đã cài đặt kiểu DSLK đơn như trên)

1. Tạo nút mới: GetNode (x):

- Chức năng: Tạo ra một phần tử (nút) của DSLK với thành phần dữ liệu là x (Thành phần liên kết của nút mới là con trỏ có giá trị NULL)
- Input: x
- Output: trả về con trỏ - lưu trữ địa chỉ của pt vừa tạo (nếu tạo thành công) - có giá trị NULL (nếu ngược lại).
- Nguyên mẫu của hàm: `NODE* GetNode(Data x);`

2. Khởi tạo DS rỗng: CreatList (l)

- Chức năng: Tạo ra một DSLK l rỗng.
- Input: l
- Output: l
- Nguyên mẫu của hàm: `void CreatList(LIST &l);`

3. Kiểm tra DSLK l có rỗng: IsEmpty(l)

- Chức năng: Kiểm tra DSLK l có rỗng rỗng?
- Input: l
- Output: 1; nếu l rỗng
0; ngược lại
- Nguyên mẫu của hàm: `int IsEmpty(LIST l);`

4. Duyệt danh sách:

- Chức năng: Duyệt danh sách từ đầu DS để xử lý dữ liệu của nút (có thể là xuất dữ liệu ra màn hình, đếm số nút, . . .)
- Input: l
- Nguyên mẫu của hàm: `void ProcessList (LIST l);`

5. Tìm nút có Info là x:

- Chức năng: Tìm trong DS có nút chứa thành phần Info là x?
- Input: l, x
- Output: p (con trỏ p chứa nút đầu tiên có Info là x); nếu có
NULL; ngược lại
- Nguyên mẫu của hàm: `NODE *Search(LIST l, Data x);`

6. Chèn nút (đã có) vào đầu DSLK đơn:

- Chức năng: Chèn một nút (đã có trước) vào đầu DS (Chú ý: thành phần liên kết của nút này là con trỏ có giá trị NULL)
- Input: l, New_Ele
- Output: l (Nút đầu là New_Ele)
- Nguyên mẫu của hàm: `void AddFirst(LIST &l, NODE* new_ele);`

7. Chèn một giá trị dữ liệu vào đầu DSLK đơn:

- Chức năng: Tạo trước nút New_Ele có info là x, con trỏ liên kết có giá trị NULL; sau đó Chèn nút này vào đầu DS.
- Input: l, x (có kiểu data)
- Output: - l (Nút đầu có Info là x kiểu Data)

- Con trỏ chứa địa chỉ của nút vừa tạo.

- Nguyên mẫu của hàm: `NODE* InsertHead(LIST &l, Data x);`

8. Chèn nút (đã có) vào Cuối DSLK đơn:

- Chức năng: Chèn một nút (đã có trước) vào cuối DS
(Chú ý: thành phần liên kết của nút này là con trỏ có giá trị NULL)
- Input: l, New_Ele
- Output: l (Nút cuối là New_Ele)
- Nguyên mẫu của hàm: `void AddTail(LIST &l, NODE *new_ele);`

9. Chèn một giá trị dữ liệu vào cuối DSLK đơn:

- Chức năng: Tạo trước nút New_Ele có info là x, con trỏ liên kết có giá trị NULL; sau đó Chèn nút này vào cuối DS.
- Input: l, x (có kiểu data)
- Output: - l (Nút cuối có Info là x kiểu Data)
- Con trỏ chứa địa chỉ của nút vừa tạo.
- Nguyên mẫu của hàm: `NODE* InsertTail (LIST &l, Data x);`

10. Chèn một nút (chưa có trước) vào sau nút do con trỏ q trỏ tới:

- Chức năng: Tạo trước nút New_Ele có info là x, con trỏ liên kết có giá trị NULL; sau đó Chèn nút này vào sau nút do con trỏ q trỏ tới.
- Input: l, q
- Output: - l (Nút cuối có Info là x kiểu Data)
- Con trỏ chứa địa chỉ của nút vừa tạo.
- Nguyên mẫu của hàm: `void InsertAfter(LIST &l, NODE *q, Data x);`

11. Hủy nút đầu ra khỏi DSLK đơn:

- Chức năng: Hủy nút đầu ra khỏi DSLK đơn
- Input: l
- Output: - l (bớt nút đầu của l input)
- Nguyên mẫu của hàm: `void RemoveHead(LIST &l);`

12. Hủy nút ở vị trí sau nút do con trỏ q trỏ tới:

- Chức năng: Hủy nút sau nút có vị trí do con trỏ q trỏ tới.
- Input: l, q
- Output: - l (bớt 1 nút)
- Nguyên mẫu của hàm: `void RemoveAfter (LIST &l, NODE *q);`

13. Hủy nút có thành phần info là x:

- Chức năng: Hủy nút có info là x.
- Input: l, x
- Output: 1; Nếu có nút
0; Nếu ngược lại
- Nguyên mẫu của hàm: `int RemoveNode(LIST &l, Data x);`

14. Hủy toàn bộ danh sách

- Chức năng: Hủy toàn bộ danh sách
- Input: l
- Output: l rỗng

- Nguyên mẫu của hàm: void RemoveList(LIST &l);

<p>15. Nhập dữ liệu từ tập tin //Tập tin chỉ chứa dữ liệu (số), chẳng hạn: //10 1 0 9 5 8 4 //Chuyển dữ liệu tập tin f vào DSLK đơn l</p> <pre> void File_List(char *f, LIST &l) { ifstream in(f); //Mô de doc if (!in) { cout << "\nLoi mo file !"; exit(1); } CreatList(l); Data x; in >> x; InsertTail(l, x); //InsertHead(l, x) while (!in.eof()) { in >> x; InsertTail(l, x); } in.close(); } </pre>	<p>16. Xuất dữ liệu</p> <pre> void XuatDS(LIST l) { NODE *p; if(IsEmpty(l)) { cout << "\nDS rong!\n"; return; } cout << "\nDu lieu cua DS:\n"; p = l.pHead; while (p != NULL) { cout << (p->Info) << "\t"; p = p->pNext; } } </pre>
---	--

III. Luyện tập

Các bài sau dữ liệu được tổ chức bằng DSLK đơn.

Mỗi bài sau thực hiện theo yêu cầu:

Tạo một Project gồm 2 tập tin:

- Tập tin thư viện *.h: Chứa các hàm chức năng của chương trình
- Tập tin chương trình *.cpp: Chứa hàm main(), các hàm tổ chức menu, nhập xuất dữ liệu.
- Dữ liệu lưu trữ trong một tập tin, chuyển dữ liệu vào danh sách liên kết để xử lý.

Bài 1:

Tổ chức chương trình menu thực hiện trên DSLK đơn với các chức năng:

1. Xem danh sách
2. Tìm nút đầu tiên có dữ liệu là x
3. Tìm nút cuối cùng có dữ liệu là x
4. Chèn giá trị dữ liệu x vào đầu danh sách
5. Chèn giá trị dữ liệu x vào cuối danh sách
6. Hủy nút đầu
7. Hủy nút cuối
8. Hủy các nút có dữ liệu là x
9. Hủy toàn bộ danh sách – khôi phục lại danh sách ban đầu
10. Tính số nút của danh sách
11. Tính số nút của danh sách có giá trị dữ liệu bằng x
12. Tìm giá trị nhỏ nhất của danh sách
13. Đếm số các số nguyên tố trong danh sách

14. Đếm xem giá trị lớn nhất trong danh sách xuất hiện bao nhiêu lần
15. Tính tổng các giá trị trong danh sách
16. Tính tổng các giá trị phân biệt trong danh sách
17. Tính tổng các giá trị chỉ xuất hiện 1 lần
18. Sắp danh sách tang dần theo giá trị của các nút
19. Đảo ngược danh sách l.
20. Tách Danh sách l thành 2 nửa trước sau lưu trữ vào các danh sách l1, l2
21. Tách luân phiên từng nút trong Danh sách l vào 2 danh sách l1, l2.
22. ...

Dữ liệu của các nút là số nguyên, được cho trong tập tin “Dayso.in” sau đây:

10 1 0 9 5 8 4 0 9 1

Bài 2:

Tổ chức chương trình menu thực hiện trên DSLK đơn với các chức năng:

1. Xem danh sách
2. Chèn một sinh viên vào đầu danh sách
3. Chèn một sinh viên vào cuối danh sách
4. Hủy nút đầu
5. Hủy nút cuối
6. Hủy tất cả các sinh viên có tên cho trước
7. Xuất các sinh viên có tên cho trước
8. Xuất các sinh viên có năm sinh cho trước
9. Tìm các sinh viên có năm sinh nhỏ nhất của danh sách
10. Sắp tang danh sách nhân viên theo năm sinh

Dữ liệu của các nút là các cấu trúc chứa thông tin của sinh viên, được cho trong tập tin “Sinhvien.in” sau đây:

Mã SV	Họ	Tên Lót	Tên	Năm sinh
1010123	Hoang	Ngoc	Trung	1996
1012100	Nguyen	Minh	Tuan	1995
1212121	Van	Thai	Lai	1994
1210110	Ly	Thanh	Tuan	1992
1212122	Doan		Trong	1994
1212131	Vo	Thi	Gai	1990
1213110	Le	Van	Vuong	1992
1213122	Do	Thi	Tinh	1995

Ghi chú:

- Về dữ liệu của họ, chữ lót, tên: nếu nhiều hơn 2 từ thì các từ kết nối nhau bằng dấu “_”. Nếu chữ lót không có thì điền dấu “_”

IV. Bài tập:

Bài 1:

Giả sử một công ty quản lý nhân viên theo các thông tin sau:

- Mã nhân viên: chuỗi 8 ký tự

- Họ của nhân viên: chuỗi 8 ký tự
- Tên lót nhân viên: chuỗi 8 ký tự
- Tên nhân viên: chuỗi 8 ký tự
- Năm sinh: số nguyên
- Trình độ văn hoá: số nguyên từ 1 đến 6
(1 = cấp 1 ; 2 = cấp 2 ; 3 = cấp 3 ; 4 = đại học; 5 = cao học, 6 = Tiến sĩ)
- Lương: số nguyên $\leq 5\,000\,000$

Viết chương trình tùy chọn thực hiện các chức năng sau:

- Tính tổng lương mà công ty trả cho nhân viên
 - Tìm các nhân viên có lương cao nhất
 - Liệt kê danh sách các nhân viên có năm sinh cho trước.
 - Sắp danh sách tăng dần theo mức lương
 - Sắp danh sách tang dần theo tên, họ, chu lot.
- Dữ liệu về nhân viên của công ty được cho trong tập tin “nhanvien.txt”:
(Gồm các thông tin: Mã nhân viên, họ, chữ lót, tên, năm sinh, trình độ, lương cơ bản)

123456	Tran	Thi	Thuy_Hoa	1980	6	4567893
123345	Nguyen	Ngoc	Minh	1985	3	2345678
113245	Hoang	Hoa	Tien	1990	2	1436784
142365	Vuong	Anh	Tuan	1966	1	1500000
154876	Vo_Tran	Ngoc	Duy	1979	5	4445670
165234	Truong	Minh	Tung	1990	2	2000000
123215	Nguyen	Minh	Nhat	1979	4	3122130
123226	Nguyen	Thi_Thuy	Tien	1980	3	3132145
114245	Hoang	_	Tuan	1966	6	4567893
160234	Tran	Minh	Tien	1980	1	2000000
123220	Nguyen	Minh	Thu	1970	6	3122130
123124	Nguyen	Thi	Thanh	1980	1	1500000

Ghi chú:

- Về dữ liệu của họ, chữ lót, tên: nếu nhiều hơn 2 từ thì các từ kết nối nhau bằng dấu “_”. Nếu chữ lót không có thì điền dấu “_”
- Dữ liệu tổ chức bằng kiểu danh sách liên kết đơn

Bài 2:

Điểm số các môn học của sinh viên trong học kỳ được cho trong tập tin “Diemso.in” sau đây (gồm 4 cột):

Mã SV	Điểm M1	Điểm M2	Điểm M3
1010123	7.5	8	5
1012100	5	6.5	7
1212121	3	7.5	8.5
1012121	3.5	9.5	5
1111100	5	8	7.5
1212021	10	9	8.5
1010121	8.5	2.5	10

Số tín chỉ của mỗi môn học được cho trong bảng sau:

Môn 1 4
 Môn 2 4
 Môn 3 2

Điểm trung bình học kỳ của sinh viên tính theo công thức:

$$DTB_Chung = \frac{\sum_i (So_TC_Mon_i * Diem_So - Mon_i)}{\sum_i So_TC_Mon_i}$$

Tính điểm trung bình học kỳ của các sinh viên, xuất ra dưới dạng:

Mã SV	Điểm M1	Điểm M2	Điểm M3	Điểm TBHK
-------	---------	---------	---------	-----------