

Bài tập thực hành tuần 7

Bài 1 – Danh sách liên kết

Cho danh sách liên kết với cấu trúc node khai báo như sau

```
// Declaration LList
// Declaration Node of LList
struct Node {
    int data;
    Node* Next;
};

// Declaration LList
//Node*& LList;
```

Yêu cầu

1. Viết hàm thêm một phần tử vào đầu danh sách
`void addFirst(Node*& head, int key)`
 2. Viết hàm thêm vào cuối danh sách
`void addLast(Node*& head, int key)`
 3. Viết hàm tạo một danh sách liên kết
`void LList_input(Node*& head)`
 4. Viết hàm tìm node đứng trước một node
`Node* findPreNode(Node*& head, int key)`
 5. Viết hàm thêm một node vào giữa danh sách
`void insertMidList(Node*& head, int key, int newKey)`
 6. Viết hàm xóa một node khỏi danh sách liên kết
 7. Viết hàm xóa một node khỏi danh sách liên kết mà không quan tâm node đó là node đầu, giữa hay cuối danh sách liên kết
- HD: Sử dụng con trỏ kép – double pointer.

Bài 2: Cho thông tin sinh viên gồm: Mã số sinh viên, họ và tên, ngày sinh, điểm, được khai báo như sau:

```
/* Ngày sinh */
struct NgaySinh {
    int ngay;
    int thang;
    int nam;
};

/* Khai bao Sinh Vien */
struct SinhVien {
    char maso[9];
    char hoten[40];
    NGAYSINH ngsinh;
    float diem;
};
```

Yêu cầu:

1. Sử dụng danh sách liên kết để cài đặt danh sách sinh viên.
2. Viết hàm đếm số node của danh sách.
3. Viết hàm thêm một sinh viên có mã số sinh viên đã cho vào danh sách.
4. Viết hàm tính điểm trung bình của danh sách.
5. Viết hàm xếp loại sinh viên theo điểm.
6. Viết hàm lưu các sinh viên có điểm nhỏ hơn điểm trung bình vào một mảng động.
7. Viết hàm xóa một sinh viên có mã số sinh viên đã cho khỏi danh sách.
8. Viết hàm sắp xếp danh sách tăng dần theo mã số sinh viên.
9. Viết hàm thống kê các sinh viên có cùng ngày sinh nhật.

