

# NỘI DUNG



## DANH SÁCH LIÊN KẾT ĐƠN (LIST)

# Hủy phần tử trong DSLK đơn

- **Nguyên tắc:** Phải cô lập phần tử cần hủy trước hủy.
- Các vị trí cần hủy
  - Hủy phần tử đứng đầu List
  - Hủy phần tử có khoá bằng x
  - Hủy phần tử đứng sau q trong danh sách liên kết đơn
- Ở phần trên, các phần tử trong DSLK đơn được cấp phát vùng nhớ động bằng hàm new, thì sẽ được giải phóng vùng nhớ bằng hàm delete.



# Thuật toán hủy phần tử đầu trong DSLK

## ➤ Bắt đầu:

- Nếu ( $pHead \neq NULL$ ) thì
- B1:  $p = pHead$
- B2:
  - +  $pHead = pHead \rightarrow pNext$
  - + delete ( $p$ )
- B3:
  - Nếu  $pHead == NULL$  thì  $pTail = NULL$



# Cài đặt thuật toán

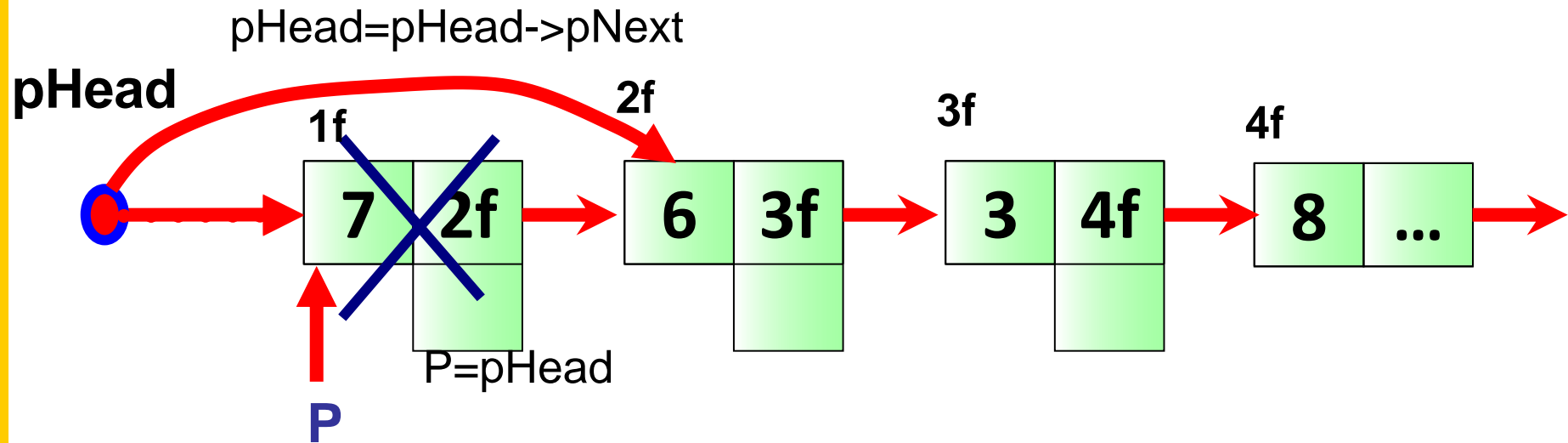
- Hủy phần tử đầu tiên của List

```
void RemoveHead(List &l)
{
    if(l.pHead!=NULL)
    {
        Node *p =l.pHead;
        l.pHead=l.pHead->pNext;
        p->next=NULL; //nên cắt liên kết giữa p và List
        delete p;
        if(l.pHead==NULL) //List có 1 nút duy nhất, sau khi xóa
            l.pTail=NULL;
    }
}
```

List trở thành rỗng



# Minh họa thuật toán



# Hủy phần tử sau phần tử q trong List

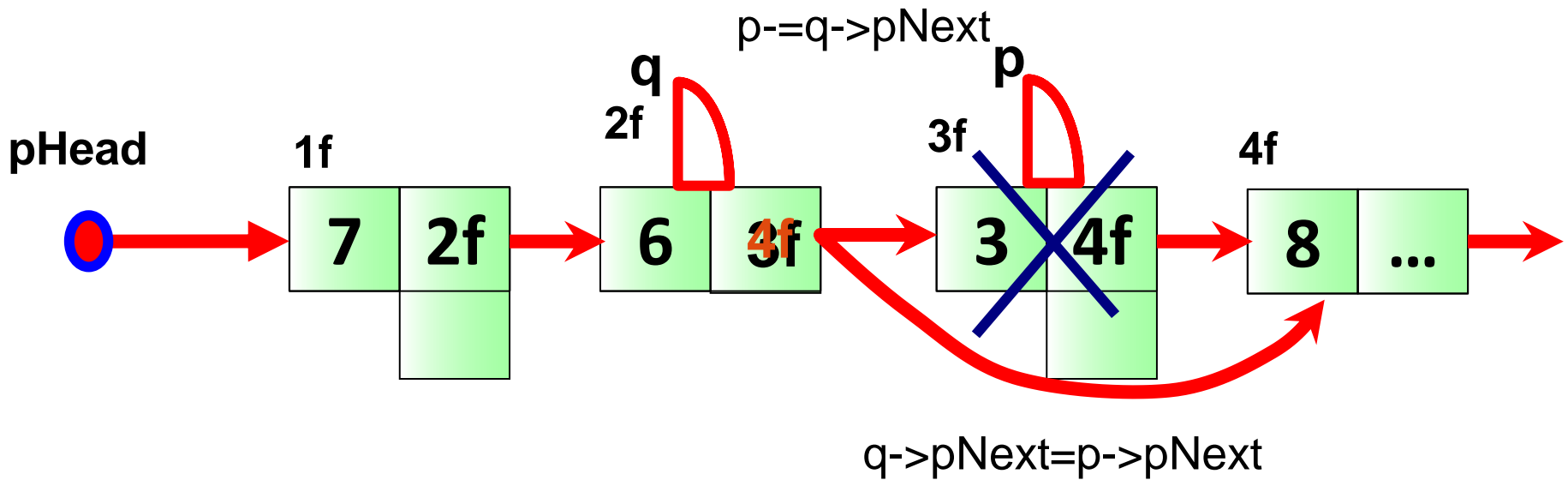
## ➤ Bắt đầu

Nếu ( $q \neq \text{NULL}$ ) thì //q tồn tại trong List

- B1:  $p = q \rightarrow pNext$ ; // p là phần tử cần hủy
  - B2: Nếu ( $p \neq \text{NULL}$ ) thì // q không phải là phần tử cuối
    - +  $q \rightarrow pNext = p \rightarrow pNext$ ;
    - + nếu ( $p == pTail$ ) // nút cần hủy là nút cuối, cập nhật nút cuối mới
- $pTail = q$ ;
- + delete p; // hủy p



# Minh họa thuật toán



# Thuật toán hủy phần tử có khoá x

## Bước 1:

Tìm phần tử p có khoá bằng x, và q đứng trước p

## Bước 2:

Nếu ( $p \neq \text{NULL}$ ) thì //tìm thấy phần tử có khoá bằng x

Hủy p ra khỏi List bằng cách hủy phần tử  
đứng sau q

Ngược lại

Báo không tìm thấy phần tử có khoá





# Hủy danh sách liên kết đơn

- Bước 1:

Trong khi (danh sách chưa hết) thực hiện

- B1.1:

$p = pHead;$

$pHead = pHead \rightarrow pNext;$  // cập nhật pHead

- B1.2:

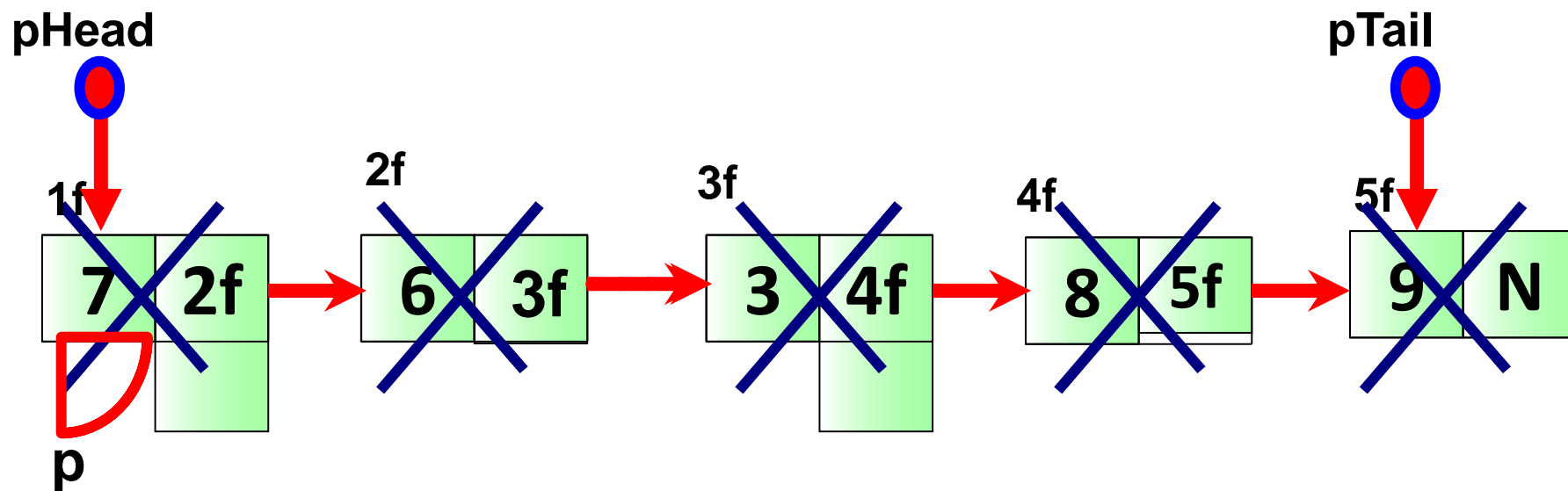
Hủy p

- Bước 2:

$pTail = NULL;$  // bảo toàn tính nhất quán khi List rỗng



# Minh họa thuật toán



# Vài ứng dụng danh sách liên kết đơn

- Dùng DSLK đơn để lưu trữ danh sách các học viên trong lớp học
- Dùng DSLK đơn để quản lý danh sách nhân viên trong một công ty, trong cơ quan
- Dùng DSLK đơn để quản lý danh sách các cuốn sách trong thư viện
- Dùng DSLK đơn để quản lý các băng đĩa trong tiệm cho thuê đĩa.
- ..VV



# Dùng DSLK đơn để quản lý lớp học

- Yêu cầu: Thông tin của một sinh viên gồm, mã số sinh viên, tên sinh viên, điểm trung bình.
1. Hãy khai báo cấu trúc dữ liệu dạng danh sách liên kết để lưu danh sách sinh viên nói trên.
  2. Nhập danh sách các sinh viên, và thêm từng sinh viên vào đầu danh sách (việc nhập kết thúc khi tên của một sinh viên bằng rỗng)
  3. Tìm một sinh viên có trong lớp học hay không
  4. Xoá một sinh viên có mã số bằng x (x nhập từ bàn phím)
  5. Liệt kê thông tin của các sinh viên có điểm trung bình lớn hơn hay bằng 5.



# Dùng DSLK đơn để quản lý lớp học

6. Xếp loại và in ra thông tin của từng sinh viên, biết rằng cách xếp loại như sau:  
ĐTB  $\leq 3.6$  : Loại yếu  
ĐTB  $\geq 5.0$  và ĐTB  $< 6.5$  : Loại trung bình  
ĐTB  $\geq 6.5$  và ĐTB  $< 7.0$ : Loại trung bình khá  
ĐTB  $\geq 7.0$  và ĐTB  $< 8.0$ : Loại khá  
ĐTB  $\geq 8.0$  và ĐTB  $< 9.0$ : Loại giỏi.  
ĐTB  $\geq 9.0$  : Loại xuất sắc
7. Sắp xếp và in ra danh sách sinh viên tăng theo điểm trung bình.
8. Chèn một sinh viên vào danh sách sinh viên tăng theo điểm trung bình nói trên, sao cho sau khi chèn danh sách sinh viên vẫn tăng theo điểm trung bình

..VV



# Cấu trúc dữ liệu cho bài toán

- Cấu trúc dữ liệu của một sinh viên  
struct SV

```
{ char tên[40];  
  char Maso[40];  
  float ĐTB;  
}
```

- Cấu trúc dữ liệu của 1 nút trong xâu  
struct Node

```
{ SV Info;  
  Node *pNext;  
};
```



# Câu hỏi và Bài tập

1. Nêu các bước để thêm một nút vào đầu, giữa và cuối danh sách liên kết đơn.
2. Nêu các bước để xóa một nút ở đầu, giữa và cuối danh sách liên kết đơn.
3. Viết thủ tục để in ra tất cả các phần tử của 1 danh sách liên kết đơn.
4. Viết chương trình thực hiện việc sắp xếp 1 danh sách liên kết đơn bao gồm các phần tử là số nguyên.
5. Viết chương trình cộng 2 đa thức được biểu diễn thông qua danh sách liên kết đơn.

