

BÀI 4

DANH SÁCH TUYỂN TÍNH

Nội dung

1. Khái niệm danh sách tuyến tính
2. Các phép toán với danh sách
3. Lưu trữ kế tiếp của danh sách tuyến tính
4. Danh sách móc nối đơn
5. Danh sách nối đôi
6. Danh sách móc nối vòng
7. Ngăn xếp
8. Hàng đợi

4.1. Khái niệm danh sách tuyến tính

4.1.1. Danh sách là gì?

- Danh sách là một tập các phần tử thuộc cùng một lớp đối tượng nào đó.
- Ví dụ:
 - Dãy số nguyên: $D = \{35, 4, -5, \dots, 72\}$ hay
 - Danh sách sinh viên:

| STT | Mã SV | Họ đệm | Tên | Giới tính | Năm sinh | Điểm TK |
|-----|-------|------------|-------|-----------|----------|---------|
| 1 | 1001 | Tran Van | Thanh | Nam | 1997 | 7.5 |
| 2 | 1002 | Nguyen Thi | Bich | Nu | 1998 | 7.2 |
| 3 | 1003 | Nguyen Van | Giang | Nam | 1996 | 6.4 |
| 4 | 1004 | Bui Thi | Hong | Nu | 1998 | 8.6 |
| 5 | 1005 | Duong Van | Hung | Nam | 1997 | 6.8 |

4.1.2. Khái niệm danh sách tuyến tính

- Giả sử L là một danh sách có n phần tử

Ký hiệu $L = \{e_1, e_2, \dots, e_n\}$

- Khi đó:
 - n gọi là độ dài của danh sách L
 - $n > 0$ thì e_1 là phần tử đầu tiên, e_n là phần tử cuối cùng
 - Với danh sách L , ta nói phần tử e_i đứng trước phần tử e_{i+1} và đứng sau phần tử e_{i-1} ($i = 1, 2, \dots, n$).
 - Danh sách mà các phần tử có thứ tự “*trước-sau*” gọi là “*danh sách tuyến tính*”.

4.2. Các phép toán trên danh sách tuyến tính

- Khởi tạo danh sách rỗng (*creat*)
- Kiểm tra danh sách rỗng (*empty*)
- Kiểm tra danh sách đầy (*full*)
- Bổ sung một phần tử vào danh sách (*add*)
- Loại bỏ một phần tử khỏi danh sách (*remove*)
- Sắp xếp danh sách (*sort*)
- Tìm kiếm trên danh sách (*search*)
- ...

4.3. Lưu trữ kế tiếp của danh sách tuyến tính

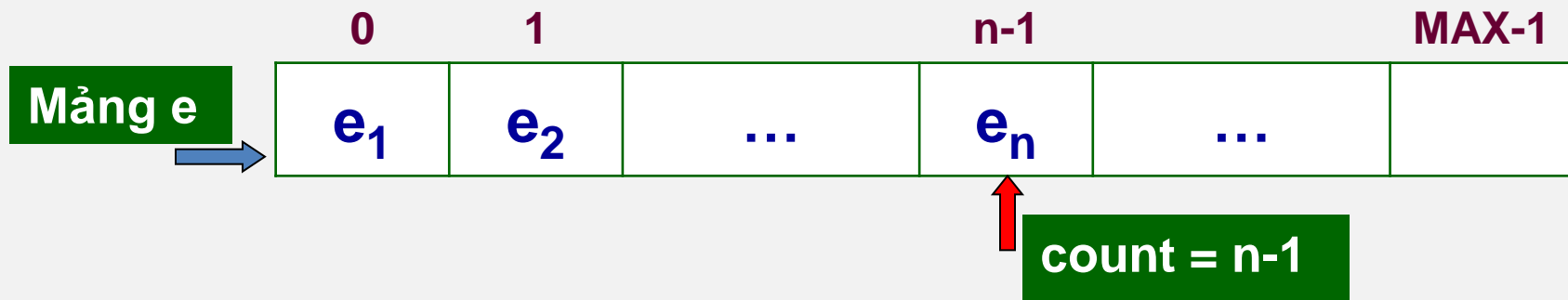
- Danh sách được lưu trữ trong bộ nhớ bởi một mảng một chiều gọi là lưu trữ kế tiếp.
- Mỗi phần tử của mảng lưu trữ một phần tử của danh sách.
- Ưu điểm:
 - Truy cập nhanh và đồng đều đối với mọi phần tử
 - Các thao tác được thực hiện khá đơn giản.
- Nhược điểm:
 - Do kích thước mảng cố định khi khai báo nên có thể dẫn đến sự lãng phí hoặc thiếu bộ nhớ.

4.3.1. Biểu diễn cấu trúc dữ liệu

- Giả sử các phần tử của danh sách $L = \{e_1, e_2, \dots, e_n\}$ có kiểu dữ liệu là “*DataType*” (tức là e_i có kiểu *DataType*)
- Danh sách có thể phát triển đến độ dài tối đa là N (N là số nguyên dương).
- Khi đó: danh sách được biểu diễn bởi một cấu trúc gồm hai thành phần:
 - Thành phần thứ nhất: mảng một chiều “ e ” lưu các phần tử của danh sách L , $e[i-1]$ lưu e_i
 - Thành phần thứ hai: biến “*count*” lưu chỉ số phần tử mảng, lưu trữ phần tử cuối cùng của danh sách.

Biểu diễn cấu trúc dữ liệu (tt)

Danh sách cần lưu trữ: $L = \{ e_1, e_2, \dots, e_n \}$



Mảng lưu trữ danh sách

Biểu diễn cấu trúc dữ liệu (tt)

- Cấu trúc dữ liệu được khai báo như sau:

```
#define MAX N
struct DataType{
    //Các thành phần dữ liệu;
};
struct List{
    int count;
    DataType e[MAX];
};
List L; //Khai báo ds L
```

`L.count = -1` -> danh sách L rỗng

`L.count = MAX - 1` -> danh sách L đầy

Biểu diễn cấu trúc dữ liệu (tt) – Ví dụ

- Cho danh sách sinh viên

| STT | Họ và tên | Tuổi | Điểm TK |
|-----|-------------|------|---------|
| 1 | Vũ Lan Anh | 22 | 7.8 |
| 2 | Hà Anh Lan | 23 | 7.7 |
| 3 | Tạ Bạch Lan | 23 | 8.7 |
| 4 | Vũ Hoa Lan | 21 | 6.7 |
| 5 | Bùi Như Lan | 23 | 8.7 |

- Danh sách được lưu trữ trong bộ nhớ máy tính dưới dạng danh sách lưu trữ kế tiếp.

Biểu diễn cấu trúc dữ liệu (tt) – Ví dụ

- Danh sách sinh viên được lưu trữ kế tiếp

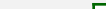
| | 0 | 1 | 2 | 3 | 4 | 5 |
|-----------|------------|------------|-------------|------------|-------------|---|
| Mảng e | Vũ Lan Anh | Hà Anh Lan | Tạ Bạch Lan | Vũ Hoa Lan | Bùi Như lan | |
| | 22 | 23 | 23 | 21 | 23 | |
| | 7.8 | 7.7 | 8.7 | 6.7 | 8.7 | |

count = 4

- Hoặc ký hiệu danh sách sinh viên là:

$S = \{sv01, sv02, sv03, sv04, sv05\}$

- và mô tả như sau

| | | | | | | |
|---|-------------|-------------|-------------|-------------|-------------|----------|
| Mảng e  | 0 | 1 | 2 | 3 | 4 | 5 |
| | sv01 | sv02 | sv03 | sv04 | sv05 | |

count = 4

Biểu diễn cấu trúc dữ liệu (tt) – Ví dụ

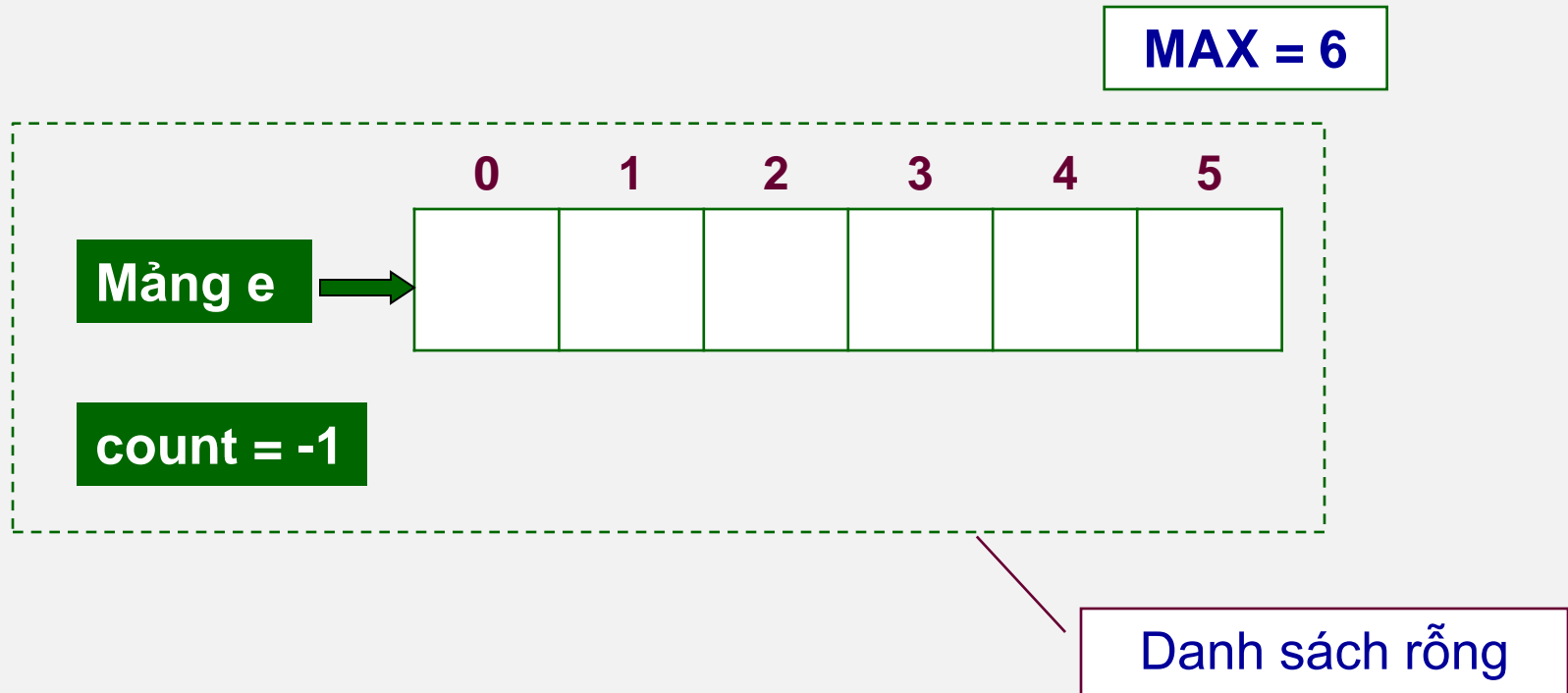
- Cấu trúc dữ liệu danh sách tuyến tính lưu trữ thông tin về sinh viên:

```
#define MAX 6
struct SinhVien{
    char hoTen[30];
    char gioiTinh[4];
    int tuoi;
};
struct List{
    int count;
    SinhVien e[MAX];
};
List L; //Khai báo ds L
```

4.3.2. Các phép toán trên danh sách kế tiếp

- Khởi tạo danh sách rỗng
- Kiểm tra danh sách rỗng
- Kiểm tra danh sách đầy
- Phép loại bỏ một phần tử khỏi danh sách
- Bổ sung một phần tử vào danh sách
- Thống kê danh sách
- Tính toán trên danh sách
- Tìm kiếm trên danh sách
- Sắp xếp danh sách

a. Khởi tạo danh sách rỗng



```
void creat(List &L) {  
    L.count = -1;  
}
```

b. Kiểm tra danh sách rỗng

MAX = 6

Mảng e

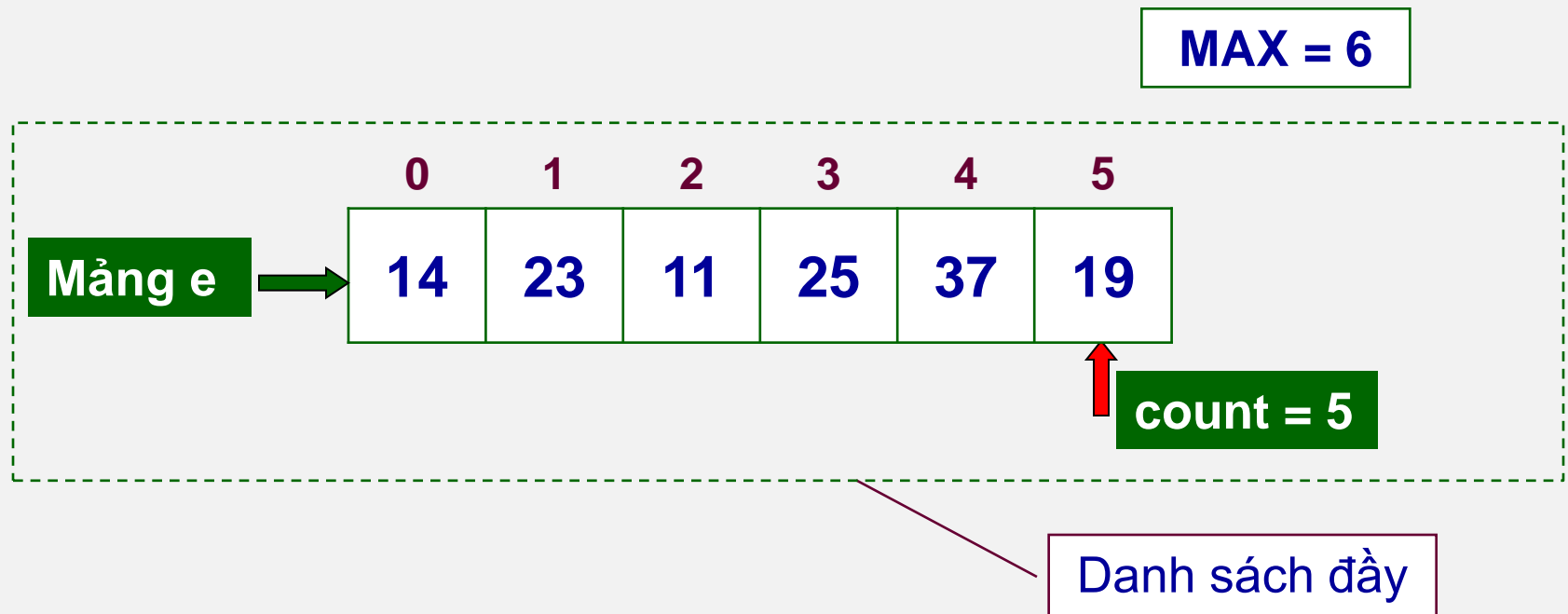


count = -1

```
int empty(List L) {  
    return (L.count == -1) ;  
}
```

- Hàm **empty** trả về giá trị **1** nếu danh sách rỗng, ngược lại trả về **0**.

c. Kiểm tra danh sách đầy

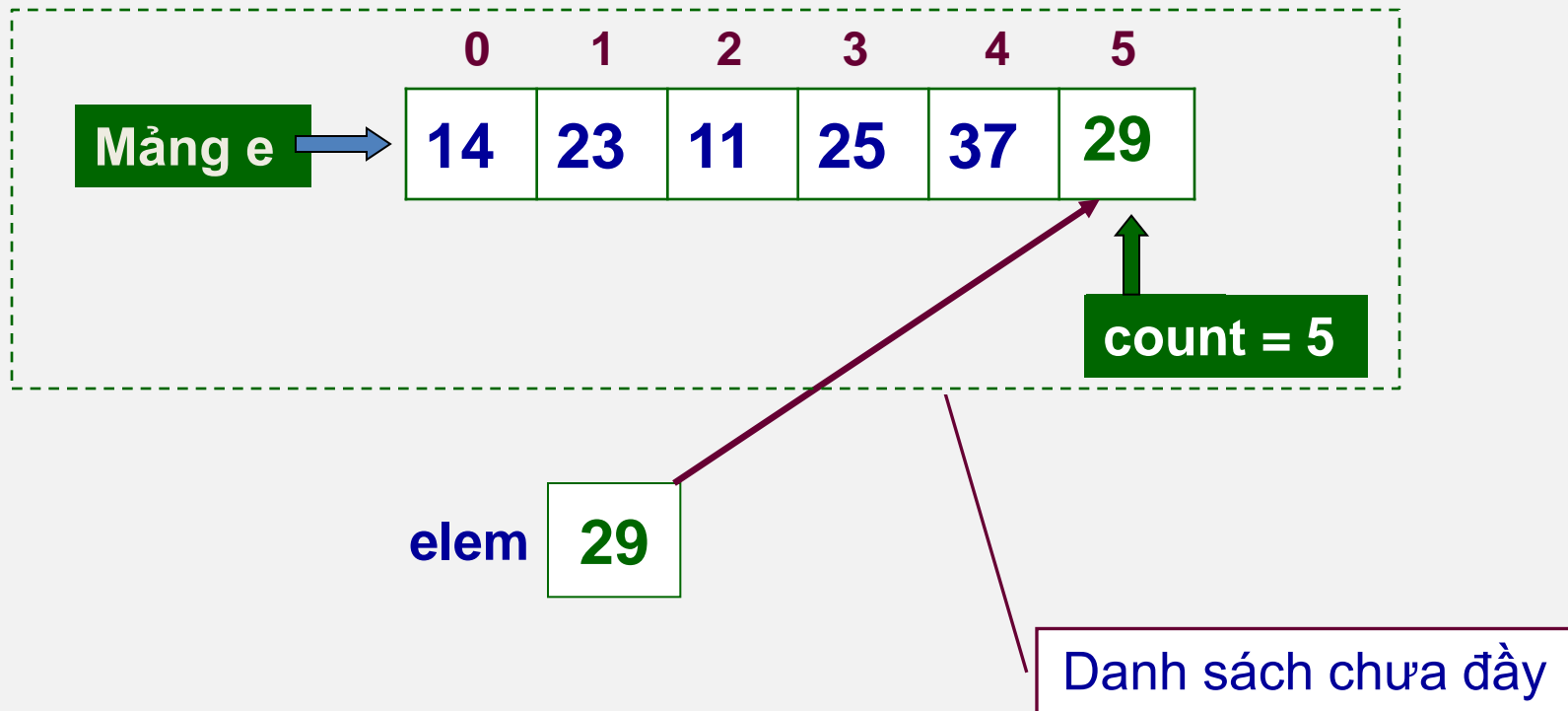


```
int full(List L){  
    return (L.count == MAX-1);  
}
```


d. Thêm một phần tử vào cuối danh sách

- Điều kiện: Danh sách chưa đầy

MAX = 6



Thêm một phần tử vào cuối danh sách (tt)

```
int add(List &L, DataType elem)
{
    if (full(L))
        return 0;
    else{
        L.count++;
        L.e[L.count] = elem;
        return 1;
    }
}
```

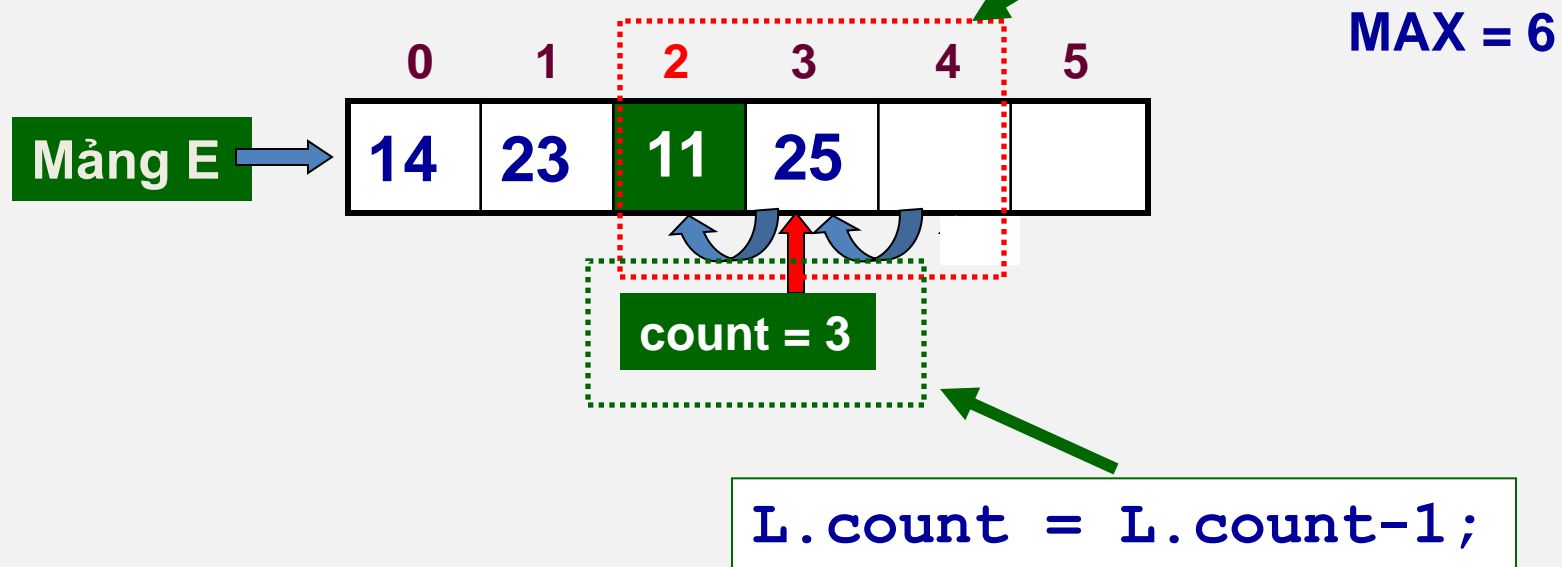
Ví dụ 1

- Cài đặt chương trình quản lý sinh viên:
 - Định nghĩa cấu trúc dữ liệu
 - Các hàm (phép toán) cơ bản
 - Hàm thêm phần tử vào cuối danh sách
 - Hàm khởi tạo dữ liệu
 - Hàm tạo danh sách
 - Hàm hiển thị danh sách
 - Hàm main()

e. Loại bỏ một phần tử khỏi danh sách

- Loại bỏ phần tử thứ k khỏi danh sách L
- Giả sử $k = 3$

```
for (i=k; i<=L.count; i++)  
    L.E[i-1] = L.E[i];
```



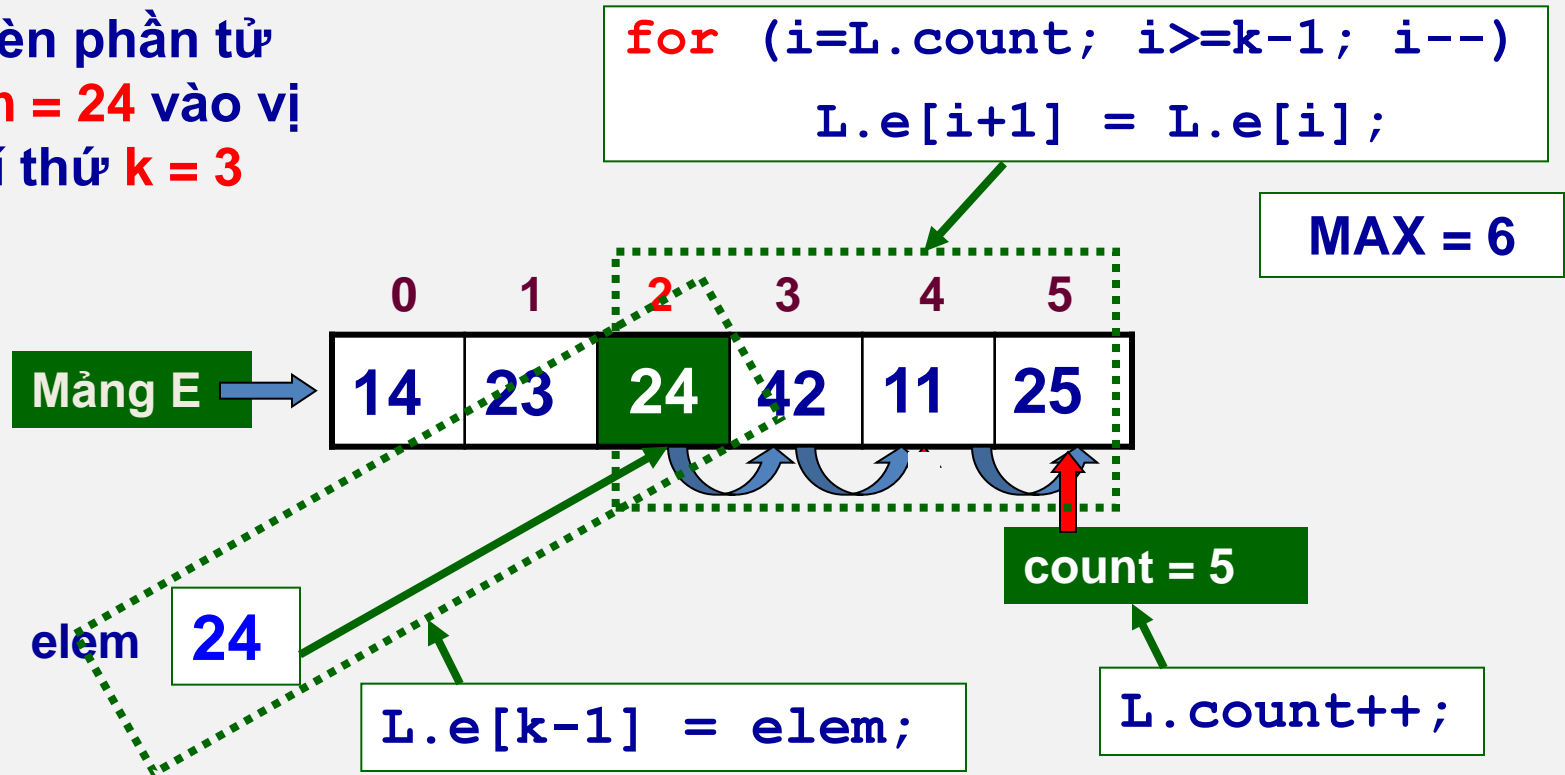
Loại bỏ một phần tử khỏi danh sách (tt)

```
int remove(int k, List &L)
{
    if (k <= L.count+1 && k>0)
    {
        for (int i = k; i <= L.count; i++)
            L.e[i-1] = L.e[i] ;
        L.count = L.count - 1;
        return 1;
    }
    else return 0;
}
```

f. Chèn một phần tử vào danh sách

- Chèn một phần tử vào vị trí k trong danh sách L

Chèn phần tử
elem = 24 vào vị
trí thứ **k = 3**



Chèn một phần tử vào danh sách (tt)

```
int insert(List &L, int k, DataType elem)
{
    if (k <= L.count+1 && k>0 && !full(L))
    {
        for (int i = L.count; i>=k-1; i--)
            L.e[i+1] = L.e[i] ;
        L.e[k-1] = elem;
        L.count ++;
        return 1;
    }
    else return 0;
}
```

Bài tập 1

- Cho danh sách sinh viên như bảng dưới đây

| STT | Mã SV | Họ đệm | Tên | Giới tính | Năm sinh | Điểm TK |
|-----|-------|------------|-------|-----------|----------|---------|
| 1 | 1001 | Tran Van | Thanh | Nam | 1997 | 7.5 |
| 2 | 1002 | Nguyen Thi | Huong | Nu | 1998 | 7.2 |
| 3 | 1003 | Nguyen Van | Binh | Nam | 1996 | 6.4 |
| 4 | 1004 | Bui Thi | Hong | Nu | 1998 | 8.6 |
| 5 | 1005 | Duong Van | Giang | Nam | 1997 | 6.8 |

Bài tập 1 (tt)

- Giả sử danh sách được lưu trữ kế tiếp trong bộ nhớ máy tính.
- Thực hiện các yêu cầu sau với danh sách:
 - Mô tả cấu trúc dữ liệu của danh sách qua hình vẽ.
 - Khai báo cấu trúc dữ liệu của danh sách.
 - Mô tả thao tác xóa phần tử đầu tiên trong danh sách bằng hình vẽ.
 - Cài đặt hàm xóa phần tử đầu tiên trong danh sách.
 - Mô tả thao tác chèn sinh viên (1006, Le Thi, Doan, Nu, 1998, 7.6) vào vị trí thứ 3 trong danh sách.
 - Mô tả thao tác sắp xếp danh sách theo chiều tăng dần của tên sinh viên bằng phương pháp lựa chọn.
 - Cài đặt chương trình mô phỏng các thao tác trên.

Bài tập 2

- Cho danh sách hàng hóa như bảng dưới đây:

| STT | Mã hàng | Tên hàng | ĐV tính | Đơn giá | Số lượng | Thành tiền |
|-----|---------|----------|---------|---------|----------|------------|
| 1 | 2001 | Vở | Quyển | 5000 | 20 | 100000 |
| 2 | 2002 | Bút chì | Cái | 8000 | 50 | 400000 |
| 3 | 2003 | Hộp bút | Chiếc | 30000 | 10 | 300000 |
| 4 | 2004 | Tẩy | Cái | 10000 | 20 | 200000 |
| 5 | 2005 | Mực | Lọ | 12000 | 5 | 60000 |
| 6 | 2006 | Thước kẻ | Chiếc | 3000 | 15 | 45000 |

Bài tập 2 (tt)

- Giả sử danh sách được lưu trữ kế tiếp trong bộ nhớ máy tính.
- Thực hiện các yêu cầu sau với danh sách: :
 - Mô tả cấu trúc dữ liệu của danh sách qua hình vẽ.
 - Khai báo cấu trúc dữ liệu của danh sách.
 - Mô tả thao tác xóa phần tử thứ 3 trong danh sách bằng hình vẽ.
 - Cài đặt hàm xóa phần tử thứ 3 trong danh sách.
 - Mô tả thao tác chèn hàng hóa (2007, Phấn, Hộp, 3000, 15, 45000) vào vị trí đầu tiên trong danh sách.
 - Mô tả thao tác sắp xếp danh sách theo chiều giảm dần của thành tiền bằng phương pháp nổi bọt.
 - Cài đặt chương trình mô phỏng các thao tác trên.

Bài tập

- 3.** Cho danh sách lưu trữ kế tiếp thông tin về các cán bộ, mỗi cán bộ gồm: Mã cán bộ, họ đệm, tên, giới tính, tuổi, hệ số lương, lương (= hệ số lương * 1150000). Hãy cài đặt chương trình thực hiện các yêu cầu sau:
- Khai báo cấu trúc dữ liệu của danh sách.
 - Nhập mới n phần tử cho danh sách.
 - Hiển thị danh sách những cán bộ nữ có tuổi từ 50 trở xuống ra màn hình.
 - Cho biết danh sách có cán bộ nào tên là “Tung” không? Nếu có hãy hiển thị thông tin đầy đủ về cán bộ này (nếu có nhiều cán bộ tên “Tung” thì chỉ hiển thị cán bộ đầu tiên trong danh sách).
 - Thêm một phần tử vào sau phần tử thứ 3 trong danh sách, hiển thị danh sách.

Bài tập

4. Cho DSTT lưu trữ kế tiếp thông tin về các sinh viên, gồm: Mã sinh viên, họ đệm, tên, năm sinh, điểm tổng kết. Hãy cài đặt chương trình thực hiện các yêu cầu sau:

- Khai báo cấu trúc dữ liệu của danh sách.
- Nhập (tạo) mới 5 phần tử cho danh sách.
- Hiển thị danh sách lên màn hình.
- Xóa sv có mã 1004 trong danh sách (nếu có), hiển thị danh sách nếu có xóa.
- Tìm sinh viên có tên “Doanh” trong danh sách, hiển thị kết quả tìm kiếm (nếu có).
- Sắp xếp danh sách theo chiều giảm dần của điểm tổng kết, hiển thị lại danh sách.

Thank you...!