



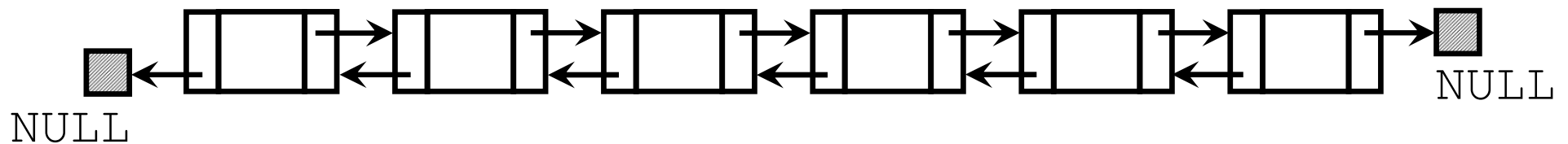
DANH SÁCH LIÊN KẾT KÉP

1. Hồ Thái Ngọc
2. ThS. Võ Duy Nguyên
3. TS. Nguyễn Tấn Trần Minh Khang



HÌNH ẢNH DANH SÁCH LIÊN KẾT KÉP

Hình ảnh danh sách liên kết kép



Hình ảnh danh sách
liên kết kép



Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

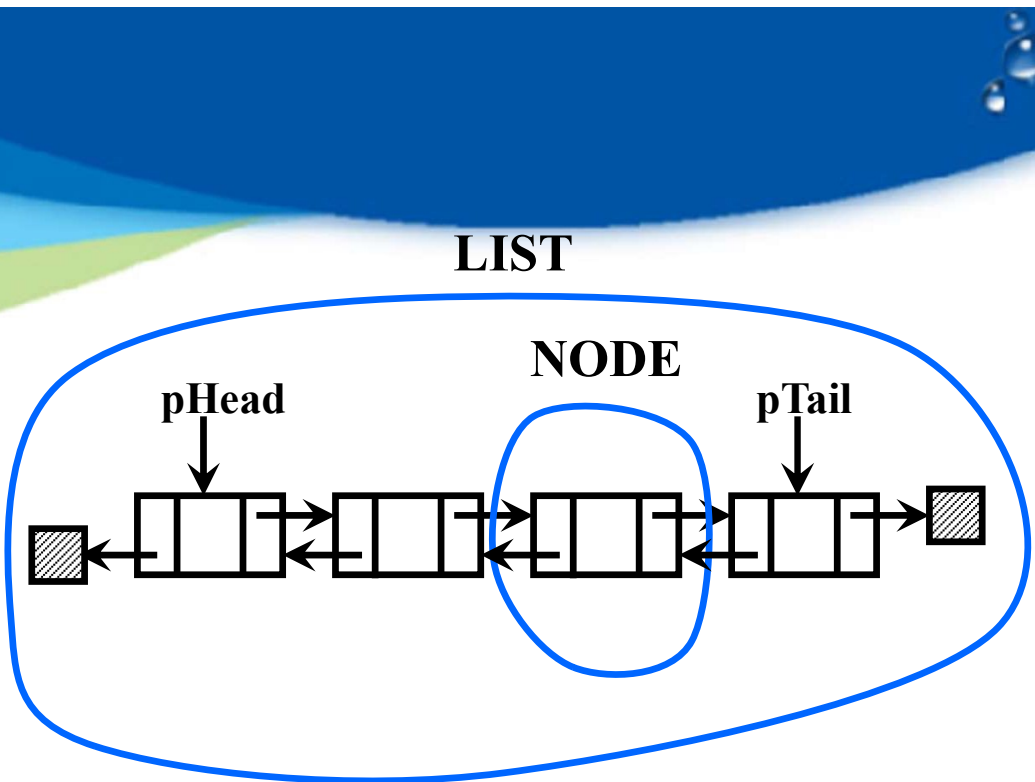
Hồ Thái Ngọc

ThS. Võ Duy Nguyên

TS. Nguyễn Tấn Trần Minh Khang



CẤU TRÚC DỮ LIỆU DSLK KÉP

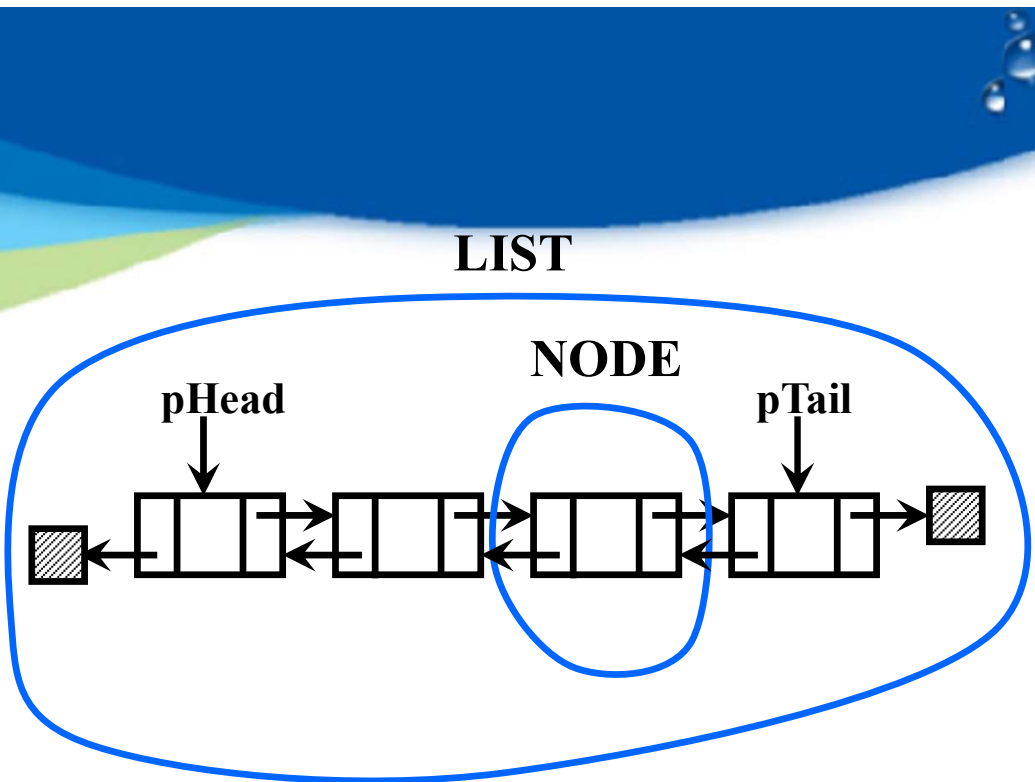


Cấu trúc dữ liệu danh sách liên kết kép

```

11.struct node
12.{
13.|    KDL info;
14.|    struct node* pNext;
15.|    struct node* pPrev;
16.};
17.typedef struct node NODE;
18.struct list
19.{
20.|    NODE* pHead;
21.|    NODE* pTail;
22.};
23.typedef struct list LIST;

```



CTDL dslk kép các
số nguyên

```

11.struct node
12.{
13.|    int info;
14.|    struct node* pNext;
15.|    struct node* pPrev;
16.};
17.typedef struct node NODE;
18.struct list
19.{
20.|    NODE* pHead;
21.|    NODE* pTail;
22.};
23.typedef struct list LIST;

```



Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

Hồ Thái Ngọc

ThS. Võ Duy Nguyên

TS. Nguyễn Tấn Trần Minh Khang



KHỞI TẠO DANH SÁCH LIÊN KẾT KÉP

Khởi tạo danh sách liên kết kép



- Khái niệm: Khởi tạo danh sách liên kết kép là tạo ra danh sách rỗng không chứa node nào hết.
- Định nghĩa hàm khởi tạo danh sách liên kết kép.

```
11. void Init(LIST& l)
12. {
13. |     l.pHead = NULL;
14. |     l.pTail = NULL;
15. }
```

Khởi tạo danh sách
liên kết kép



KIỂM TRA DSLK KÉP RỖNG

Kiểm tra dslk kép rỗng



— Khái niệm: Kiểm tra danh sách liên kết kép rỗng là hàm trả về giá trị 1 khi danh sách rỗng. Trong tình huống danh sách không rỗng thì hàm sẽ trả về giá trị 0.

```
11. int IsEmpty(LIST l)
12. {
13.     if(l.pHead==NULL)
14.         return 1;
15.     return 0;
16. }
```

Kiểm tra danh sách
liên kết kép rỗng



Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

Hồ Thái Ngọc

ThS. Võ Duy Nguyên

TS. Nguyễn Tấn Trần Minh Khang



TẠO NODE CHO DSLK KÉP

Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu NODE để chứa thông tin đã được biết trước.

Tạo node cho danh
sách liên kết kép

Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu **NODE** để chứa thông tin đã được biết trước.

Tạo node cho danh
sách liên kết kép

```
11. NODE* GetNode(KDL x)
12. {
    |
    |
    |
20. }
```


Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu **NODE** để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
```

```
12. {
```

Tên hàm tạo node cho dslk kép là **GetNode**.

```
20. }
```

Tạo node cho dslk kép



- **Khái niệm:** Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu **NODE** để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
```

```
12. {
```

Có một tham số đầu vào, tên tham số là x, tham số là tham trị.

```
20. }
```

Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu **NODE** để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
```

```
12. {
```

KDL trả về của hàm **GetNode** là con trỏ kiểu cấu trúc **NODE**.

```
20. }
```

Tạo node cho dslk kép



- **Khái niệm:** Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu **NODE** để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
```

```
12. {
```

Về mặt bản chất hàm **GetNode** sẽ trả về một địa chỉ ô nhớ.

```
20. }
```

Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu NODE để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
12. {
    NODE* p=new NODE;
    20. }
```



Tạo node cho dslk kép

- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu `NODE` để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
```

```
12. {
```

```
    NODE* p=new NODE;
```

`p` là một biến con trỏ kiểu cấu trúc `NODE`.

```
20. }
```

Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu `NODE` để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
```

```
12. {
```

```
    NODE* p=new NODE;
```

Miền giá trị của biến con trỏ `p` là địa chỉ ô nhớ.

```
20. }
```

Tạo node cho dslk kép



- **Khái niệm:** Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu NODE để chứa thông tin đã được biết trước.

Tạo node cho danh
sách liên kết kép

```
11. NODE* GetNode(KDL x)
```

```
12. {
```

```
    NODE* p=new NODE;
```

new NODE là xin cấp phát động một vùng nhớ có kích thước bằng kích thước của kiểu dữ liệu NODE.

```
20. }
```


Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu **NODE** để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
```

```
12. {
```

```
    NODE* p=new NODE;
```

Nếu việc cấp phát thành công OS sẽ trả về địa chỉ ô nhớ đầu tiên của vùng nhớ được cấp phát, địa chỉ ô nhớ này được gán cho biến con trỏ **p**.

```
20. }
```

Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu NODE để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
```

```
12. {
```

```
    NODE* p=new NODE;
```

Nếu việc cấp phát thất bại, OS sẽ trả về một địa chỉ đặc biệt là địa chỉ NULL, địa chỉ NULL này sẽ được gán cho biến con trỏ **p**.

```
20. }
```

Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu **NODE** để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
```

```
12. {
```

```
    NODE* p=new NODE;
```

Như vậy, thông thường sau câu lệnh thứ 13, biến con trỏ **p** sẽ giữ địa chỉ ô nhớ đầu tiên của vùng nhớ có kích thước bằng kích thước của KDL **NODE**.

```
20. }
```

Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu NODE để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
12. {
13.     NODE* p=new NODE;
14.     if(p==NULL)
15.         return NULL;
16.     p->info = x;
17.     p->pNext=NULL;
18.     p->pPrev=NULL;
19.     return p;
20. }
```

Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu `NODE` để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
12. {
13.     NODE* p=new NODE;
14.
15.
16.
17.
18.
19.     return p;
20. }
```

Kết thúc lời gọi hàm và trả về địa chỉ ô nhớ đang được lưu trong biến con trỏ `p`.

Tạo node cho dslk kép



- **Khái niệm:** Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu **NODE** để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
```

```
12. {
```

```
13.     NODE* p=new NODE;
```

```
14.     if(p==NULL)
```

```
15.         return NULL;
```

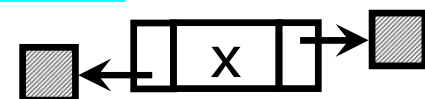
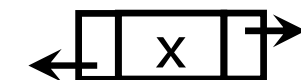
```
16.     p->info = x;
```

```
17.     p->pNext=NULL;
```

```
18.     p->pPrev=NULL;
```

```
19.     return p;
```

```
20. }
```



Tạo node cho dslk kép



— Phân tích câu lệnh dòng 11

11. **NODE* GetNode(KDL x)**

— Tên hàm tạo node cho dslk kép là **GetNode**.

— Có một tham số đầu vào, tên tham số là x, tham số là tham trị.

— KDL trả về của hàm **GetNode** là con trỏ kiểu cấu trúc **NODE**.

— Về mặt bản chất hàm **GetNode** sẽ trả về một địa chỉ ô nhớ.

— Phân tích câu lệnh dòng 19

19. **return p;**

— Kết thúc lời gọi hàm và trả về địa chỉ ô nhớ đang được lưu trong biến con trỏ **p**.

— Ý nghĩa của địa chỉ ô nhớ đang lưu biến con trỏ **p** xem ở slide ngay sau.

Tạo node cho dslk kép



Phân tích dòng lệnh 13.

13. `NODE *p=new NODE;`

- `p` là một biến con trỏ kiểu cấu trúc `NODE`.
- Miền giá trị của biến con trỏ `p` là địa chỉ ô nhớ.
- `new NODE` là xin cấp phát động một vùng nhớ có kích thước bằng kích thước của kiểu dữ liệu `NODE`.

- Nếu việc cấp phát thành công OS sẽ trả về địa chỉ ô nhớ đầu tiên của vùng nhớ được cấp phát, địa chỉ ô nhớ này được gán cho biến con trỏ `p`.
- Nếu việc cấp phát thất bại, OS sẽ trả về một địa chỉ đặc biệt là địa chỉ `NULL`, địa chỉ `NULL` này sẽ được gán cho biến con trỏ `p`.
- Như vậy, thông thường sau câu lệnh thứ 13, biến con trỏ `p` sẽ giữ địa chỉ ô nhớ đầu tiên của vùng nhớ có kích thước bằng kích thước của KDL `NODE`.

Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu NODE để chứa thông tin đã được biết trước.

Tạo node cho danh sách liên kết kép

```
11. NODE* GetNode(KDL x)
12. {
13.     NODE* p=new NODE;
14.     if(p==NULL)
15.         return NULL;
16.     p->info = x;
17.     p->pNext=NULL;
18.     p->pPrev=NULL;
19.     return p;
20. }
```



TẠO NODE CHO DSLK KÉP SỐ NGUYÊN

Tạo node cho dslk kép



- Khái niệm: Tạo node cho danh sách liên kết kép là xin cấp phát bộ nhớ có kích thước bằng với kích thước của kiểu dữ liệu NODE để chứa thông tin đã được biết trước.

Tạo node cho dslk
kép các số nguyên

```
11. NODE* GetNode(int x)
12. {
13.     NODE* p=new NODE;
14.     if(p==NULL)
15.         return NULL;
16.     p->info = x;
17.     p->pNext=NULL;
18.     p->pPrev=NULL;
19.     return p;
20. }
```



Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

Hồ Thái Ngọc

ThS. Võ Duy Nguyên

TS. Nguyễn Tấn Trần Minh Khang



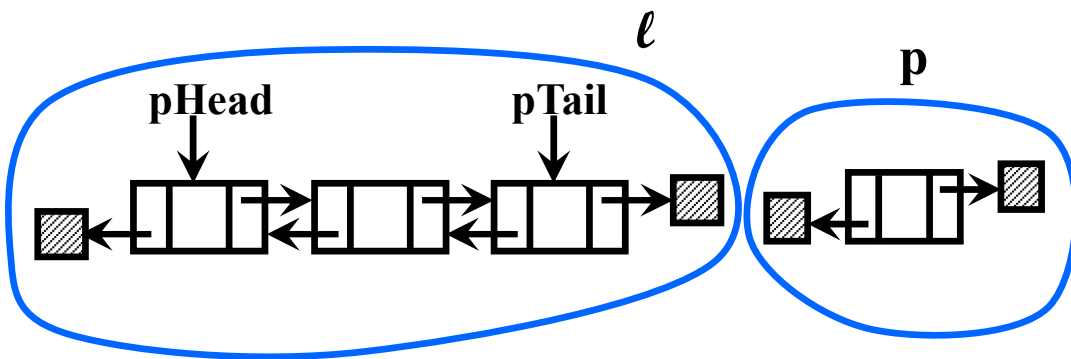
THÊM NODE VÀO CUỐI DSLK KÉP

Thêm node vào cuối dslk kép



— Khái niệm: Thêm một node vào cuối danh sách liên kết kép là gắn node đó vào cuối danh sách.

— Hình vẽ



Thêm vào cuối dslk kép

```
11. void AddTail(LIST&l, NODE*p)
12. {
13.     if(l.pHead==NULL)
14.         l.pHead=l.pTail=p;
15.     else
16.     {
17.         l.pTail->pNext=p;
18.         p->pPrev=l.pTail;
19.         l.pTail = p;
20.     }
21. }
```



Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

Hồ Thái Ngọc

ThS. Võ Duy Nguyên

TS. Nguyễn Tấn Trần Minh Khang



NHẬP DSLK KÉP CÁC SỐ NGUYÊN TỪ FILE

Nhập dslk kép các số nguyên từ file



```
11.int Nhap(LIST& l, string filename)
```

```
12.{
```

```
13.    ifstream fi(filename);
```

Nhập danh
sách liên
kết kép từ
file

Nhập dslk kép các số nguyên từ file



```
11.int Nhap(LIST& l, string filename)
```

```
12.{
```

```
13.    ifstream fi(filename);
```

Ý1: fi là đối tượng thuộc lớp
ifstream.

Nhập danh
sách liên
kết kép từ
file

Nhập dslk kép các số nguyên từ file



```
11.int Nhap(LIST& l, string filename)
```

```
12.{
```

```
13.    ifstream fi(filename);
```

Ý2: Dòng lệnh số 13 khai báo đối tượng fi với đối số có tên filename và có kiểu string.

Nhập danh sách liên kết kép từ file

Nhập dslk kép các số nguyên từ file



```
11.int Nhap(LIST& l, string filename)
```

```
12.{
```

```
13.    ifstream fi(filename);
```

Ý3: Khi chương trình thực hiện tới dòng lệnh số 13. Đối tượng fi gọi thực hiện phương thức thiết lập với tham số có kiểu string. Phương thức thiết lập sẽ mở tập tin có tên lưu trong biến filename.

Nhập danh sách liên kết kép từ file

Nhập dslk kép các số nguyên từ file



```
11. int Nhap(LIST& l, string filename)
12. {
13.     ifstream fi(filename);
14.     if (fi.fail() == true)
15.         return 0;
16.     int n;
17.     int x;
18.     fi >> n;
19.     Init(l);
20.     ...
```

Nhập danh
sách liên
kết kép từ
file

Nhập dslk kép các số nguyên từ file



```
11. | ...
12. | for (int i = 1; i <= n; i++)
13. | {
14. |     fi >> x;
15. |     NODE* p = GetNode(x);
16. |     if (p != NULL)
17. |         AddTail(1, p);
18. | }
19. | return 1;
20. | }
```

Nhập danh
sách liên
kết kép từ
file



Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

Hồ Thái Ngọc

ThS. Võ Duy Nguyên

TS. Nguyễn Tấn Trần Minh Khang

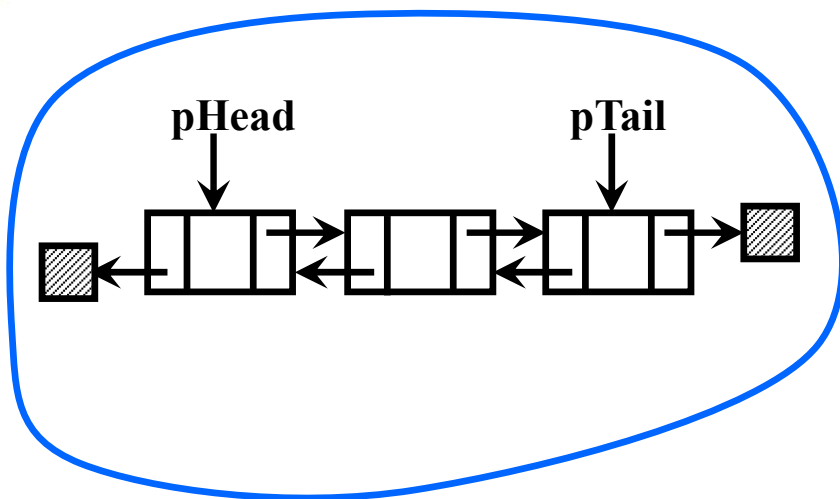


XUẤT DSLK KÉP CÁC SỐ NGUYÊN

Xuất dslk kép các số nguyên



ℓ



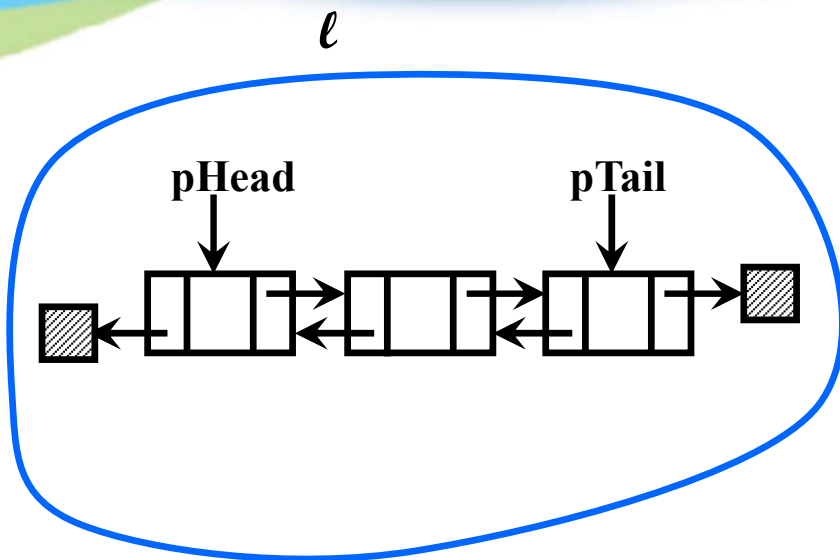
Xuất danh sách liên
kết kép ra màn hình

```
11. void Xuat(LIST l)
12. {
13.     NODE* p = l.pHead;
14.     while (p != NULL)
15.     {
16.         cout<<p->info;
17.         p = p->pNext;
18.     }
19. }
```



ĐẾM NODE TRONG DSLK KÉP

Đếm node trong ds lk kép



Đếm node trong danh sách liên kết kép

```
11. int DemNode(LIST l)
12. {
13.     int dem = 0;
14.     NODE* p = l.pHead;
15.     while (p != NULL)
16.     {
17.         dem++;
18.         p = p->pNext;
19.     }
20.     return dem;
21. }
```



Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

Hồ Thái Ngọc

ThS. Võ Duy Nguyên

TS. Nguyễn Tấn Trần Minh Khang



XUẤT DSLK KÉP CÁC SỐ NGUYÊN RA FILE

```
11.int Xuat(LIST l, string filename)
```

```
12.{
```

```
13.    ofstream fo(filename);
```

```
24.}
```



Xuất danh
sách liên
kết kép ra
file

```
11.int Xuat(LIST l, string filename)
```

```
12.{
```

```
13.    ofstream fo(filename);
```

Ý1: fo là đối tượng thuộc lớp
ofstream.

```
24.}
```



Xuất danh
sách liên
kết kép ra
file

```
11.int Xuat(LIST l, string filename)
```

```
12.{
```

```
13.    ofstream fo(filename);
```

Ý2: Dòng lệnh số 13 khai báo đối tượng fo với đối số có tên filename và có kiểu string.

```
24.}
```



Xuất danh
sách liên
kết kép ra
file


```
11.int Xuat(LIST l, string filename)
12.{
13.    ofstream fo(filename);
```

Ý3: Khi chương trình thực hiện tới dòng lệnh số 13. Đối tượng fo gọi thực hiện phương thức thiết lập với tham số có kiểu string. Phương thức thiết lập sẽ mở tập tin có tên lưu trong biến filename.

```
24.}
```



Xuất danh
sách liên
kết kép ra
file



```
11.int Xuat(LIST l, string filename)
12.{
13.    ofstream fo(filename);
14.    if (fo.fail()==true)
15.        return 0;
16.    fo << setw(5) << DemNode(1) << endl;
17.    NODE* p = l.pHead;
18.    while (p != NULL)
19.    {
20.        fo << setw(5) << p->info;
21.        p = p->pNext;
22.    }
23.    return 1;
24.}
```

Xuất danh
sách liên
kết kép ra
file



Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

Hồ Thái Ngọc

ThS. Võ Duy Nguyên

TS. Nguyễn Tấn Trần Minh Khang



SẮP TĂNG DSLK KÉP CÁC SỐ NGUYÊN

Sắp tăng dslk kép các số nguyên



```
11. void InterchangeSort(LIST& l)
12. {
13.     for (NODE*p=l.pHead; p->pNext!=NULL; p=p->pNext)
14.         for (NODE*q=p->pNext; q!=NULL; q=q->pNext)
15.             if (p->info > q->info)
16.                 swap(p->info, q->info);
17. }
```

Sắp xếp danh sách
liên kết kép tăng dần



Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

Hồ Thái Ngọc

ThS. Võ Duy Nguyên

TS. Nguyễn Tấn Trần Minh Khang