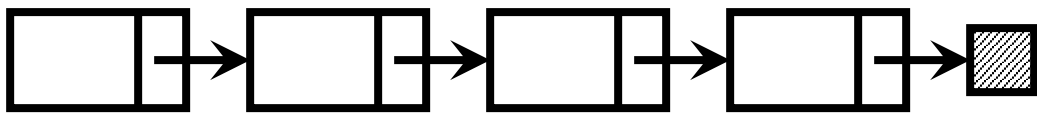


# **Chương 1**

## **CẤU TRÚC DỮ LIỆU**

### **DANH SÁCH LIÊN KẾT ĐƠN**

# 1. HÌNH ẢNH DANH SÁCH LIÊN KẾT ĐƠN

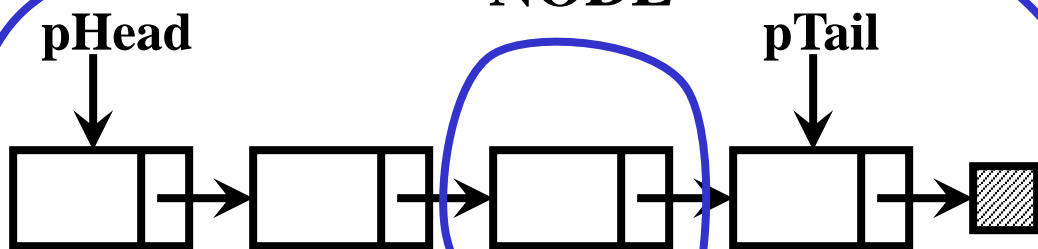


NULL

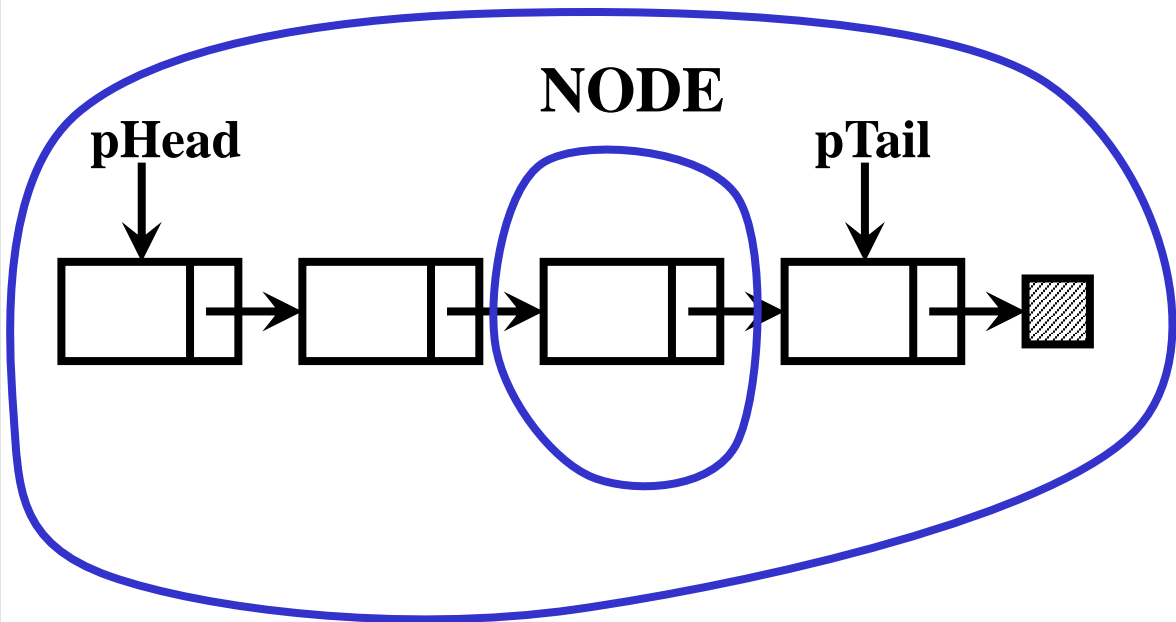
## 2. CẤU TRÚC DỮ LIỆU DANH SÁCH LIÊN KẾT ĐƠN

LIST

NODE



# LIST



```

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

```

## 2. CẤU TRÚC DỮ LIỆU DANH SÁCH LIÊN KẾT ĐƠN

– Ví dụ 1: Hãy khai báo CTDL cho dslk đơn các số nguyên.

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

## 2. CẤU TRÚC DỮ LIỆU DANH SÁCH LIÊN KẾT ĐƠN

– Ví dụ 2: Hãy khai báo CTDL cho dslk đơn các số thực.

```
10. struct node
```

```
11. {
```

```
12. |     float info;
```

```
13. |     struct node* pNext;
```

```
14. };
```

```
15. typedef struct node NODE;
```

```
16. struct list
```

```
17. {
```

```
18. |     NODE* pHead;
```

```
19. |     NODE* pTail;
```

```
20. };
```

```
21. typedef struct list LIST;
```

## 2. CẤU TRÚC DỮ LIỆU DANH SÁCH LIÊN KẾT ĐƠN

**Ví dụ 3: Hãy khai báo ctdl cho dslk đơn các phân số.**

```
10.struct phanso
11.{
12.|    int tu;
13.|    int mau;
14.};
15.typedef struct phanso PHANSO;
16.struct node
17.{
18.|    PHANSO info;
19.|    struct node*pNext;
20.};
21.typedef struct node NODE;
22.struct list
23.{
24.|    NODE*pHead;
25.|    NODE*pTail;
26.};
27.typedef struct list LIST;
```

## 2. CẤU TRÚC DỮ LIỆU DANH SÁCH LIÊN KẾT ĐƠN

**Ví dụ 4: Hãy khai báo ctdl cho dslk đơn tọa độ các điểm trong mp Oxy.**

```
10.struct diem
11.{
12.|    float x;
13.|    float y;
14.};
15.typedef struct diem DIEM;
16.struct node
17.{
18.|    DIEM info;
19.|    struct node*pNext;
20.};
21.typedef struct node NODE;
22.struct list
23.{
24.|    NODE*pHead;
25.|    NODE*pTail;
26.};
27.typedef struct list LIST;
```



### 3. KHỞI TẠO DANH SÁCH LIÊN KẾT ĐƠN

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

```
1. void Init(LIST &l)  
2. {  
3.     |    l.pHead = NULL;  
4.     |    l.pTail = NULL;  
5. }
```

## 4. KIỂM TRA DANH SÁCH LIÊN KẾT ĐƠN RỖNG

- Khái niệm: Kiểm tra danh sách liên kết đơn 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.

- Định nghĩa hàm

```
1. int IsEmpty(LIST l)  
2. {  
3.     if (l.pHead==NULL)  
4.         return 1;  
5.     return 0;  
6. }
```

## 5. TẠO NODE CHO DANH SÁCH LIÊN KẾT ĐƠN

- **Khái niệm:** Tạo node cho danh sách liên kết đơn 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.

- Định nghĩa hàm trừu tượng


1. **NODE\*** **GetNode** (KDL x)

2. {

3.     NODE \*p=new NODE; 

4.     if (p==NULL)

5.         return NULL;

6.     p->info = x; 

7.     p->pNext = NULL; 

8.     return p;

9. }

TS. Nguyễn Tấn Trần Minh Khang

Bài 1 - 11

ThS. Cáp Phạm đình Thăng

## 5. TẠO NODE CHO DANH SÁCH LIÊN KẾT ĐƠN

- Ví dụ 1: Định nghĩa hàm tạo một **NODE** cho dslk đơn các số nguyên để chứa thông tin đã được biết trước.

- Định nghĩa hàm

```
1. NODE* GetNode (int x)
2. {
3.     NODE *p = new NODE;
4.     if (p==NULL)
5.         return NULL;
6.     p->info = x;
7.     p->pNext = NULL;
8.     return p;
```

## 5. TẠO NODE CHO DANH SÁCH LIÊN KẾT ĐƠN

- Ví dụ 2: Định nghĩa hàm tạo một **NODE** cho **dslk** đơn các số thực để chứa thông tin đã được biết trước.

- Định nghĩa hàm

```
1. NODE* GetNode (float x)
2. {
3.     NODE *p = new NODE;
4.     if (p==NULL)
5.         return NULL;
6.     p->info = x;
7.     p->pNext = NULL;
8.     return p;
9. }
```

## 5. TẠO NODE CHO DANH SÁCH LIÊN KẾT ĐƠN

- Ví dụ 3: Định nghĩa hàm tạo một **NODE** cho dslk đơn các phân số để chứa thông tin đã được biết trước.

```
1. NODE* GetNode (PHANSO x)
2. {
3.     NODE *p = new NODE;
4.     if (p==NULL)
5.         return NULL;
6.     p->info = x;
7.     p->pNext = NULL;
8.     return p;
9. }
```

## 5. TẠO NODE CHO DANH SÁCH LIÊN KẾT ĐƠN

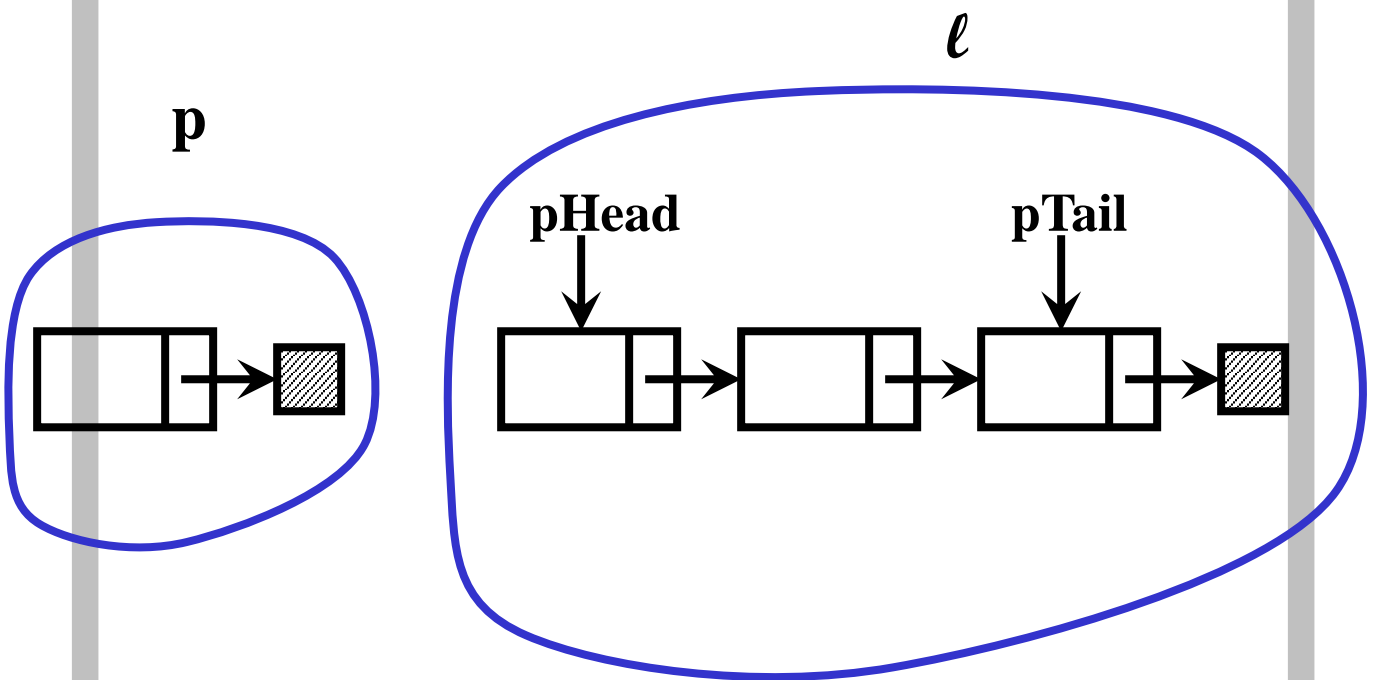
- Ví dụ 4: Định nghĩa hàm tạo một **NODE** cho dslk đơn tọa độ các điểm trong mặt phẳng Oxy.

```
1. NODE* GetNode (DIEM P)
2. {
3.     NODE *p = new NODE;
4.     if (p==NULL)
5.         return NULL;
6.     p->info = P;
7.     p->pNext = NULL;
8.     return p;
9. }
```

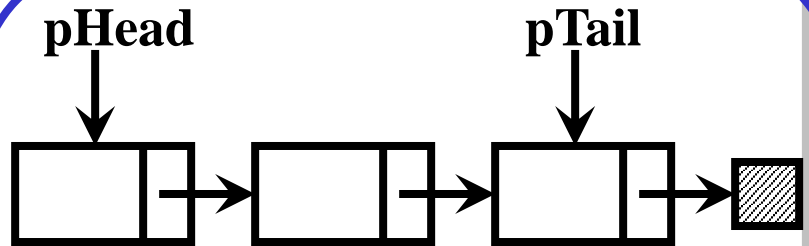
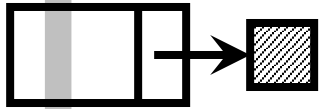
## 6. THÊM MỘT NODE VÀO ĐẦU DANH SÁCH LIÊN KẾT ĐƠN

– Khái niệm: Thêm một node vào đầu danh sách liên kết đơn là gắn node đó vào đầu danh sách.

– Hình vẽ





**p**

– Định nghĩa hàm:

```

11. void AddHead (LIST& $\ell$ , NODE*p)
12. {
13.     if ( $\ell$ .pHead==NULL)
14.          $\ell$ .pHead =  $\ell$ .pTail = p;
15.     else
16.     {
17.         p->pNext =  $\ell$ .pHead;
18.          $\ell$ .pHead = p;
19.     }
20. }

```

## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

- Khái niệm: Nhập từ bàn phím dslk đơn là lần lượt nhập các thông tin của từng node trong danh sách.
- Định nghĩa hàm trừu tượng

```
10. void Input (LIST&ℓ)
11. {
12.     int n;
13.     printf("Nhap n: ");
14.     scanf("%d", &n);
15.     Init(ℓ);
16.     for(int i=1; i<=n; i++)
17.     {
18.         KDL x;
19.         Nhap(x);
20.         NODE*p = GetNode(x);
21.         if (p!=NULL)
22.             AddHead(ℓ, p);
23.     }
24. }
```

## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

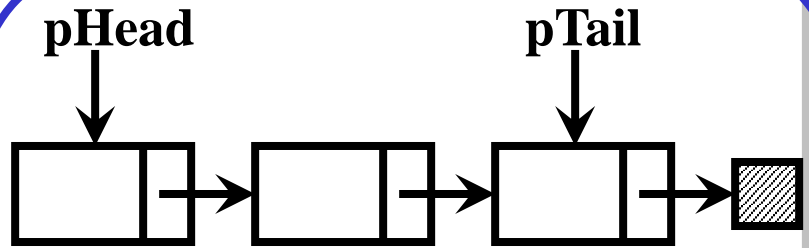
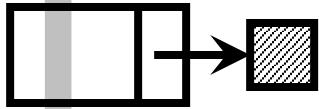
- Ví dụ 1: Nhập danh sách liên kết đơn các số nguyên.

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

## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

- Ví dụ 1: Nhập danh sách liên kết đơn các số nguyên.

```
1. void Init (LIST&l)
2. {
3.     l.pHead = NULL;
4.     l.pTail = NULL;
5. }
6. NODE* GetNode (int x)
7. {
8.     NODE *p = new NODE;
9.     if (p==NULL)
10.         return NULL;
11.     p->info = x;
12.     p->pNext = NULL;
13.     return p;
14. }
```

**p**

– Định nghĩa hàm:

```

11. void AddHead (LIST& $\ell$ , NODE*p)
12. {
13.     if ( $\ell$ .pHead==NULL)
14.          $\ell$ .pHead =  $\ell$ .pTail = p;
15.     else
16.     {
17.         p->pNext =  $\ell$ .pHead;
18.          $\ell$ .pHead = p;
19.     }
20. }

```

## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

– Ví dụ 1: Nhập danh sách liên kết đơn các số nguyên.

– Định nghĩa hàm

```
10. void Input (LIST&ℓ)
11. {
12.     int n;
13.     printf("Nhap n: ");
14.     scanf("%d", &n);
15.     Init(ℓ);
16.     for(int i=1; i<=n; i++)
17.     {
18.         int x;
19.         printf("Nhap so nguyen:");
20.         scanf("%d", &x);
21.         NODE*p = GetNode(x);
22.         if (p!=NULL)
23.             AddHead(ℓ, p);
24.     }
25. }
```

## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

- Ví dụ 2: Nhập danh sách liên kết đơn các số thực.

```
10.struct node
11.{
12.|    float info;
13.|    struct node*pNext;
14.};
15.typedef struct node NODE;
16.struct list
17.{
18.|    NODE*pHead;
19.|    NODE*pTail;
20.};
21.typedef struct list LIST;
```

## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

- Ví dụ 2: Nhập danh sách liên kết đơn các số thực.

```
1. void Init (LIST&l)
2. {
3.     l.pHead = NULL;
4.     l.pTail = NULL;
5. }
6. NODE* GetNode (float x)
7. {
8.     NODE *p = new NODE;
9.     if (p==NULL)
10.         return NULL;
11.     p->info = x;
12.     p->pNext = NULL;
13.     return p;
14. }
```



## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

– Ví dụ 2: Nhập danh sách liên kết đơn các số thực.

– Định nghĩa hàm

```
11. void Input (LIST&ℓ)
12. {
13.     int n;
14.     printf("Nhap n: ");
15.     scanf("%d", &n);
16.     Init(ℓ);
17.     for(int i=1; i<=n; i++)
18.     {
19.         float x;
20.         printf("Nhap so thuc: ");
21.         scanf("%f", &x);
22.         NODE*p=GetNode(x);
23.         if (p!=NULL)
24.             AddHead(ℓ, p);
25.     }
26. }
```

## 2. CẤU TRÚC DỮ LIỆU DANH SÁCH LIÊN KẾT ĐƠN

Ví dụ 3: Hãy định nghĩa hàm nhập dslk đơn các phân số.

```
10.struct phanso
11.{
12.|    int tu;
13.|    int mau;
14.};
15.typedef struct phanso PHANSO;
16.struct node
17.{
18.|    PHANSO info;
19.|    struct node*pNext;
20.};
21.typedef struct node NODE;
22.struct list
23.{
24.|    NODE*pHead;
25.|    NODE*pTail;
26.};
27.typedef struct list LIST;
```

## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

- Ví dụ 3: Hãy định nghĩa hàm nhập dslk đơn các phân số.

```
1. void Init(LIST&l)
2. {
3.     |    l.pHead = NULL;
4.     |    l.pTail = NULL;
5. }
6. NODE* GetNode(PHANSO x)
7. {
8.     |    NODE *p = new NODE;
9.     |    if (p==NULL)
10.    |        return NULL;
11.    |    p->info = x;
12.    |    p->pNext = NULL;
13.    |    return p;
14. }
```

## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

– Ví dụ 3: Nhập danh sách liên kết đơn các phân số.

– Định nghĩa hàm

```
11. void Nhap (PHANSO &x)
12. {
13.     printf("Nhap tu: ");
14.     scanf("%d", &x.tu);
15.     printf("Nhap mau: ");
16.     scanf("%d", &x.mau);
17. }
```

## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

– Ví dụ 3: Nhập danh sách liên kết đơn các phân số.

– Định nghĩa hàm

```
11. void Input (LIST&ℓ)
12. {
13.     int n;
14.     printf("Nhap n: ");
15.     scanf("%d", &n);
16.     Init(ℓ);
17.     for(int i=1; i<=n; i++)
18.     {
19.         PHANSO x;
20.         printf("Nhap phan so: ");
21.         Nhap(x);
22.         NODE*p=GetNode(x);
23.         if (p!=NULL)
24.             AddHead(ℓ, p);
25.     }
26. }
```

## 2. CẤU TRÚC DỮ LIỆU DANH SÁCH LIÊN KẾT ĐƠN

Ví dụ 4: Hãy định nghĩa hàm nhập `dslk` đơn tọa độ các điểm trong mp Oxy

```
10. struct diem
11. {
12. |    float x;
13. |    float y;
14. };
15. typedef struct diem DIEM;
16. struct node
17. {
18. |    DIEM info;
19. |    struct node* pNext;
20. };
21. typedef struct node NODE;
22. struct list
23. {
24. |    NODE* pHead;
25. |    NODE* pTail;
26. };
27. typedef struct list LIST;
```

## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

- Ví dụ 4: Hãy định nghĩa hàm nhập dslk đơn tọa độ các điểm trong mp Oxy

```
1. void Init (LIST&l)
2. {
3.     l.pHead = NULL;
4.     l.pTail = NULL;
5. }
6. NODE* GetNode (DIEM M)
7. {
8.     NODE *p = new NODE;
9.     if (p==NULL)
10.         return NULL;
11.     p->info = M;
12.     p->pNext = NULL;
13.     return p;
14. }
```

## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

– Ví dụ 4: Hãy định nghĩa hàm nhập dữ liệu đơn tọa độ các điểm trong mp Oxy

– Định nghĩa hàm

```
11. void Nhap (DIEM &P)
12. {
13.     float temp;
14.     printf("Nhap x: ");
15.     scanf("%f", &temp);
16.     P.x = temp;
17.     printf("Nhap y: ");
18.     scanf("%f", &temp);
19.     P.y = temp;
20. }
```



## 7. NHẬP TỪ BÀN PHÍM DANH SÁCH LIÊN KẾT ĐƠN

– Ví dụ 4: Hãy định nghĩa hàm nhập dsk đơn tọa độ các điểm trong mp Oxy

– Định nghĩa hàm

```
11. void Input (LIST&l)
12. {
13.     int n;
14.     printf("Nhap n: ");
15.     scanf("%d", &n);
16.     Init(l);
17.     for(int i=1; i<=n; i++)
18.     {
19.         DIEM M;
20.         printf("Nhap diem: ");
21.         Nhap(M);
22.         NODE*p=GetNode (M);
23.         if (p!=NULL)
24.             AddHead(l,p);
25.     }
26. }
```

## 8. DUYỆT TUẦN TỰ DANH SÁCH LIÊN KẾT ĐƠN

- Khái niệm: duyệt danh sách liên kết đơn là thăm qua tất cả các node mỗi node một lần.
- Định nghĩa hàm trừu tượng

```
11. KDL <Tên Hàm> (LIST l)
12. {
13.     ...
14.     NODE*p = l.pHead;
15.     while (p!=NULL)
16.     {
17.         ...
18.         p = p->pNext;
19.     }
20.     ...
21. }
```

## 8. DUYỆT TUẦN TỰ DANH SÁCH LIÊN KẾT ĐƠN

- Ví dụ: Định nghĩa hàm tính tổng các số lẻ trong dslk đơn các số nguyên.

```
10.struct node
```

```
11.{
```

```
12.    int info;
```

```
13.    struct node*pNext;
```

```
14.};
```

```
15.typedef struct node NODE;
```

```
16.struct list
```

```
17.{
```

```
18.    NODE*pHead;
```

```
19.    NODE*pTail;
```

```
20.};
```

```
21.typedef struct list LIST;
```

## 8. DUYỆT TUẦN TỰ DANH SÁCH LIÊN KẾT ĐƠN

- Ví dụ: Định nghĩa hàm tính tổng các số lẻ trong dslk đơn các số nguyên.
- Định nghĩa hàm

```
10. int TongLe (LIST l)
11. {
12.     int s = 0;
13.     NODE* p = l.pHead;
14.     while (p != NULL)
15.     {
16.         if (p->info % 2 != 0)
17.             s = s + p->info;
18.         p = p->pNext;
19.     }
20.     return s;
21. }
```

## 8. DUYỆT TUẦN TỰ DANH SÁCH LIÊN KẾT ĐƠN

- Ví dụ: Định nghĩa hàm xuất dslk đơn các số nguyên.
- Định nghĩa hàm

```
10. void Xuat (LIST l)
11. {
12.     NODE* p = l.pHead;
13.     while (p != NULL)
14.     {
15.         printf ("%4d", p->info) ;
16.         p = p->pNext;
17.     }
18. }
```

## 9. CHƯƠNG TRÌNH ĐẦU TIÊN DANH SÁCH LIÊN KẾT ĐƠN

- Bài toán: Viết chương trình thực hiện các yêu cầu sau:
  - + Nhập dslk đơn các số nguyên.
  - + Tính tổng các giá trị trong dslk đơn.
  - + Xuất dslk đơn.

- Chương trình

```
10. #include "stdio.h"
11. #include "conio.h"
12. #include "math.h"
13. #include "string.h"
14. struct node
15. {
16.     int info;
17.     struct node *pNext;
18. };
19. typedef struct node NODE;
```

```
10.struct list
11.{
12.    NODE*pHead;
13.    NODE*pTail;
14.};
15.typedef struct list LIST;
16.// Khai báo hàm
17.void Init(LIST&);
18.NODE* GetNode(int);
19.void AddHead(LIST&, NODE*);
20.void Input(LIST&);
21.void Output(LIST);
22.int Tong(LIST);
23.// Hàm main
24.void main()
25.{
26.    LIST lst;
27.    Input(lst);
28.    Output(lst);
29.    int kq = Tong(lst);
30.    printf("Tong la: %d", kq);
31.}
```

## 9. CHƯƠNG TRÌNH ĐẦU TIÊN DANH SÁCH LIÊN KẾT ĐƠN

```
1. void Init(LIST &l)  
2. {  
3.     l.pHead = NULL;  
4.     l.pTail = NULL;  
5. }
```



## 9. CHƯƠNG TRÌNH ĐẦU TIÊN DANH SÁCH LIÊN KẾT ĐƠN

– Định nghĩa hàm

```
1. NODE* GetNode (int x)
2. {
3.     NODE *p = new NODE;
4.     if (p==NULL)
5.         return NULL;
6.     p->info = x;
7.     p->pNext = NULL;
8.     return p;
9. }
```

## 9. CHƯƠNG TRÌNH ĐẦU TIÊN DANH SÁCH LIÊN KẾT ĐƠN

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

## 9. CHƯƠNG TRÌNH ĐẦU TIÊN DANH SÁCH LIÊN KẾT ĐƠN

```
10. void Input (LIST&l)
11. {
12.     int n;
13.     printf("Nhap n: ");
14.     scanf ("%d", &n);
15.     Init (l) ;
16.     for (int i=1; i<=n; i++)
17.     {
18.         int x;
19.         printf("Nhap so nguyen:") ;
20.         scanf ("%d", &x) ;
21.         NODE*p=GetNode (x) ;
22.         if (i!=1)
23.             AddHead (l, p) ;
24.     }
25. }
```

## 9. CHƯƠNG TRÌNH ĐẦU TIÊN DANH SÁCH LIÊN KẾT ĐƠN

```
1. void Output (LIST l)
2. {
3.     NODE*p = l.pHead;
4.     while (p!=NULL)
5.     {
6.         printf ("%4d", p->info);
7.         p = p->pNext;
8.     }
9. }
```

## 9. CHƯƠNG TRÌNH ĐẦU TIÊN DANH SÁCH LIÊN KẾT ĐƠN

– Định nghĩa hàm

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