# DANH SÁCH LIÊN KẾT

## MỤC TIÊU

Hoàn tất bài thực hành này, sinh viên có thể:

- Hiểu được các thành phần của danh sách liên kết.
- Thành thạo các thao tác trên danh sách liên kết: thêm phần tử, xóa phần tử, duyệt danh sách liên kết.
- Áp dụng cấu trúc dữ liệu danh sách liên kết vào việc giải quyết một số bài toán đơn giản.

Thời gian thực hành: **từ 120 phút đến 400 phút**

## TÓM TẮT

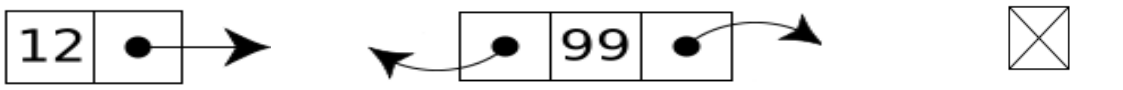
Danh sách liên kết là cấu trúc dữ liệu dùng để lưu trữ một danh sách (tập hợp hữu hạn) dữ liệu. Điểm đặc biệt của cấu trúc này là khả năng chứa của nó **động** (có thể mở rộng và thu hẹp dễ dàng).

Có các loại danh sách liên kết:

- Danh sách liên kết đơn
- Danh sách liên kết kép
- Danh sách liên kết vòng

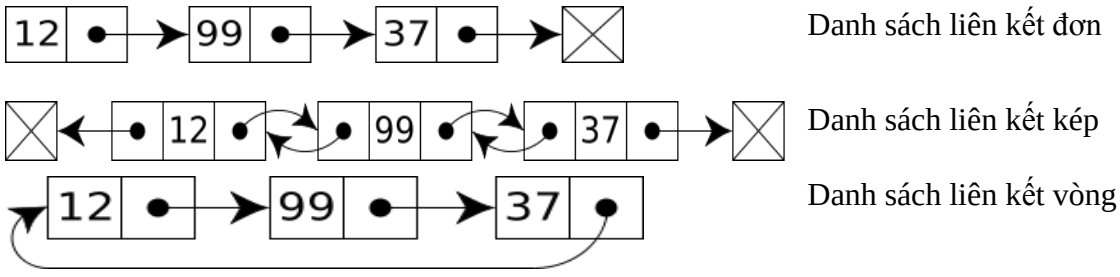
Mỗi danh sách liên kết là tập hợp các phần tử (node) chứa thông tin lưu trữ của dữ liệu. Giữa các phần tử có một hoặc nhiều liên kết để đảm bảo danh sách liên kết có thể giữ các phần tử này một cách chặt chẽ.

*Ví dụ 1:*




Phần tử có một liên kết | Phần tử có hai liên kết | Phần tử rỗng

*Ví dụ 2:*




Danh sách liên kết đơn

Danh sách liên kết kép

Danh sách liên kết vòng

Trong mỗi phần tử của danh sách liên kết, thông tin liên kết là vô cùng quan trọng. Chỉ cần một xử lý không cẩn thận có thể làm mất phần liên kết này thì danh sách liên kết sẽ bị '**gãy**' từ phần tử đó (không thể truy xuất tiếp các phần tử từ phần tử đó trở về trước hoặc trở về sau).

Các thao tác cơ bản trên danh sách liên kết:

- Thêm phần tử: vào đầu danh sách liên kết, vào cuối danh sách liên kết, vào trước/sau một phần tử trên danh sách liên kết.
- Xóa phần tử: ở đầu danh sách liên kết, ở cuối danh sách liên kết, một phần tử trên danh sách liên kết.
- Duyệt danh sách liên kết: để có thể đi được hết các phần tử trên danh sách liên kết.

## NỘI DUNG THỰC HÀNH

### Cơ bản

Sinh viên đọc kỹ phát biểu bài tập và thực hiện theo hướng dẫn:

*Tổ chức một danh sách liên kết đơn trong đó mỗi phần tử chứa thông tin dữ liệu nguyên.*

*Người dùng sẽ nhập các giá trị nguyên từ bàn phím. Với mỗi giá trị nguyên được nhập vào, giá trị đó được thêm vào **phía đầu** của danh sách liên kết. Nếu người dùng nhập vào giá trị **-1**, quá trình nhập dữ liệu sẽ kết thúc.*

*Sau đó, in ra các phần tử đang có trên danh sách liên kết.*

*Khi chương trình kết thúc, tất cả các phần tử trên danh sách liên kết bị **xóa bỏ** khỏi bộ nhớ.*

#### *Phân tích*

- Danh sách liên kết đơn gồm mỗi phần tử chứa dữ liệu nguyên. Thông tin của mỗi phần tử được khai báo theo ngôn ngữ C/C++ như sau:

```
struct NODE{
	int Key;
	NODE *pNext;
};
```

- Thao tác cần thực hiện: **thêm** phần tử nguyên **vào đầu** danh sách liên kết (**AddHead**), **in** các phần tử của danh sách liên kết (**PrintList**), **loại bỏ tất cả** các phần tử trên danh sách liên kết (**RemoveAll**).

#### *Chương trình mẫu*

```
#include "stdafx.h"

struct NODE{
	int Key;
	NODE *pNext;
};

NODE* CreateNode(int Data)
{
	NODE* pNode;
	pNode = new NODE; //Xin cấp phát bộ nhớ động để tạo một phần tử (node) mới
	if (pNode == NULL)
		return NULL;
	pNode->Key = Data;
	pNode->pNext = NULL;
	return pNode;
}

bool AddHead(NODE* &pHead, int Data)
{
	NODE *pNode;
	pNode = CreateNode(Data);
	if (pNode == NULL)
		return false;
	if (pHead == NULL)
		pHead = pNode;
	else  {
		pNode->pNext = pHead;
		pHead = pNode;
	}
	return true;
}
```

```
void PrintList(NODE *pHead)
{
	NODE *pNode;
	pNode = pHead;
	while (pNode != NULL)
	{
		printf("%5d", pNode->Key);
		pNode = pNode->pNext; //Ghi chu: thao tác này dùng để làm gì?
	}
}

void RemoveAll(NODE* &pHead) //Ghi chu: Ý nghĩa của ký hiệu &
{
	NODE *pNode;
	while (pHead != NULL)
	{
		pNode = pHead;
		pHead = pHead->pNext;
		delete pNode;
	}
	pHead = NULL; //Ghi chu: Tại sao phải thực hiện phép gán này?
}


int _tmain(int argc, _TCHAR* argv[])
{
	NODE *pRoot;
	//Ghi chu: Tại sao lại phải thực hiện phép gán phía dưới?
	pRoot = NULL;
	int Data;
	do
	{
		printf("Nhap vao du lieu, -1 de ket thuc: ");
		scanf("%d", &Data);
		if (Data == -1)
			break;
		AddHead(pRoot, Data);
	}while (Data != -1);
	printf("\nDu lieu da duoc nhap: \n");

	//Ghi chu: Chức năng của dòng lệnh phía dưới
	PrintList(pRoot);

	//Ghi chu : Chức năng của dòng lệnh phía dưới
	RemoveAll(pRoot);

	return 0;
}
```

### *Yêu cầu*

1. Biên dịch đoạn chương trình nêu trên.
2. Cho biết kết quả in ra màn hình khi người dùng nhập vào các dữ liệu sau:

	-1

	5	-1

	7	10	-23	-25	-4	1	-1

	1	2	3	4	-1

3. Nêu nhận xét ngắn gọn mối liên hệ giữa thứ tự nhập dữ liệu vào với thứ tự in dữ liệu ra màn hình.

4. Vẽ hình danh sách liên kết theo dữ liệu được nhập ở câu 2.

5. Nếu trong hàm `main` (`_tmain`) thứ tự hai dòng lệnh sau đây bị hoán đổi cho nhau thì kết quả kết xuất ra màn hình sẽ như thế nào đối với dữ liệu câu 2? Giải thích lý do?

```
	//Ghi chu
	PrintList(pRoot);
	//Ghi chu
	RemoveAll(pRoot);
```

6. Nếu trong hàm `main` (`_tmain`) vòng lặp `do…while` được thay đổi như dưới đây thì kết quả kết xuất ra màn hình sẽ như thế nào đối với dữ liệu câu 2? Giải thích lý do?

```
	do
	{
		printf("Nhap vao du lieu, -1 de ket thuc: ");
		scanf("%d", &Data);
		AddHead(pRoot, Data);
		if (Data == -1)
			break;
	}while (Data != -1);
```

7. Với các hàm `CreateNode`, `AddHead` được cung cấp sẵn, hãy cho biết ý nghĩa của các giá trị trả về của hàm.

8. Hãy ghi chú các thông tin bằng cách trả lời các câu hỏi ứng với các dòng lệnh có yêu cầu ghi chú (`//Ghi chú`) trong các hàm `RemoveAll, PrintList, _tmain`.

9. Kết quả sẽ như thế nào nếu hàm `RemoveAll` được thay đổi như dưới đây? Giải thích lý do

```
void RemoveAll(NODE* &pHead)
{
	while (pHead != NULL)
	{
		pHead = pHead->pNext;
		delete pHead;
	}
	pHead = NULL; //Ghi chu: Tại sao phải thực hiện phép gán này?
}
```

10. Giá trị cuối cùng của biến `pRoot` trong đoạn chương trình mẫu là gì? Giải thích lý do.

## Áp dụng – Nâng cao

1. Bổ sung chương trình mẫu cho phép tính **tổng giá trị** các phần tử trên danh sách liên kết đơn gồm các giá trị nguyên.

Gợi ý: tham khảo hàm `PrintList` để viết hàm **`SumList`**.

2. Bổ sung chương trình mẫu cho phép tìm **giá trị nguyên lớn nhất** trong số các phần tử nguyên trên danh sách liên kết đơn gồm các giá trị nguyên.

Gợi ý: tham khảo hàm `PrintList` để viết hàm **`MaxList`**.

3. Bổ sung chương trình mẫu cho phép tính **số lượng các phần tử** của danh sách liên kết đơn gồm các giá trị nguyên.

Gợi ý: tham khảo hàm `PrintList` để viết hàm **`CountList`**.

4. Bổ sung chương trình mẫu cho phép **thêm vào cuối** danh sách liên kết đơn một giá trị nguyên.

Gợi ý: tham khảo hàm `AddHead` để viết hàm **`AddTail`**.

5. Bổ sung chương trình mẫu cho phép **xóa phần tử đầu** danh sách liên kết đơn.

6. Bổ sung chương trình mẫu cho phép **xóa phần tử cuối** danh sách liên kết đơn.

7. Bổ sung chương trình mẫu cho biết **số lượng các phần tử** trên danh sách liên kết đơn có giá trị trùng với giá trị ***x*** được cho trước.

Gợi ý: tham khảo thao tác duyệt danh sách liên kết trong hàm **`PrintList`**.

8. Bổ sung chương trình mẫu cho phép tạo một danh sách liên kết đơn gồm các phần tử mang giá trị nguyên trong đó không có cặp phần tử nào mang giá trị giống nhau.

Gợi ý: sử dụng hàm `AddHead` hoặc `AddTail` có bổ sung thao tác kiểm tra phần tử giống nhau.

9. Cho sẵn một danh sách liên kết đơn gồm các phần tử mang giá trị nguyên và một giá trị nguyên ***x***. Hãy tách danh sách liên kết đã cho thành 2 danh sách liên kết: một danh sách gồm các phần tử có giá trị nhỏ hơn giá trị ***x*** và một danh sách gồm các phần tử có giá trị lớn hơn giá trị ***x***. Giải quyết trong 2 trường hợp:

a. Danh sách liên kết ban đầu không cần tồn tại.
b. Danh sách liên kết ban đầu bắt buộc phải tồn tại.

## BÀI TẬP THÊM

1. Đề xuất cấu trúc dữ liệu thích hợp để biểu diễn đa thức ($a_nx^n + a_{n-1}x^{n-1}+..+ a_1x + a_0$) bằng danh sách liên kết (đơn hoặc kép). Cài đặt các thao tác trên danh sách liên kết đơn biểu diễn đa thức:

a. In đa thức
b. Rút gọn đa thức
c. Cộng hai đa thức
d. Nhân hai đa thức

2. Thông tin của một quyển sách trong thư việc gồm các thông tin:

- Tên sách (chuỗi)
- Tác giả (chuỗi, tối đa 5 tác giả)
- Nhà xuất bản (chuỗi)
- Năm xuất bản (số nguyên)

a. Hãy tạo danh sách liên kết (đơn hoặc kép) chứa thông tin các quyển sách có trong thư viện (được nhập từ bàn phím).
b. Cho biết số lượng các quyển sách của một tác giả bất kỳ (nhập từ bàn phím).
c. Trong năm YYYY (nhập từ bàn phím), nhà xuất bản ABC (nhập từ bàn phím) đã phát hành những quyển sách nào.