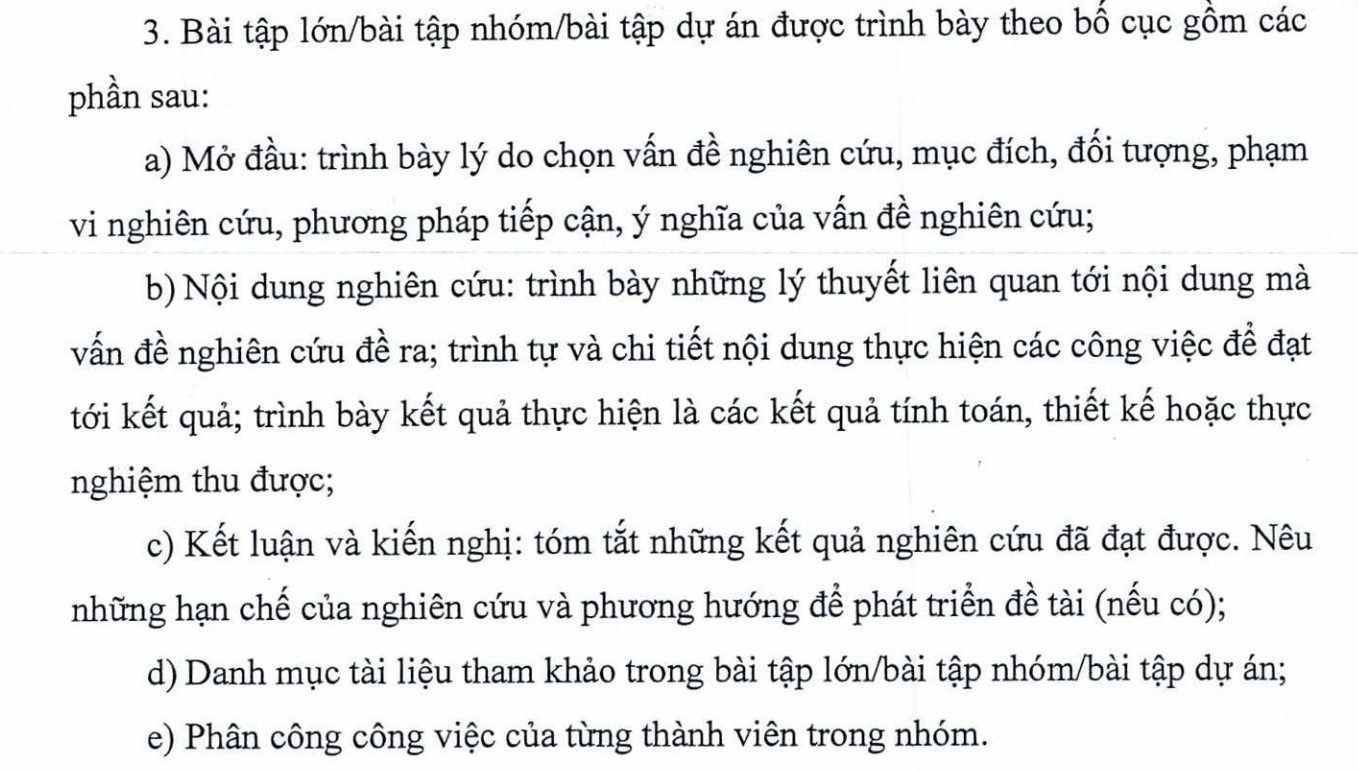
**Dưới đây là những gì cô đọng nhất về phần danh sách liên kết đơn mình sẽ tổng hợp lại và viết thành 1 cái khung cơ bản còn phần thiết kế Hoàng và Duy sẽ đảm nhiệm 2 bạn hỗ trợ lẫn nhau nhưng vai trò chính mình phân công là : Duy chuyên về word báo cáo, Hoàng là slide. Mọi người cố gắng làm tốt chúc nhóm mình A++ \*3 =)))**

**Anh em thiết kế nhớ làm theo yêu cầu của thầy như ảnh dưới có gì không hiểu cần tham khảo nói mình ngay để sửa đổi cho phù hợp với nội dung + tiện thuyết trình =)) chứ anh em làm tùy tiện lúc bị hỏi là không nói được đâu**

* **Trong bài chỗ nào mình in đậm câu từ anh em lúc làm cũng phải in đậm nhé tránh mất key word**

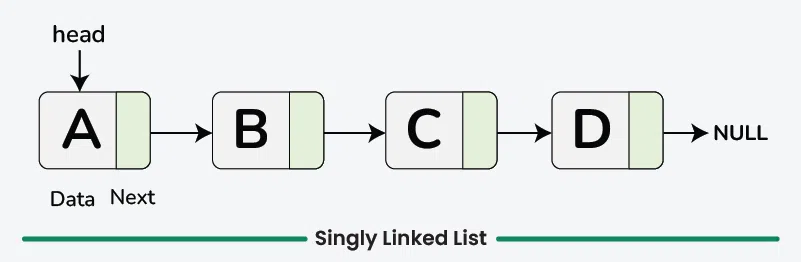


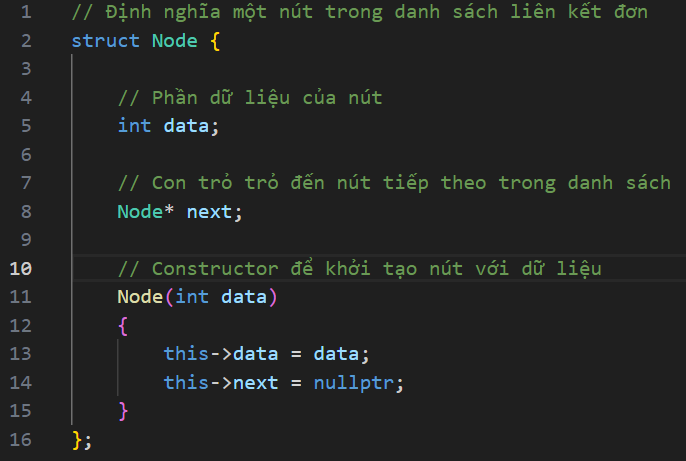
**--------------------------------------------------------------------------------------------------------------------------------**

**I.Danh sách liên kết đơn là gì ?**

**- Danh sách liên kết đơn** là một cấu trúc dữ liệu cơ bản, nó bao gồm các **nodes**(nút) trong đó mỗi nút chứa trong đó mỗi nút chứa một **trường dữ liệu** và tham thiếu tới nút tiếp theo của danh sách liên kết. Nút tiếp theo của nút cuối cùng là **null,** cho biết sự kết thúc của danh sách. Danh sách liên kết hỗ trợ các thao tác **chèn** và **xóa** hiệu quả.

**II.Cấu trúc của một node.**

* Trong một danh sách liên kết đơn, mỗi node bao gồm hai phần : dữ liệu và một con trỏ tới nút tiếp theo. Cấu trúc này cho phép các node được liên kết động với nhau tạo thành một chuỗi giống như một dãy liên tiếp
* hình x.x( danh sách liên kết đơn) Duy tự đánh số nhé



Trong ví dụ này lớp node chứa một trường dữ liệu số nguyên lưu trữ thông tin và một con trỏ tới node khác tạo liên kết tới nút tiếp theo trong danh sách.

**III. Các thao tác trên danh sách liên kết đơn**

* **Duyệt**
* **Tìm kiếm**
* **Tìm độ dài**
* **Chèn : chèn vào đầu, chèn vào cuối, chèn vào một vị trí cụ thể**
* **Xóa: xóa phần tử đầu, xóa phần tử cuối, xóa một nút cụ thể**

Được rồi hãy đi qua từng thao tác được nhắc đến ở trên, từng cái một.

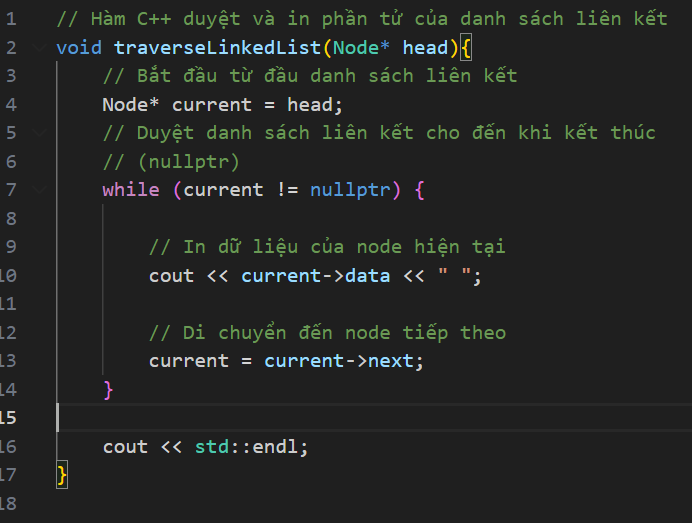
DUYỆT DANH SÁCH LIÊN KẾT ĐƠN

* Duyệt bao gồm việc truy cập từng nút trong DSLK đơn và thực hiện một số thao tác trên dữ liệu.
* Một hàm duyệt đơn giản sẽ in hoặc xử lý dữ liệu của từng nút

Các bước tiếp cận:

* Khởi tạo một con trỏ hiện tại ở đầu danh sách
* Sử dụng một vòng lặp while lặp qua danh sách cho đến khi con trỏ hiện tại đạt giá trị NULL
* Bên trong vòng lặp, in dữ liệu của nút hiện tại và di duyển con trỏ hiện tại sang nút tiếp theo

*Bên dưới là hàm duyệt DSLK đơn*



Hình ảnh anh em tự đánh số còn tên thì tôi ghi trên đấy phần code mà có cách nào đưa thẳng vào word thì tót nhìn cho đẹp k thì ảnh cũng đc

Đầu ra   
1 2 3

TÌM KIẾM TRONG DANH SÁCH LIÊN KẾT ĐƠN

* Tìm kiếm trong một danh sách liên kết đơn đề cập tới quá trình tìm kiếm một phần tử cụ thế hoặc giá trị bên trong các phần tử của danh sách liên kiết.

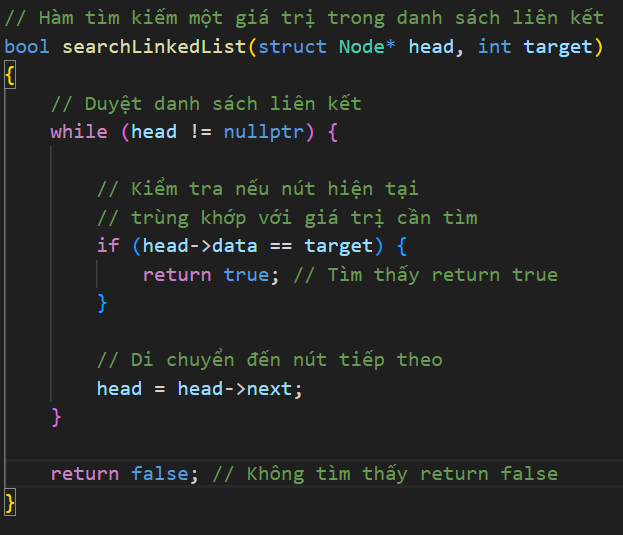
Các bước tiếp cận

1. Duyệt DSLK bắt đầu từ phần tử đầu
2. Kiểm tra dữ liệu của nút hiện tại có khớp với giá trị cần tìm hay không.

* Nếu có return **true**.

1. Nếu không thì, di chuyển tới node tiếp theo và lặp lại bước 2
2. Nếu tới cuối của danh sách mà không có kết quả nào phù hợp thì return **false**

Bên dưới làm hàm tìm kiếm trong DSLK đơn



ĐỘ DÀI CỦA DSLK ĐƠN

* Tìm độ dài trong danh sác liên kết đơn đề cập tới quá trình xác định tổng số node trong một DSLK đơn

Các bước tiếp cận:

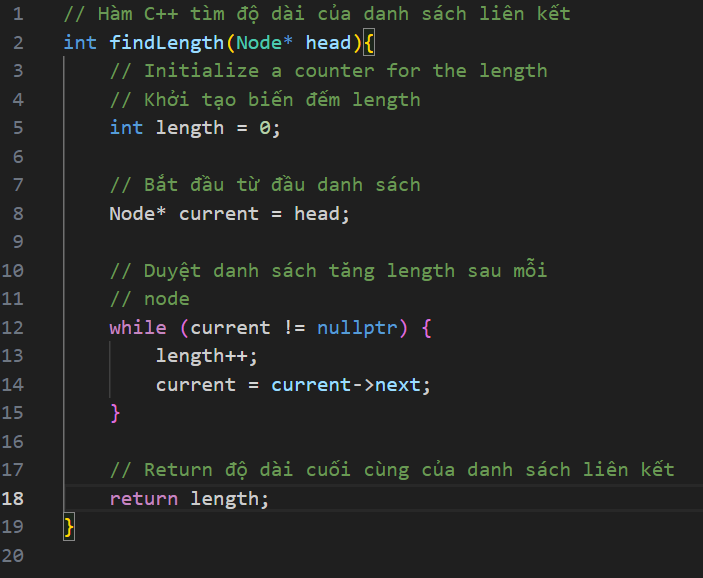
* Khởi tạo một biến đếm **length** từ 0.
* Bắt đầu từ đầu của danh sách, gán nó cho hiện tại
* Duyệt danh sách

+ Tăng **length** sau mỗi node.

+ Di chuyển tới node tiếp theo **( current = current->)**

* Return kết quả cuối cùng của **length**

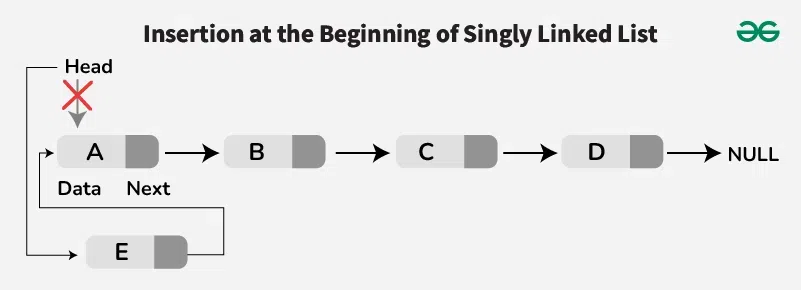
Bên dưới là hàm tìm độ dài trong DSLK đơn:



CHÈN TRONG DSLK ĐƠN

* Chèn là một thao tác cơ bản trong DSLK liên quan đến việc thêm một nút mới vào danh sách. Có một số tình huống để chèn:

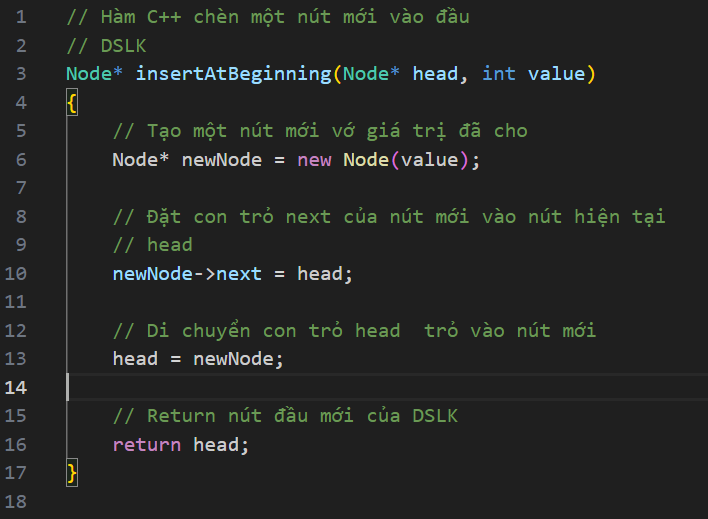
a.Chèn vào đầu DSLK đơn:



Các bước tiếp cận:

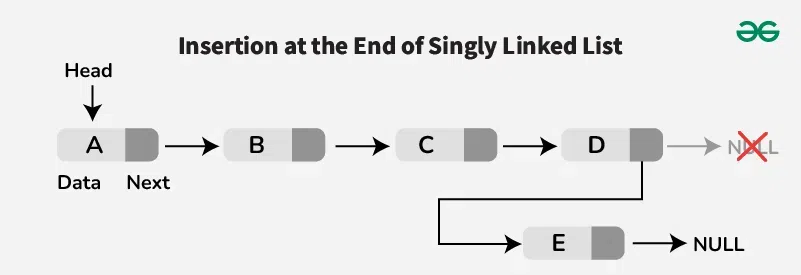
* Tạo một node mới với giá trị đã cho
* Đặt con trỏ **next** của nút mới trỏ đến nút hiện tại(head).
* Di chuyển con trỏ **head** để trỏ đến nút mới
* Return nút đầu (head) mới của DSLK

Bên dưới là hàn chèn đầu DSLK



b. chèn ở cuối DSLK đơn

- Để chèn một node ở cuối danh sách, duyệt danh sách cho đến khi đến nút cuối cùng, và sau đó liên kết node mới với node cuối cùng hiện tại.



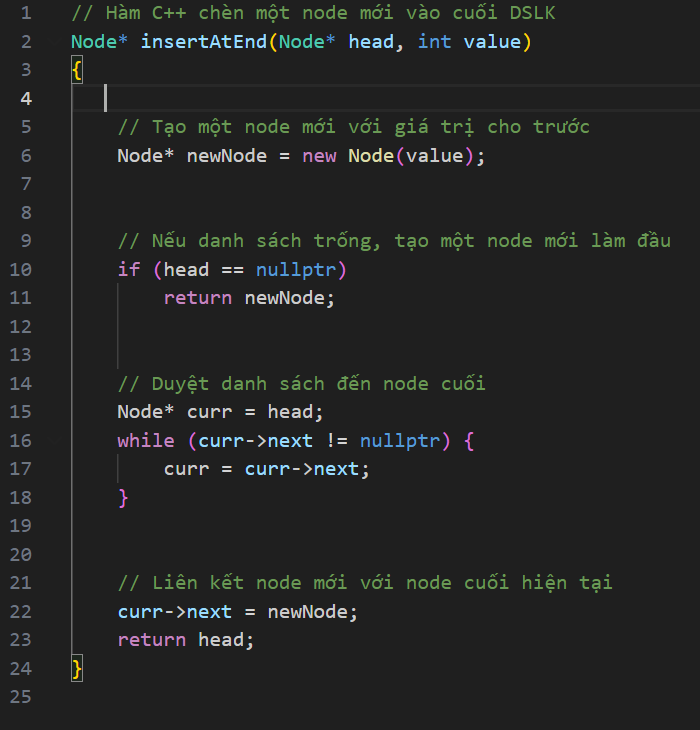
Các bước tiếp cận:

* Tạo một node mới với giá trị đã cho.
* Kiểm tra nếu danh sách trống

+ Nếu đúng như vậy, tạo một node mới làm đầu và return.

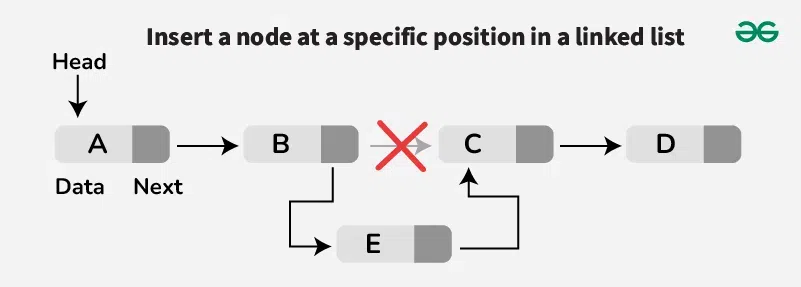
* Duyệt danh sách tới node cuối cùng
* Liên kết node mới với node cuối cùng hiện tại bằng cách đặt con trỏ next của nút cuối cùng tới nút mới

Bên dưới là hàm chèn ở cuối DSLK đơn



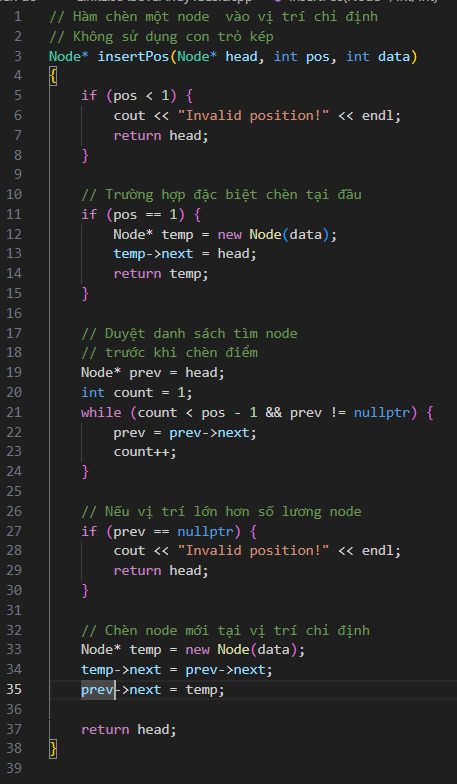
c.Chèn vào một vị trí cụ thể của DSLK đơn:

- Đề chèn một node vào một vị trí cụ thể, duyệt danh sách đến vị trí mong muốn, liên kết node mới với next node, và cập nhật liên kết tương ứng.



Chúng ta chủ yếu tìm node sau đó chúng ta cần chèn node mới. Nếu chúng ta bắt gặp một giá trị NULL trước khi đến node đó, điều đó có nghĩa là vị trí cung cấp không hợp lệ.

Bên dưới là hàm chèn tại vị trí cụ thể của DSLK đơn

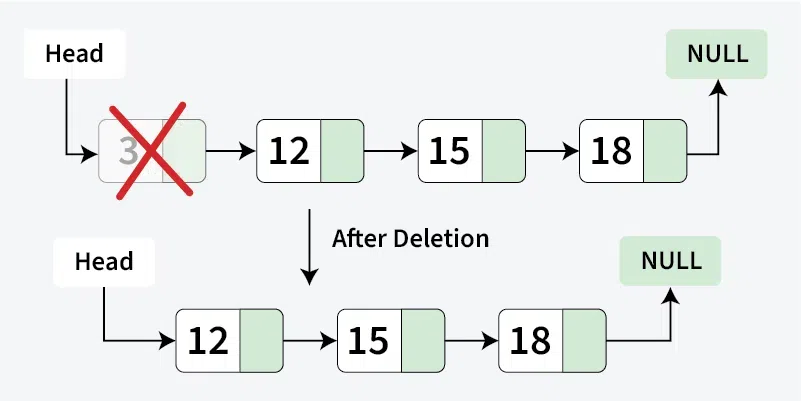


XÓA TRONG DSLK ĐƠN

* Xóa bao gồm việc loại bỏ một node từ DSLK. Tương tự với chèn, xóa cũng nhiều trường hợp xóa khác nhau:

1. Xóa ở đầu DSLK đơn:

Để xóa node đầu, cập nhật phần đầu để trỏ đến node thứ hai trong danh sách



*Xóa ở vị trí đầu trong danh sách*

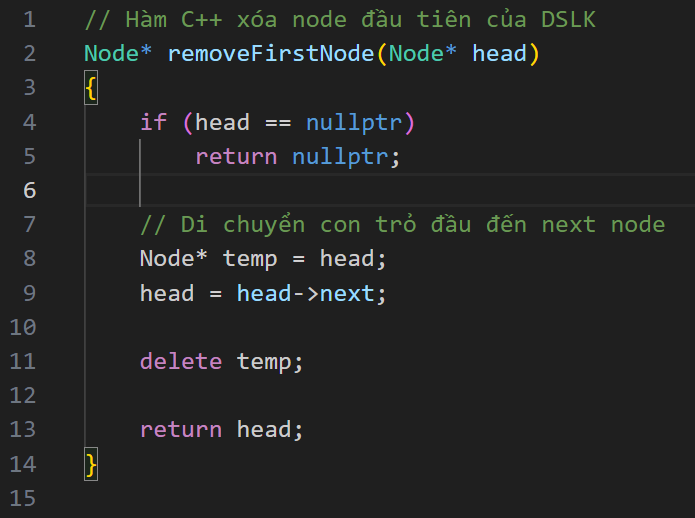
Cách bước tiếp cận:

* Kiểm tra nếu đầu là NULL.

+ nếu nó đúng, return NULL ( danh sách trống)

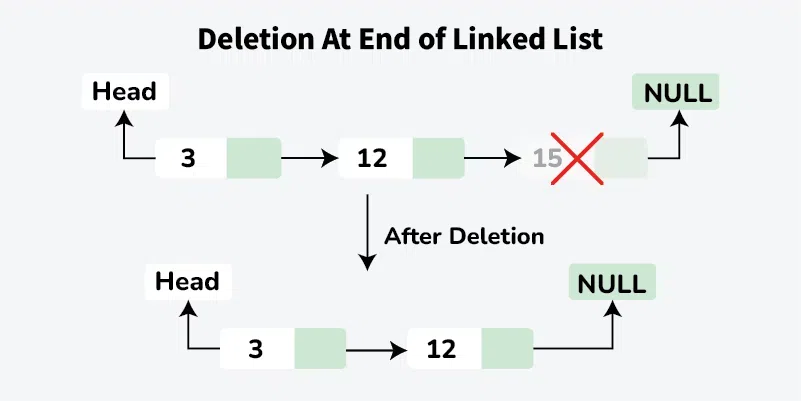
* Lưu trữ node đầu hiện tại trong một biếm tạm **temp.**
* Di chuyển đầu con trỏ với next node.
* Xóa node tạm.
* Return đầu mới của DSLK

Bên dưới là hàm xóa ở đầu DSLK đơn:



1. Xóa cuối DSLK đơn

* Để xóa node cuối, duyệt danh từ node thứ hai đến node cuối cùng và cập nhật trường tiếp theo của nó thành không.



Các bước tiếp cận:

* Kiểm tra nếu head là **NULL**.

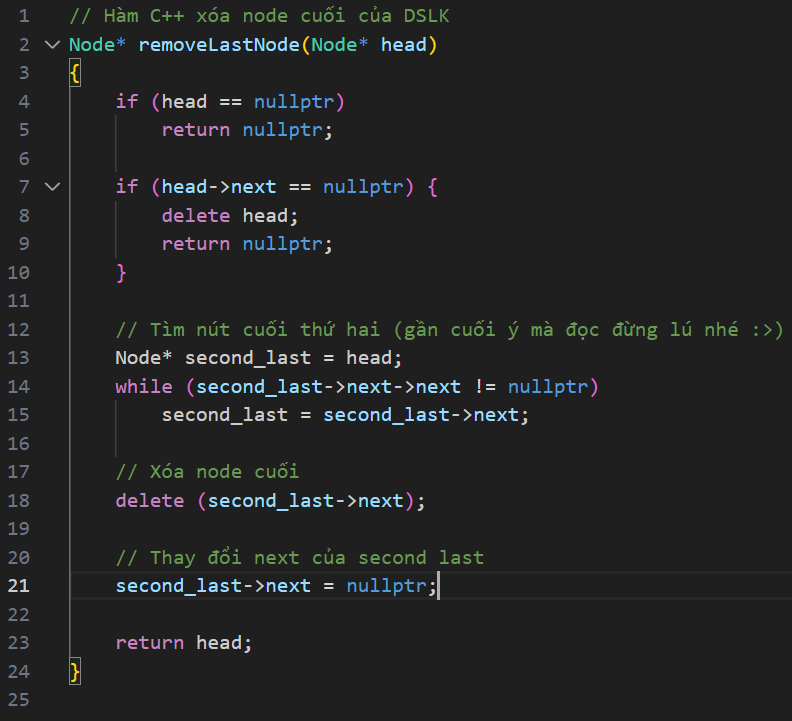
+ nếu đúng như vậy, return **NULL**(danh sách trống).

* Kiểm tra nếu head của **next** là **NULL** (chỉ có duy nhất 1 node trong danh sách).

+ nếu đúng như vậy, xóa head và return **NULL.**

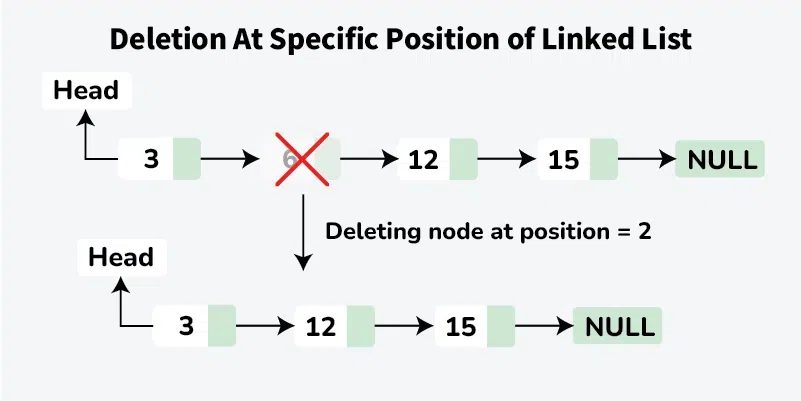
* Duyệt danh sách tìm node cuối cùng thứ hai **(second\_last)**.
* Xóa node cuối **(**node sau **second\_last).**
* Đặt con trỏ **next** của node cuối cùng thứ hai bằng **NULL**
* Return head của DSLK

Bên dưới là hàm xóa cuối DSLK đơn:



c.Xóa tại vị trí cụ thể của DSLK đơn.

- Để xóa một node tại vị trí cụ thể, duyệt danh sách vị trí mong muốn, cập nhật liên kết truyền cập nhật các liên kết để bỏ qua node bị xóa.

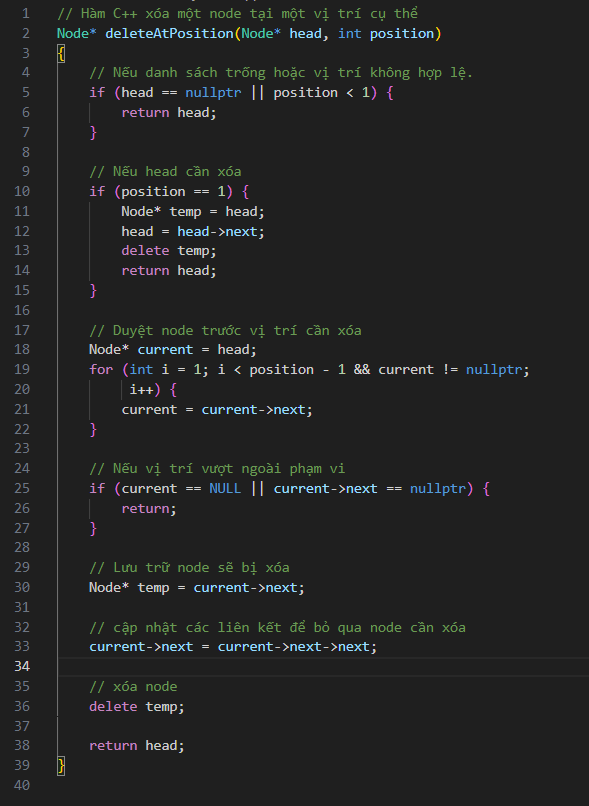


*Xóa một DSLK node tại vị trí chỉ định*

Các bước tiếp cận:

* Kiểm tra nếu danh sách trống hoặc vị trí không hợp lệ, return nếu có.
* Nếu head cần xóa, cập nhật head và xóa node.
* Duyệt node trước vị trí cần xóa.
* Nếu vị trí vượt ngoài phạm vi, return.
* Lưu trữ node sẽ bị xóa
* Cập nhật các liên kết để bỏ qua node.
* Xóa node được lưu trữ.

Bên dưới là hàm xóa tại vị trí cụ thể của DSLK đơn.

**