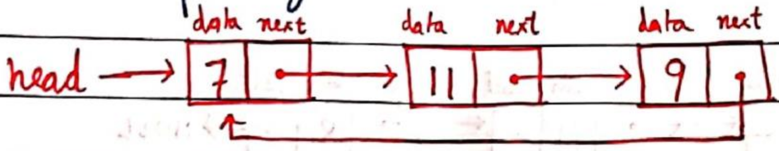
Circular Linked List

A circular linked list is a linked list where the last element points to the first element (head) hence forming a circular chain



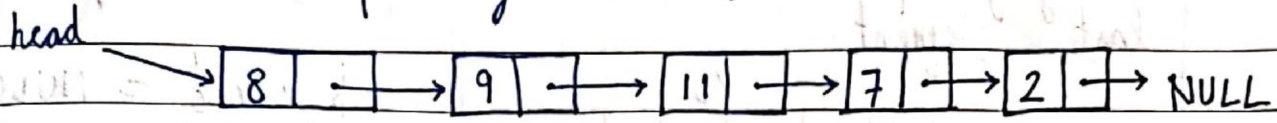
Operations on a circular Linked List

operations on a circular linked lists can be performed exactly like a singly linked list.

Visit www.codewithharry.com for practice sets/code/more

Deletion in a Linked List

Consider the following Linked List

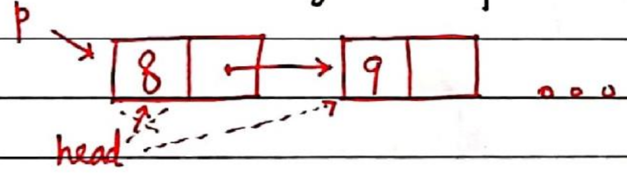


Deletion can be done for the following cases:

1. Deleting the first Node
2. Deleting the node at an index
3. Deleting the last Node
4. Deleting the first node with a given value.

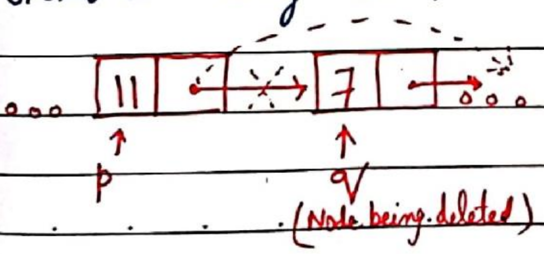
The deletion just like insertion is done by rewiring the pointer connections, the only caveat being: we need to free the memory of the deleted node using free().

Case 1: Deleting the first node



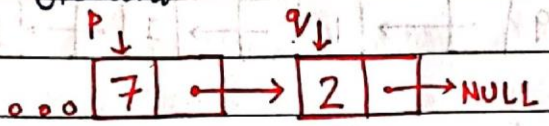
```
Struct Node * p = head
head = head → next;
free(p);
```

Case 2: Deleting the node at an index
for deleting a given node, we first bring a temporary pointer p before element to be deleted and q on the element being deleted



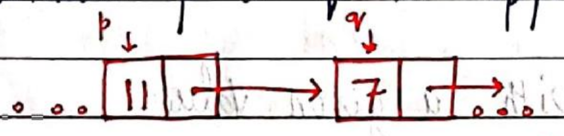
```
p → next = q → next;
free(q);
```

Case 3: Deleting the last Node
Last node can be deleted just like Case 2 by bringing p on second last element and q on last element.



p → next = NULL
free(q) → To free the memory!

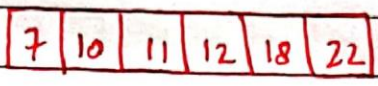
Case 4: Delete the first node with a given value
This can be done exactly like case 2 by bringing pointers p & q to appropriate positions



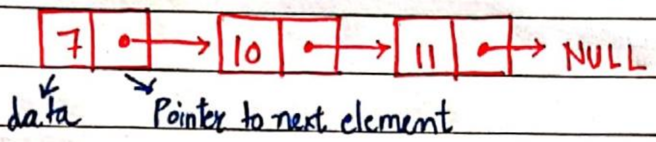
p → next = q → next;
free(q);

Introduction to Linked Lists

Linked lists are similar to arrays (Linear data structures)



⇒ In Arrays elements are Stored in Contiguous memory locations



⇒ In linked lists, elements are stored in non contiguous memory locations

Why Linked Lists?
Memory and the capacity of an array remains fixed.
In case of linked lists, we can keep adding and removing elements without any capacity constraints

Drawbacks of Linked Lists
→ Extra memory Space for pointers is required (for every node 1 pointer is needed).
→ Random access not allowed as elements are not Stored in contiguous memory locations.

Implementation
Linked list can be implemented using a Structure in C language

```
Struct Node {
        int data;
        Struct Node* next;        => Self refrencing Structure
        };
```