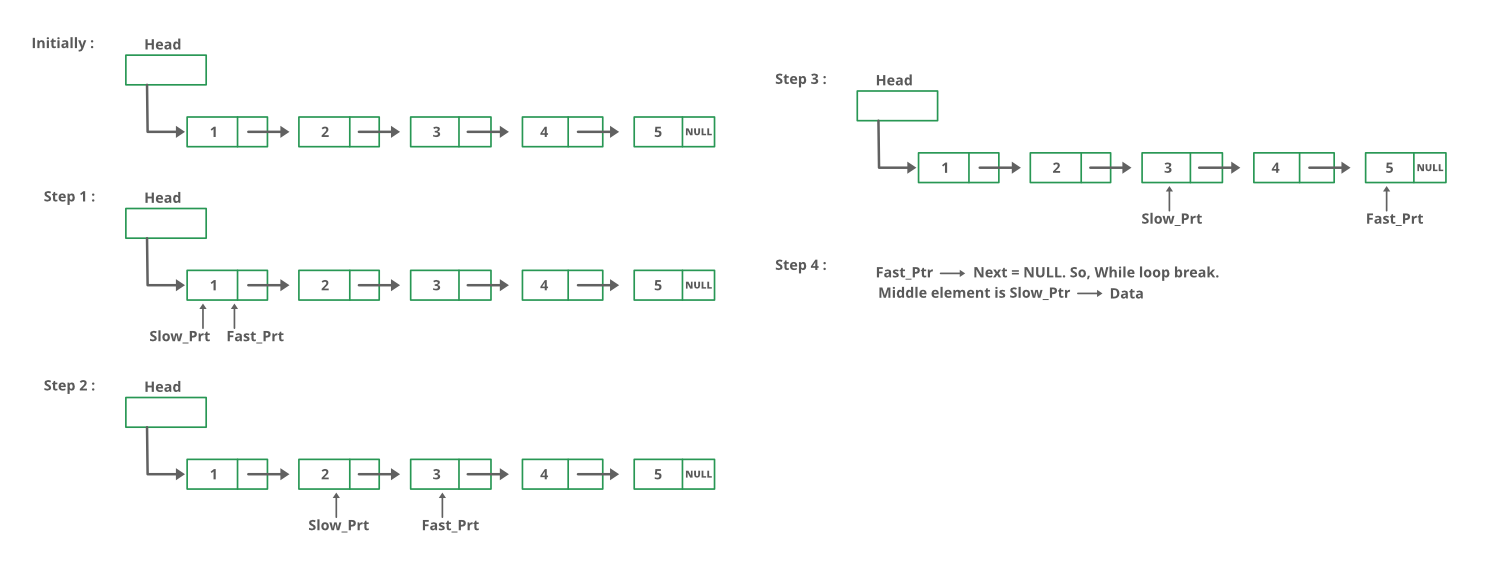
**Find the middle of a given linked list**

Given a singly linked list, find the middle of the linked list. For example, if the given linked list is 1->2->3->4->5 then the output should be 3.   
If there are even nodes, then there would be two middle nodes, we need to print the second middle element. For example, if given linked list is 1->2->3->4->5->6 then the output should be 4.

**Method 1:**   
Traverse the whole linked list and count the no. of nodes. Now traverse the list again till count/2 and return the node at count/2.

**Method 2:**   
Traverse linked list using two pointers. Move one pointer by one and the other pointers by two. When the fast pointer reaches the end slow pointer will reach the middle of the linked list.

Below image shows how printMiddle function works in the code:



void printMiddle(struct Node \*head)

{

    struct Node \*slow\_ptr = head;

    struct Node \*fast\_ptr = head;

    if (head!=NULL)

    {

        while (fast\_ptr != NULL && fast\_ptr->next != NULL)

        {

            fast\_ptr = fast\_ptr->next->next;

            slow\_ptr = slow\_ptr->next;

        }

        printf("The middle element is [%d]\n\n", slow\_ptr->data);

    }

}

Similarly – Program to delete middle element of list.

Q. Remove duplicate elements from sorted Linked List.