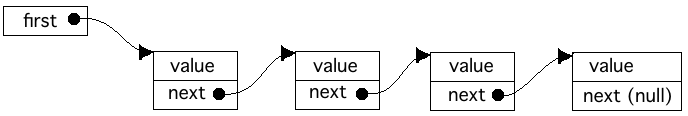
**DATA STRUCTURES AND ALGORITHM**

**HASSAAN AKBAR CHEEMA**

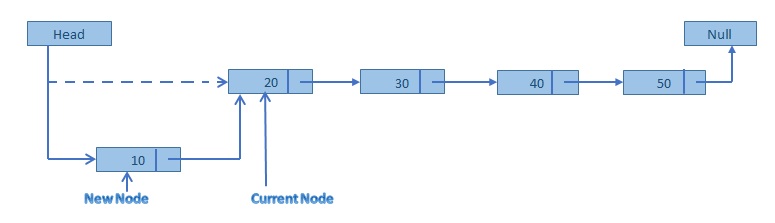
**174351**

**LAB5**

**LINK LIST**



INSERTION AT BEGINNING:



EXPLANATION: To insert at beginning follow the following steps:

1. point the newnode to the second node.
2. Point the head to the newnode.

PESUDOCODE:

Start

Create a new node

Enter data in newnode -> data

If head -> next = null

Then set head –> next = newnode

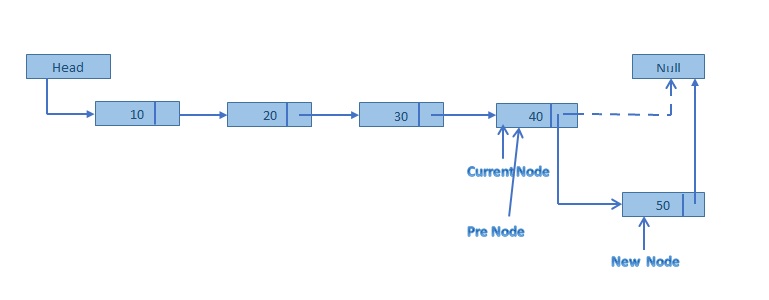
Else if head -> next != null

Then set newnode -> next = head i.e. the address of the second node

And set head -> next = newnode

End

INSERTION AT ENDING:



EXPLANATION: To insert at ending follow the following steps:

1. Traverse to the last node with the help of a pointer.
2. Point the end node to the newnode.
3. Point the newnode to the null.

PESUDOCODE:

Start

Create a new node

Enter data in newnode -> data

If head -> next = null

Then set head –> next = newnode

if head -> next != null

Then start a loop to reach the end node:

if ptr -> next = null

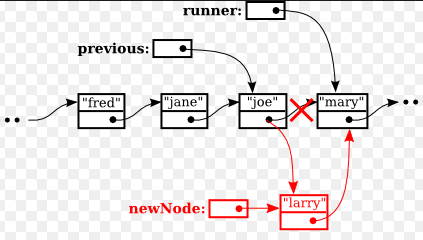
then set ptr -> next = newnode

else move ptr to the next node

end loop

end

UPDATION:



EXPLANATION: To insert at any point follow the following steps:

1. Traverse the link list with the help of two pointers.
2. One pointer will point to location of insertion node and one pointer is pointing to previous node.
3. Point newnode to the insertion node.
4. Point previous node to the newnode.

PESUDOCODE:

start

create a new node

Enter data in newnode -> data

Set newnode -> next = NULL

if position of insertion is 1

Then set newnode -> next = head

Set head = newnode

else

loop i from 0 to n-2

and traverse the link list

pointer = pointer -> next

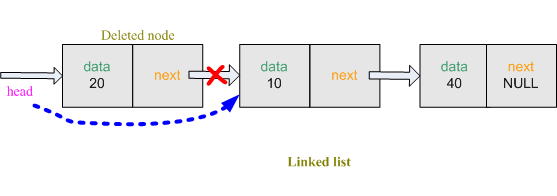
set newnode -> next = pointer -> next

move the pointer to the next position

by setting pointer = pointer -> next

end

DELETION AT BEGINNING:



EXPLANATION: To delete at beginning follow the following steps:

1. point the head to the third node.
2. Point the second node to the null.

PESUDOCODE:

Start

Make a pointer

Set pointer = head

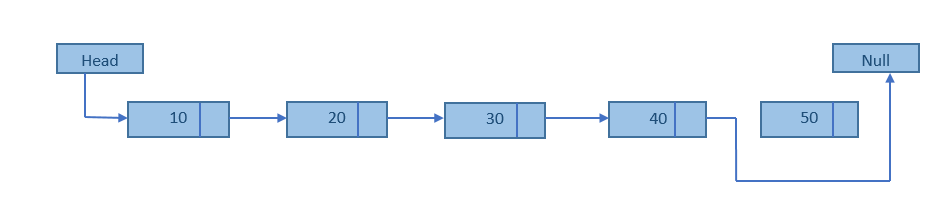
If pointer -> next != null

Then set head = pointer -> next

Else head = null

End

DELETION AT ENDING:



EXPLANATION: To delete at ending follow the following steps:

1. Traverse to the second last node with the help of a pointer.
2. Point the second last node to null.

PESUDOCODE:

Start

if head -> next != null

Then start a loop to reach the second last node:

if pointer -> next -> next = null

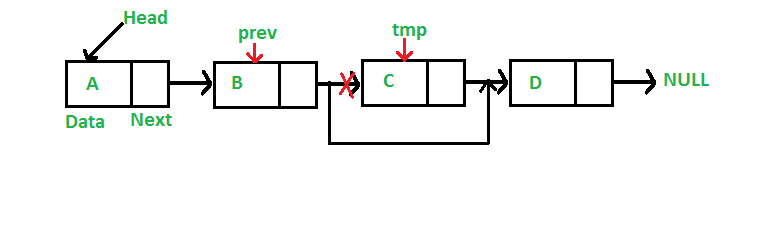
then set pointer -> next = null

else move ptr to the next node

end loop

end

DELETION AT ANY POINT:



EXPLANATION: To delete at any point follow the following steps:

1. Traverse the link list with the help of two pointers.
2. One pointer will point to location of deletion node and one pointer is pointing to the previous node.
3. Point previous node to the next node of deletion node.
4. Point deletion node to null.

PESUDOCODE:

Start

Make a pointer of node type

Set pointer = head

if position of deletion is 1

Then set head = pointer -> next -> next

Set pointer -> next = null

else

loop i from 0 to n-2

and traverse the link list

pointer = pointer -> next

set pointer -> next = pointer -> next -> next

move the pointer to the next position

by setting pointer = pointer -> next

end

COUNT:

EXPLANATION: To count the nodes follow the following steps:

1. Traverse the link list with the help of a pointer.
2. Increment at each traversal.

PESUDOCODE:

Start

Make a pointer of node type

set pointer = head;

traverse the link list until pointer = NULL

add 1 to the count at each traverse

move the pointer to the next node by setting pointer = pointer -> next

print the value of count

end

DISPLAY:

EXPLANATION: To count the nodes follow the following steps:

1. Traverse the link list with the help of a pointer.
2. Print node at each traversal.

PESUDOCODE:

Start

Make a pointer of node type

set pointer = head;

traverse the link list until pointer = NULL

print data of each node i.e. print pointer -> data

move the pointer to the next node by setting pointer = pointer -> next

end