Linked list algorithms

- لاحول ولا قوة الا بالله ولكنني مرغم على ذلك

بسم الله المعين.

Creat node

Create a node variable from the node structure.

Allocate memory for the node
Store data in the new node.

Append node

Create a new node.
Store data in the new node.
If there are no nodes in the list
    Make the new node the first node (head).
Else //the list is not empty
    Traverse the List to Find the last node.
    Add the new node to the end of the list.
End If.

Display List

Assign List head to node curr
While curr is not NULL
    Display the `value` *member of the curr.*
    Assign curr to its own next member.
End While.

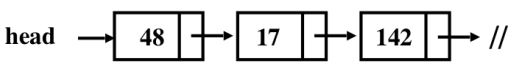
INSERTION

Insertion at the top

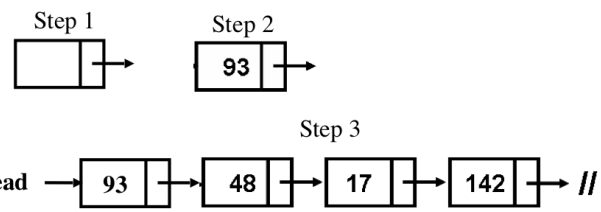
Steps:

- Create a Node (step 1)
- Set the node data Values (step 2)
- Connect the pointers (step 3)

Example



- Follow the previous steps and we get

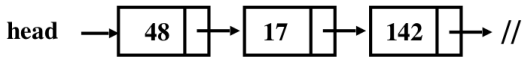


Insertion at the end

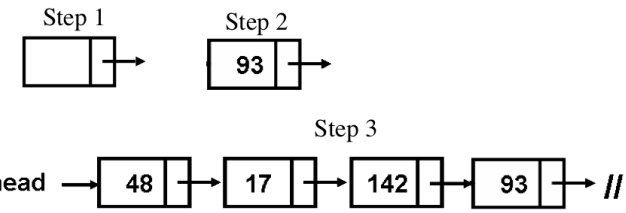
Steps:

- Create a Node
- Set the node data Values
- Connect the pointers

Example



- Follow the previous steps and we get

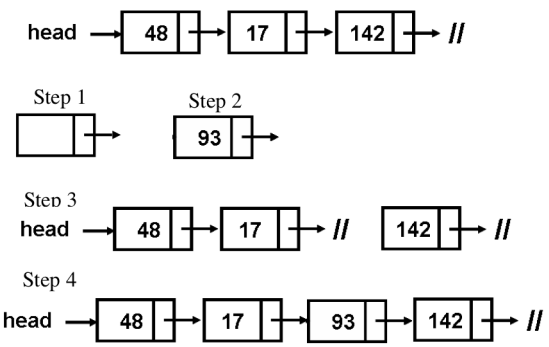


Insertion in the middle

Steps:

- Create a Node
- Set the node data Values
- Break pointer connection
- Re-connect the pointers

Example

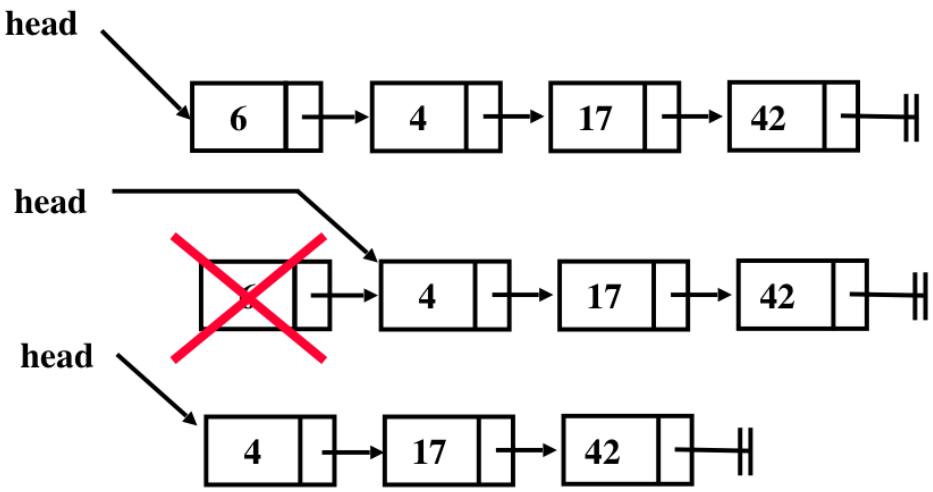


Insertion algorithm

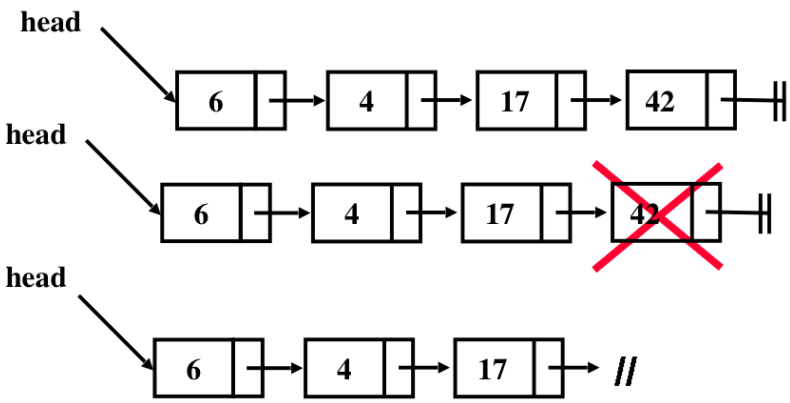
Create a new node.
Store data in the new node.
If there are no nodes in the list
    Make the new node the first node.
Else *//the list is not empty*
    Find the first node whose value is greater than or equal
    the new value, or the end of the list (whichever is first).
    Insert the new node before the found node, or at the end of
    the list if no node was found.
End If.

Deletion

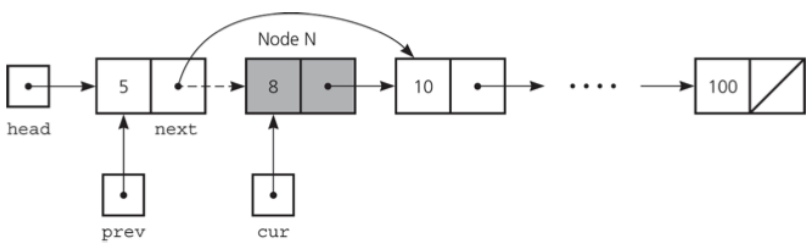
Deletion Top Node Description



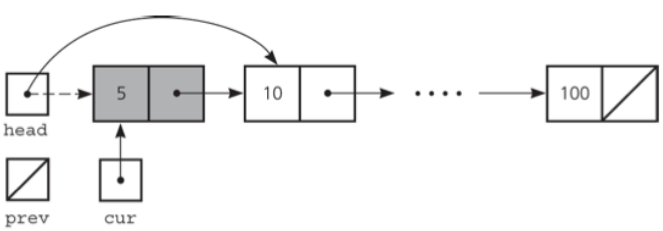
Deletion Last Description



Deletion Middle Node Description

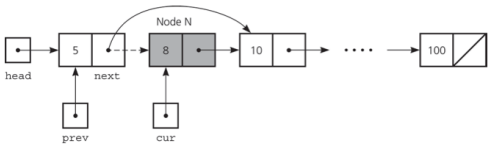


Deleting a node from a linked list

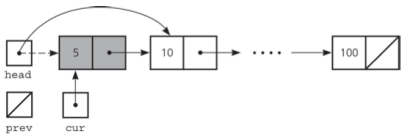


Deleting the first node

Deleting a Specified Node from a Linked List



Deleting a node from a linked list



Deleting the first node

DELETING STEPS

1 – Remove the node from the list **without breaking the links** created by the next pointers

2 – Deleting the node from memory