Page No.
Date

Assignment No. 7

Sumit Gulab Bhamare
SE Comp A08
Sub - DSL.

Aim- To understand & implement singly linked list.

Problem Definition: Department of Computer Engineering has student's club named 'Pinnacle Club'. Students of second, third & final years of department can be granted membership on request. Similarly, one may cancel the membership of club. First node is reserved for president of club & last node is reserved for secretary of club. Write C++ program to maintain club member's information using singly linked list. Store student PRN & Name. Write functions to.

a) Add & delete the members as well as president or even secretary.
b) Compute total number of members of club.
c) Display members
d) Display list in reverse order using recursion.
e) Two linked lists exist for two divisions. Concatenate two lists.

Learning objectives:-
To understand concept of sll (singly linked list)
To analyze the working of functions.

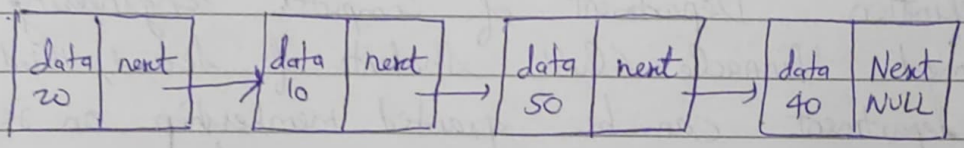
Learning outcome:-
Students will be able to analyze problems to use variants of linked list & solve various real-life problems.

Page No.
Date

Theory:

Linked list:

Linked list is one of the fundamental data structures, & can be used to implement other data structures. In a linked list there are different numbers of nodes. Each node consists of two fields. The first field holds the value or data & the second field holds the reference to the next node or null if the linked list is empty.

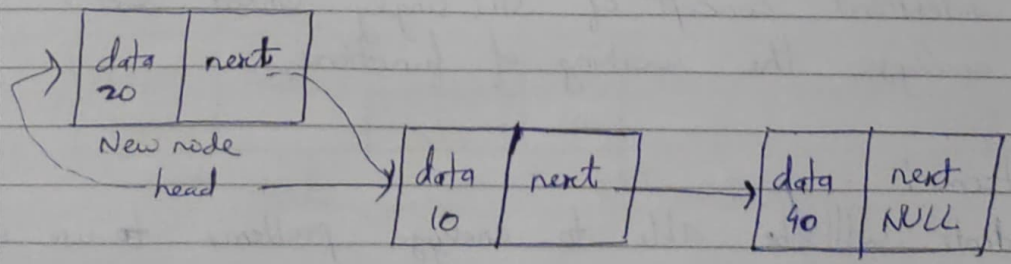


The singly - linked list is the easiest of the linked list, which has one link per node.

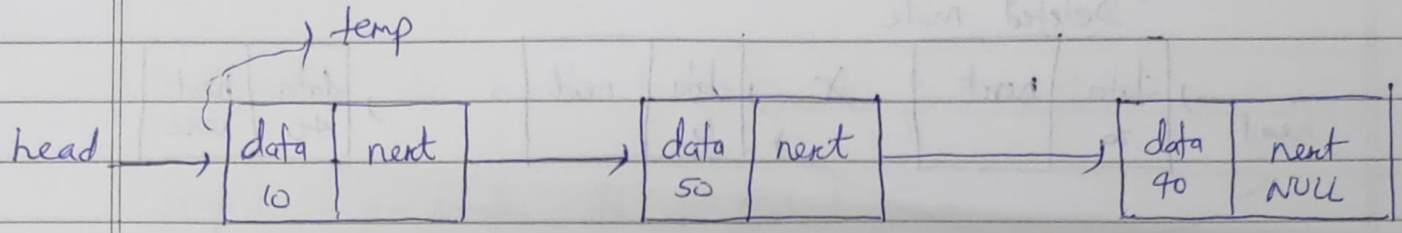
Linked list operation:

First, we create a structure "node". It has two members & first is int data which will store the information & second is node *next which will hold the address of the next node. Linked list structure is complete so now we will create linked list. We can insert data in the linked list from 'front' & at the same time from 'back'. Now we will examine how we can insert data from front in the linked list.

1.) Insert from front:

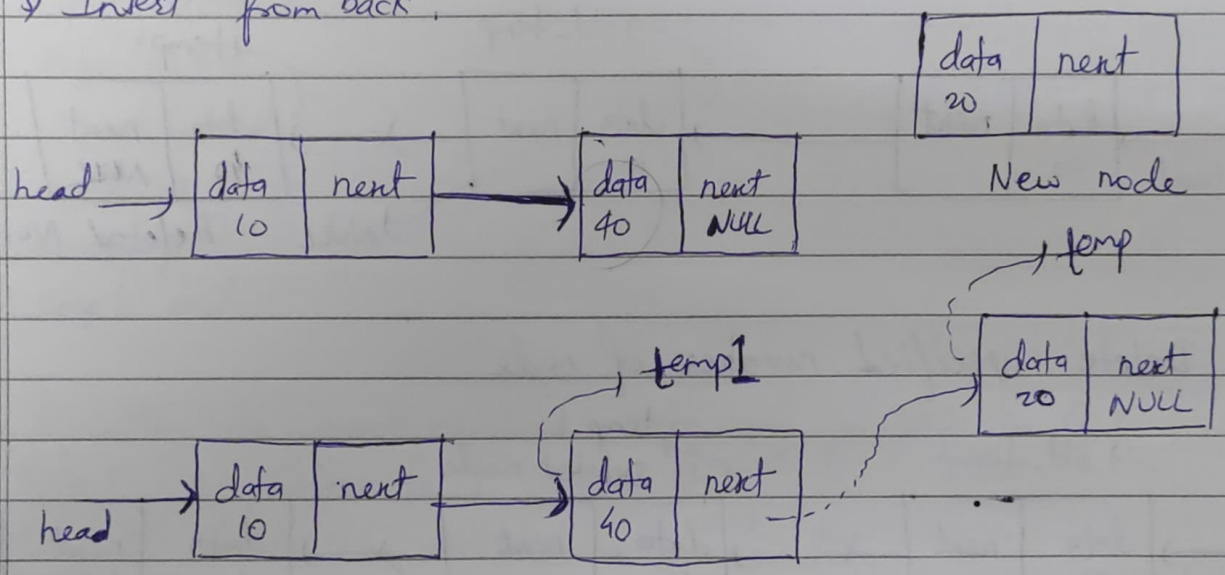


2) Traverse :-

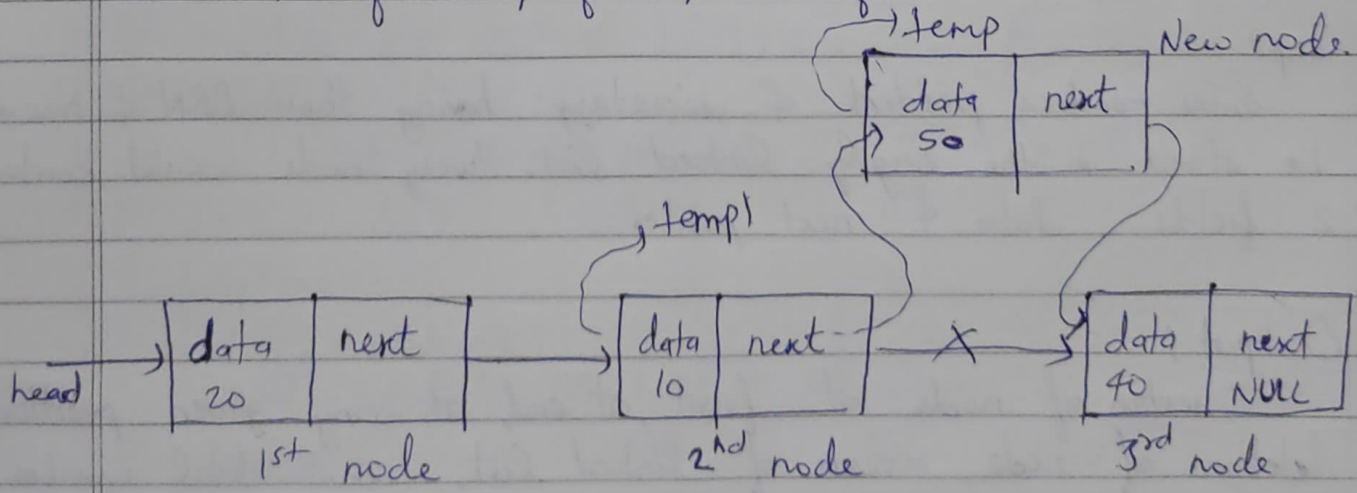


This process will run until the linked list's next is NULL.

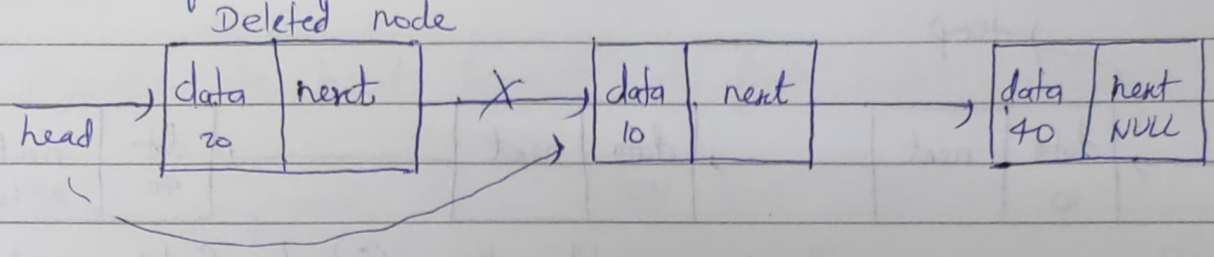
3) Invest from back :



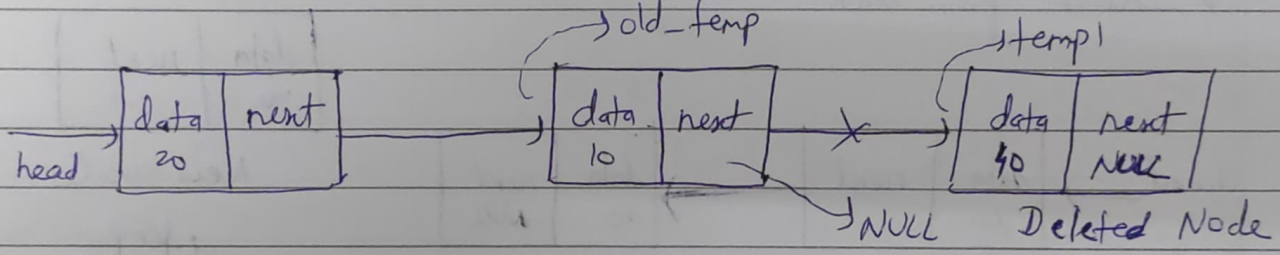
4) Invest after specified number of nodes.



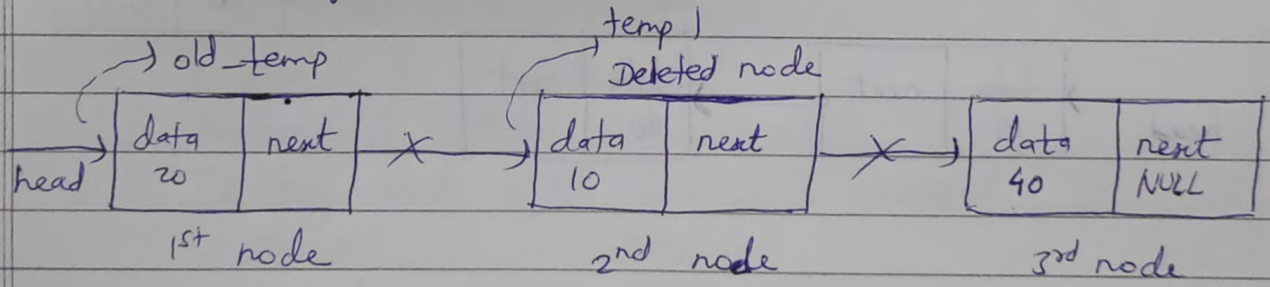
5) Delete from front:



6) Delete From back:



7) Delete specified number of node



Input:
Enter members, president & secretary having their PRN & Name to be stored in the singly linked list. Every node would contain 2 fields: data & next pointer.

Output:
Insertion of node at front, at end, at any given position, deletion of node, reverse of linked list, count total number of nodes, merge two linked list.

Page No.
Date

Algorithm:

. Algorithm to insert president at the first node:

Step 1: p - pointer to a linked list.

Step 2: e - element to be added.
Step 3: Getnode() returns a new node
q = Getnode()
info(q) = e
next(q) = p
p = q
return p
Step 4: end.

. Algorithm to insert members at any position in link list:
Step 1: p - pointer to a linked list
Step 2: Getnode() returns a new node
Step 3: x - key node after which e is inserted.
Step 4: e - element to be added.
k = p
while k ≠ NIL & info(k) = x do // find the key node
k = next(k) if k = NIL then
write "Node not found" return p
q = Getnode()
info(q) = e
next(q) = next(k)
next(k) = q
return p
Step 5: Stop.

Page No.
Date

· Algorithm to insert secretary at last node:
Step 1: p-pointer to a linked list
Step 2: Getnode() returns a new node
Step 3: e-element to be added
if p=NIL then
return p
k=p
while next(k) ≠ NIL do // find the last node k=next(k)
q=Getnode()
info(q)=e
next(k)=q
next(q)=NIL
return p
Step 4: end

Algorithm to Delete any member Node

Step 1: p-pointer to linked list
Step 2: x-key node to be deleted
Step 3: k & pred-temporary variables
Step 4: k=p; pred=NIL
while k≠NIL and info(k)=x do // find the key node pred=k
k=next(k)
if k=NIL then
write "Node not found" else
if pred = NIL // only one node in the list then
p=next(p) else
next(pred)=next(k)
return p
end
Delete Node
Step 5: end.

Page No.
Date

Algorithm to Display the members of link list.
Step 1: Create link list.
Step 2: Scan & print the entire link list of the members having their PRN and name one by one.
Step 3: end

Algorithm to Count the total number of member nodes in the link list.
Step 1: count=0
Step 2: Increment count as each node is traversed
current = head.
while (current != NULL) then
count++
Step 3: end.

. Algorithm to merge two linked list.
Step 1: enter two linked list.
Step 2: after first list is traverse merge the second linked list.
Step 3: end

Algorithm to reverse the link list.
Step 1: Iterative list reverse. Iterate through the list left-right. Move/ insert each node to the front of the result list like a Push of the node
Step 2: result = NULL;
Step 3: while (current != NULL)
next = current → next (note the next node current → next = result move the node onto the result)
result = current
current = next
Step 4: end

Page No.
Date

Software required : g++ / gcc compiler - / 64 bit fedora

Conclusion:-
We understand & implement different operations on of linked list.