Linked list operations Single Linked List

Insert at first, insert at last , insert at middle(random position), after specific node, in order.

Creation of a Single link list

```
1.  ptr=(struct node *)malloc(sizeof(struct node *));
2.  Set ptr->info=item
3.  Set start = ptr
4.  Initialize ch
5.  Read ch
6.  Repeat step no 5 to step no  11  while ch===‘y’
7.  cpt =(struct node *)malloc(sizeof(struct node *));
8.  Read cpt->info
9.  Ptr->next=cpt
10. Ptr=cpt
```

Continue…

11. Read ‘Y’ if you want to enter more nodes
12. ptr->next=NULL

Insert after specific node

```
1. ptr=(struct node *)malloc(sizeof(struct node ));
2. If(ptr==NULL) then
   Print overflow
   Exit
3. ptr->info=item
4. Initialize m  and *temp
5. Read m
6. Temp=start
7. Repeat step no 8 until temp->info!=m
8. temp=temp->next
```

Continue…

9. ptr->next=temp->next
10. Temp-> next=ptr

Insert in order

```
1. ptr=(struct node *)malloc(sizeof(struct node ));
2. If(ptr==NULL) then
   Print overflow
   Exit
3. ptr->info=item
4. If(ptr->info  < start->info)
   ptr->next=start
   start=ptr
   exit
5. temp=start
```

Continue…

```
6.  Repeat step no  6 to step no  8 while
    temp->info <= ptr->info
7.  If (temp->next==NULL) then
    prev=temp
    break
8.  Othewise
    Prev =temp
    temp=temp->next
9.  ptr->next=prev->next
10. Prev->next= ptr
```

Deletion of a node from linked list

- From first node
- From last node
- Middle (random)
- Specific node

Deletion of First node

```
1. If( start==NULL) then
   print "underflow"
   exit
2. Set ptr=start
3. if(ptr->next==NULL) then
   start=NULL
   printf   deleted element is ptr->info
   free(ptr)
   end if
4. Set ptr=start
```

Continue....

```
5. Set start= start->next
   printf element deleted is ptr->info
6. free(ptr)
```

Deletion of last node

```
1. If( start==NULL) then
   print "underflow"
   exit
2. Set ptr=start
3. if(ptr->next==NULL) then
   start=NULL
   printf   deleted element is ptr->info
   free(ptr)
   end if
4. Set temp=start
```

Continue…

5. Repeat step 6 and 7 while temp->next!=NULL
6. Prev=temp
7. Temp=temp->next
8. Set prev->next=NULL
9. Free(temp)

Deletion of an element from random position

```
1. If( start==NULL) then
   print “underflow”
   exit
2. Initialize loc,i=1,*temp
3. Read loc
4. Set temp=start
5. Repeat step no 6 to step no 8 until i<=loc
6. Prev=temp
7. Set temp=temp->next
```

Continue…

8. Set i=i+1
9. Prev->next=temp->next
10. Free(temp)

Deletion of a specific node

```
1.  If( start==NULL) then
    print “underflow”
    exit
2.  Initialize no  , *temp
    Read no, temp=start
3.  Repeat step no 4 to step no 5 while temp->info!=m
4.  If (temp->next !=NULL) then
    prev=temp
    temp=temp->next
5   Else
    print node is not found
```

Continue

6. Prev->next=temp->next
   print deleted node is temp->info
7. Free(temp)

Traversing of a Link List/Display of Link List

```
1. If Start==NULL then
       Write "list is empty"
       Exit
       End if
2. temp=start
3. Repeat step 3 to 5 while (temp!=NULL)
4. Print temp->info
5. temp=temp->info
6. stop
```