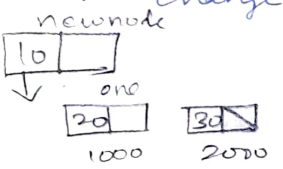
Insertion in Singly Linked list.

Insertion can be performed at various positions as follows:

* Insertion at the beginning
* Insertion at specific position
* Insertion at the end.

Insertion at the Beginning.

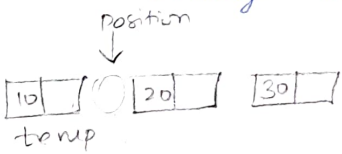
1. Allocate memory for a new node
2. Store data
3. Change the next of new node
4. Change head to point to recently created node




```
Struct node * newnode;
newnode = malloc (sizof (Struct node));
newnode -> data = 10;
newnode -> next = head;
head = newnode;
```

Insertion at the middle/Specific position.

1. Allocate memory & store data - newnode
2. Traverse to node just before the required position of new node.
3. Change the pointers.




```
Struct node *newnode;
newnode = malloc (sizeof (Struct node));
newnode -> data = 10;
Struct node *temp = head
for (int i = 2; i < position; i++)
{   if (temp -> next != NULL)
}           temp = temp -> next;
newnode -> next = temp -> next;
temp -> next = newnode;
```

Insertion at the end.

* Allocate memory for newnode
* Store the data in newnode
* Traverse to last node
* Change next of last node to newnode.

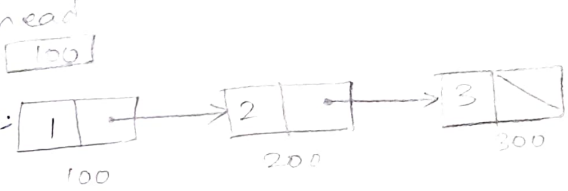
```
Struct node *newnode;
newnode = malloc (Struct node);
newnode -> data = 40;
newnode -> next = NULL;
Struct node *temp = head;
while (temp -> next != NULL)
{
    temp = temp -> next;
}
temp -> next = newnode;
```

Deletion in Singly Linked List.

Deletion can be performed at various positions as follows:

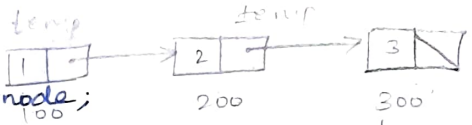
* Deletion at the beginning
* Deletion at end
* Deletion at mid position.

Deletion at the Beginning




```
if (head == NULL)
    pf ("cannot delete");
else
{Struct node *temp;
  temp = head;
  head = head -> next;
  free (temp);
}
```

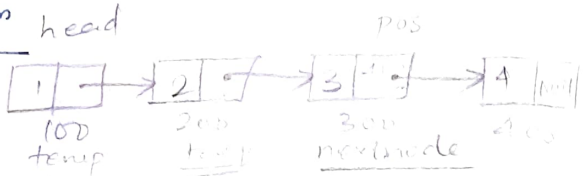
Deletion at end




```
Struct node *temp, *prenode;
temp = head;
if (temp -> next != null)
{
    premode = temp;
    temp = temp -> next;
}
premode -> next = null;
free (temp);
```

```
temp = head;
if (temp -> next != null)
{
premode = temp;
temp = temp -> next;
}
prenode -> next = null;
free (temp);
```

Deletion at Specific position




```
Struct node *temp;
temp = head;
for (i = 2; i < pos; i++)
{
    if (temp -> next != NULL)
        temp = temp -> next;
}
temp -> next = temp -> next -> next;
free (temp);
```

```
temp = head;
for (i = 2; i < pos; i++)
{
    temp = temp -> next;
}
nextnode = temp -> next;
temp -> next = nextnode -> next;
free (nextnode);
```

Traverse the elements of Linked list.

```
Struct node *temp = head;
printf ("\n The linked list elements are:");
while (temp -> next != NULL) {
        printf (" %d \t", temp -> data);
        temp = temp -> next;
}
```