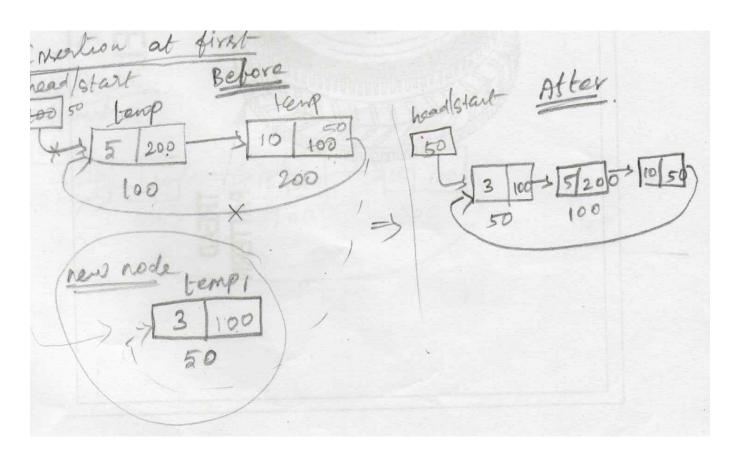
CREATION OF NODE IN CLL

```
void create(int)
if(head==null)
temp=(struct node *)malloc (size of (struct node));
printf("enter the elements");
scanf("%d",&temp->data);
temp->next=head // links the address field to NULL
head=temp;
else
temp1=(struct node *)malloc (size of (struct node));
printf("enter the elements");
scanf("%d",&temp1->data);
temp->next=temp1;
temp1->next=head;
temp=temp1;
        vealwa
                                  100
                                                              tempi
100
                 100
                                                                  100
                                                  200
                                                               200
                                                00
             100
head (stare
                                                       temps
 100
                                     300
                                                            100
                                                       300
               100
                                   200
```

Insertion at beginning

```
void insertfirst(int)
{
  head=temp;
  struct node *temp1;
  temp1=(struct node *)malloc (size of (struct node));
  printf("enter the elements");
  scanf("%d",&temp1->data);

while (temp->next! == head)
  {
    temp=temp->next;
  }
  temp->next=temp1;
  temp1->next=head;
  head=temp1
  }
```



```
insertion at end (similar to insertion at first except head=temp1)
void insertend(int)
head=temp;
struct node *temp1;
temp1=(struct node *)malloc (size of (struct node));
printf("enter the elements");
scanf("%d",&temp1->data);
while (temp->next! == head)
  {
    temp=temp->next;
}
  temp->next=temp1;
  temp1->next=head;
  }
invertion at e
headlstart Before
 100
                                    1-00
                               200
            100
                                           now node
                                                    (00
                                                 300
 100
                                                 100
                                 300
                                              300
              100
                               200
```

```
void DeleteFirst()
          temp=head;
          while(temp ->next!= head)
               temp = temp ->next;
          temp1=head;
          head=head->next;
          printf("\n The deleted node is -> %d",temp1->num);
          temp->next=head;
          free (temp1);
00
                                             300
                 100
 hoad start
  200
                                200
              300
          10
            200
```

```
void DeleteLast()
          temp=head;
          while(temp ->next!= head)
               temp1= temp;
               temp = temp ->next;
          }
          temp1->next= head;
          printf("\n The deleted node is : %d",temp->num);
          free(temp);
                                               temp
   100
                                 100
                                               300
                100
    headlstart
     100
                   200
```

100

Display:

```
void displayList()
{
    struct node *tmp;
    temp=head;
    if(head == NULL)
    {
        printf(" List is empty.");
    }
    else
    {
        while(temp->next != head)
        {
            printf("temp->data"); // prints the data of current node
            temp = temp->next; // advances the position of current node
        }
        printf("temp->data");
    }
}
```