//////////////////Function for making intersection/////////

void intersection(Node \*head1, Node \*head2, int pos)

{

    Node \*temp = head1;

    pos--;

    while (pos--)

    {

        temp = temp->next;

    }

    Node \*temp2 = head2;

    while (temp2->next != NULL)

    {

        temp2 = temp2->next;

    }

    temp2->next = temp;

}

int length(Node \*&head)

{

    int l = 0;

    Node \*temp = head;

    while (temp->next != NULL)

    {

        l++;

        temp = temp->next;

    }

    return l;

}

int detectIntersection(Node \*&head1, Node \*&head2)

{

    int l1 = length(head1);

    int l2 = length(head2);

    int d = 0;

    Node \*ptr1;

    Node \*ptr2;

    if (l1 > l2)

    {

         d = l1 - l2;

        ptr1 = head1;

        ptr2 = head2;

    }

    else

    {

         d = l2 - l1;

        ptr1 = head2;

        ptr2 = head1;

    }

    while (d)

    {

        ptr1 = ptr1->next;

        if (ptr1 == NULL)

        {

            return -1;

        }

        d--;

    }

    while (ptr1 != NULL && ptr2 != NULL)

    {

        if (ptr1 == ptr2)

        {

            return ptr1->data;

        }

        ptr1 = ptr1->next;

        ptr2 = ptr2->next;

    }

    return -1;

}

////////////// merge two sorted linked list////////////

Node\* merge(Node\* &head1,Node\* &head2){

    Node\* ptr1=head1;

        Node\* ptr2=head2;

        Node\* dummy= new Node(-1);

        Node\*ptr3=dummy;

        while(ptr1!=NULL && ptr2!=NULL){

            if(ptr1->data<ptr2->data){

                ptr3->next=ptr1;

                ptr1=ptr1->next;

            }

            else{

                ptr3->next=ptr2;

                ptr2=ptr2->next;

            }

            ptr3=ptr3->next;

        }

        while(ptr1!=NULL){

                ptr3->next=ptr1;

                ptr1=ptr1->next;

                ptr3=ptr3->next;

            }

            while(ptr2!=NULL){

                ptr3->next=ptr2;

                ptr2=ptr2->next;

                ptr3=ptr3->next;

            }

            return dummy->next;

}

**Merge two sorted linked list using recursion**

Node\* mergeRecursive(Node\* head1,Node\* head2){

    Node\*result;

    if(head1==NULL){

        return head2;

    }

    if(head2==NULL){

        return head1;

    }

    if(head1->data<head2->data){

        result=head1;

        result->next=mergeRecursive(head1->next,head2);

    }

        else{

            result=head2;

            result->next=mergeRecursive(head1,head2->next);

        }

    return result;

}