

```
#include<stdio.h>
#include<stdlib.h>
typedef struct linklist
{
    int data;
    struct node *next;
}Node;
Node *creatlist(Node *,int );
void insertEnd(Node **,int );
int checkdata(Node *,int );
void Display(Node *);
Node*head3=NULL;
int main()
{
    Node*head;
    printf("\nEnter Elements For List 1\n");
    Node *head1=creatlist(NULL,6);
    printf("\nEnter Elements For List 2\n");
    Node *head2=creatlist(NULL,6);
    Node *ptr1=head1,*ptr2=head2,*ptr,*pt;
    printf("List 1\n\n");
    Display(head1);
    printf("\nList 2\n\n");
    Display(head2);
    printf("\n\nCommon Element\n");
    while(ptr1!=NULL)
    {
        while(ptr2!=NULL)
        {
            if (ptr2->data==ptr1->data)
            {
                Node *node=(Node*)malloc(sizeof(Node));
                node->next=NULL;
                node->data=ptr1->data;
                if (head3==NULL)
                    head3=node;
                else
                {
                    ptr=head3;
                    while(ptr->next!=NULL)
                        ptr=ptr->next;
                    ptr->next=node;
                }
            }
            ptr2=ptr2->next;
        }
        ptr2=head2;
        ptr1=ptr1->next;
    }
    printf("\n");
    DeleteCommom(&head3);
}
```

```
printf("\n");
Display(head3);

}
Node *creatlist(Node *head,int co)
{
    Node * ptr=head;
while(co>0)
    {
        Node* node = (Node*)malloc(sizeof(Node));
        printf("\nEnter the element : ");
        scanf("%d",&node->data);
        node->next=NULL;
        if(head==NULL)
        {
            head=node;
            co--;
        }
        else
        {
            ptr=head;
            while(ptr->next!=NULL)
                ptr=ptr->next;
            ptr->next=node;
            co--;
        }
    }
    return head;
}
Display(Node *head)
{
    Node *ptr=head;
    while(ptr!=NULL)
    {
        printf("%d ",ptr->data);
        ptr=ptr->next;
    }
}
void DeleteCommom(Node **temp)
{
    Node *ptr1=*temp,*ptr2=*temp;
    while(ptr1!=NULL)
    {
        ptr2=ptr1->next;
        while(ptr2!=NULL)
        {
            if(ptr1->data==ptr2->data)
                ptr1->next=ptr2->next;
            ptr2=ptr2->next;
        }
        ptr1=ptr1->next;
    }
    free(ptr1);
    free(ptr2);
}
```

```
    return temp;  
}
```