#include <iostream>  
using namespace std;  
struct node  
{  
        int data;  
        node \*next,\*prev;  
};  
  
class dll  
{  
public:  
  node\*head1,\*head2;  
  node\*head3,\*move;  
dll()  
{  
        head1=NULL;  
        head2=NULL;  
        head3=NULL;  
}  
node\*create(node\*);  
void display(node\*);  
void sort(node\*);  
node\* merge(node\*,node\*);  
}b;  
  
node\*dll::create(node\*h)  
{  
        node \*temp;  
        int i,n;  
        cout<<"enter how many data you want";  
        cin>>n;  
        cout<<"enter data";  
        for(i=1;i<=n;i++)  
        {  
                temp=new node;  
                temp->next=NULL;  
                temp->prev=NULL;  
                cin>>temp->data;  
                if(h==NULL)  
                {  
                        h=temp;  
                        move=h;  
                }  
                else  
                {  
                        move->next=temp;  
                        temp->prev=move;  
                        move=temp;  
                }  
        }  
return(h);  
}  
  
void dll::display(node \*h)  
{  
        node \*temp;  
        temp=h;  
        while(temp!=NULL)  
        {  
                cout<<temp->data<<"\n";  
                temp=temp->next;  
        }  
}  
  
/\*void dll::sort(node\*h)  
{  
node\*p,\*q;  
for(p=h;p!=NULL;p=p->next)  
{  
for(q=p->next;q!=NULL;q=q->next)  
{  
if(p->data>q->data)  
{  
p->data=p->data+q->data;  
q->data=p->data-q->data;  
p->data=p->data-q->data;  
}  
}  
}  
b.display(h);  
}\*/  
  
node\* dll::merge(node\*h1,node\*h2)  
{  
  
        node\*newnode;  
        node\*temp1,\*temp2,\*temp3;  
        temp1=h1;  
        temp2=h2;  
  
  
        while(temp1!=NULL&&temp2!=NULL)  
        {  
                newnode=new node;  
                newnode->next=NULL;  
                newnode->prev=NULL;  
  
                if(temp1->data<temp2->data)  
                {  
                        newnode->data=temp1->data;  
                        if(head3==NULL)  
                        {  
                                head3=newnode;  
                                temp3=head3;  
                        }  
                        else  
                        {  
                                temp3->next=newnode;  
                                newnode->prev=temp3;  
                                temp3=newnode;  
                        }  
                        temp1=temp1->next;  
                }  
                else  
                {  
                        newnode->data=temp2->data;  
                        if(head3==NULL)  
                        {  
                                        head3=newnode;  
                                        temp3=head3;  
                        }  
                        else  
                        {  
                                        temp3->next=newnode;  
                                        newnode->prev=temp3;  
                                        temp3=newnode;  
                        }  
                        temp2=temp2->next;  
                }  
        }  
  
while(temp1!=NULL)  
{  
        newnode=new node;  
        newnode->next=NULL;  
        newnode->prev=NULL;  
  
        newnode->data=temp1->data;  
        if(head3==NULL)  
        {  
                        head3=newnode;  
                        temp3=head3;  
        }  
        else  
        {  
                        temp3->next=newnode;  
                        newnode->prev=temp3;  
                        temp3=newnode;  
        }  
        temp1=temp1->next;  
}  
  
while(temp2!=NULL)  
{  
        newnode=new node;  
                newnode->next=NULL;  
                newnode->prev=NULL;  
  
                newnode->data=temp2->data;  
                if(head3==NULL)  
                {  
                                head3=newnode;  
                                temp3=head3;  
                }  
                else  
                {  
                                temp3->next=newnode;  
                                newnode->prev=temp3;  
                                temp3=newnode;  
                }  
                temp2=temp2->next;  
  
}  
  
return (head3);  
}  
int main()  
{  
b.head1=b.create(b.head1);  
cout<<"\n";  
b.head2=b.create(b.head2);  
cout<<"\n";  
cout<<"1st linked list is";  
b.display(b.head1);  
cout<<"\n";  
cout<<"2st linked list is";  
b.display(b.head2);  
cout<<"\n";  
//cout<<"sorted 1st link list is";  
//b.sort(b.head1);  
cout<<"\n";  
//cout<<"sorted 2st link list is";  
//b.sort(b.head2);  
cout<<"\n";  
cout<<"merged link list is";  
b.head3=b.merge(b.head1,b.head2);  
b.display(b.head3);  
return 0;  
}