merge_two_linklist.h

```c
struct node
{
int data;
struct node *next;
};
struct node *head=NULL,*ptr=NULL,*nw=NULL,*temp=NULL,*head4=NULL,*ptr1=NULL;



void create()
{
int n,i,x;
printf("\n How many values :- ");
scanf("%d",&n);
printf("\n Enter actual values :-");
for(i=0;i<n;i++)
{
nw=(struct node *)malloc(sizeof(struct node));
scanf("%d",&x);
nw->data=x;
if(head==NULL)
{
head=nw;
ptr=nw;
}
else
{
```

```c
ptr->next=nw;

ptr=ptr->next;

}

}

}


struct node *create1(struct node *head)

{

int n,i,x;

printf("\n How many values :- ");

scanf("%d",&n);

printf("\n Enter actual values :-");

for(i=0;i<n;i++)

{

nw=(struct node *)malloc(sizeof(struct node));

scanf("%d",&x);

nw->data=x;

if(head==NULL)

{

head=nw;

ptr=nw;

}

else

{

ptr->next=nw;

ptr=ptr->next;

}

}

return(head);
```

```c
printf("\n The second link list is \n");

ptr=head;
do
{
printf("%d \t",ptr->data);
ptr=ptr->next;
}
while(ptr!=NULL);
}

void display(struct node *head)
{
ptr=head;
do
{
printf("%d \t",ptr->data);
ptr=ptr->next;
}
while(ptr!=NULL);
}

void insert()
{
```

```c
int pos,count=1,x;

printf(" \n Enter position \n");

scanf("%d",&pos);

ptr=head;

nw=(struct node *)malloc(sizeof(struct node));

printf("\n Enter the value to insert :- ");

scanf("%d",&x);

nw->data=x;

nw->next=NULL;

if(pos==1)

{

nw->next=head;

head=nw;

}

else

{

while(count!=pos-1)

{

ptr=ptr->next;

count++;

}nw->next=ptr->next;

ptr->next=nw;

}

}



void delete()

{

int pos,count=1;
```

```c
ptr=head;

printf("\n Enter position \n");

scanf("%d",&pos);

if(pos==1)

{

temp=head;

printf("\n The deleted element is %d",temp->data);

head=temp->next;

head=head->next;

free(temp);

}

else

{

while(count!=pos-1)

{

ptr=ptr->next;

count++;

}

if(ptr!=NULL)

{

temp=ptr->next;

printf("\n The deleted element is %d",temp->data);

ptr->next=temp->next;

free(temp);

}

else

{

printf("\n The Position is out of order");

}
```

```c
}

}


void search()

{

int key,flag=0;

printf("\n Enter the value to be search \n");

scanf("%d",&key);

ptr=head;

while(ptr!=NULL)

{

if(ptr->data==key)

{

flag=1;

break;

}

else

{

ptr=ptr->next;

}

}

if(flag==1)

printf("The element is found \n");

else

printf("The element is not found \n");

}
```