

singly_link_list.h

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

void display()
{
ptr=head;
do
{
printf("%d \t",ptr->data);
ptr=ptr->next;
}
while(ptr!=NULL);
}
void insert()
{
int pos,count=1,x;
printf("\n Enter the position : ");
scanf("%d",&pos);
ptr=head;
nw=(struct node*)malloc(sizeof(struct node));
printf("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;
ptr=head;
printf("Enter the position : ");
scanf("%d",&pos);
if(pos==1)
{
temp=head;
printf("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 position out of order");
}
}
}

```

```
void search()
{
int key,flag=0,x;
printf("\n Enter the value searched : ");
scanf("%d",&key);
ptr=head;
while(ptr!=NULL)
{
if(ptr->data==key)
{
flag=1;
break;
}
else
{
ptr=ptr->next;
}
}
if(flag==1)
printf("\n The element is found \n");
else
printf("\n The element not found \n");
}
```