

```

#include<stdio.h>
void create();
void insert();
void delet();
void display();
struct node
{
int data;
struct node *llink;
struct node *rlink;
}*header,*neww,*ptr,*ptr1,*ptr2;
void main()
{
int ch;
header=NULL;
header=(struct node *)malloc(sizeof(struct node));
while(1)
{
printf("\nEnter choice :1.create,2.insert,3.delete,4.display");
scanf("%d",&ch);
    switch(ch)
    {
        case 1:create();break;
        case 2:insert();break;
        case 3:delet();break;
        case 4:display();break;
        default:exit(0);
    }
}
}
void create()
{
int x;
printf("Enter data to insert:");
scanf("%d",&x);
neww=(struct node *)malloc(sizeof(struct node));
ptr=header;
    if(ptr->rlink==NULL)
    {
        ptr->rlink=neww;
        neww->rlink=NULL;
        neww->llink=ptr;
        neww->data=x;
    }
    else
    {
        while(ptr->rlink!=NULL)
        {

```

```

        ptr=ptr->rlink;
    }
    ptr->rlink=neww;
    neww->rlink=NULL;
    neww->llink=ptr;
    neww->data=x;
}
}
void insert()
{
    int x,pos,key;
    printf("Enter position 1.Begin,2.End,3.Any");
    scanf("%d",&pos);
    printf("Enter data to insert");
    scanf("%d",&x);
    neww=(struct node *)malloc(sizeof(struct node));
    if(neww==NULL)
    {
        printf("Unable to insert");
    }
    else if(pos==1)
    {
        ptr=header->rlink;
        neww->rlink=ptr;
        neww->llink=header;
        header->rlink=neww;
        neww->data=x;
    }
    else if(pos==2)
    {
        ptr=header;
        while(ptr->rlink!=NULL)
        {
            ptr=ptr->rlink;
        }
        ptr->rlink=neww;
        neww->llink=ptr;
        neww->rlink=NULL;
        neww->data=x;
    }
    else if(pos==3)
    {
        ptr=header;
        printf("Enter key:");
        scanf("%d",&key);
        while(ptr->rlink!=NULL&&ptr->data!=key)
        {
            ptr=ptr->rlink;

```

```

    }
    if(ptr->rlink==NULL)
    {
        printf("Key value not found");
    }
    else
    {
        ptr1=ptr->rlink;
        neww->rlink=ptr1;
        ptr1->llink=neww;
        neww->llink=ptr;
        ptr->rlink=neww;
        neww->data=x;
    }
}
else
{
    printf("Invalid position");
}
}
void delet()
{
    int num,pos;
    printf("Enter position 1.Begin, 2.End, 3.Any");
    scanf("%d",&pos);
    ptr=header;
    if(ptr->rlink==NULL)
    {
        printf("List is Empty");
    }
    else if(pos==1)
    {
        ptr=header->rlink;
        ptr1=ptr->rlink;
        header->rlink=ptr1;
        ptr1->llink=header;
        printf("Deleted node is %d",ptr->data);
    }
    else if(pos==2)
    {
        while(ptr->rlink!=NULL)
        {
            ptr1=ptr;
            ptr=ptr->rlink;
        }
        ptr1->rlink=NULL;
        printf("Deleted node is %d",ptr->data);
    }
}

```

```

else if(pos==3)
{
printf("Enter data to delete:");
scanf("%d",&num);
    while(ptr->rlink!=NULL && ptr->data!=num)
    {
        ptr=ptr->rlink;
    }
ptr1=ptr->llink;
ptr2=ptr->rlink;
ptr1->rlink=ptr2;
ptr2->llink=ptr1;
printf("Deleted node is %d",ptr->data);
}
else
{
printf("Invalid position");
}
}
void display()
{
printf("Elements in the list are:");
ptr=header;
while(ptr->rlink!=NULL)
{
ptr=ptr->rlink;
printf("%d ",ptr->data);
}
}

```