```
/*Singly LinkedList –Creation and display*/
#include<stdio.h>
struct node
{
int info;
struct node *next;
}*root;
void main()
{
 struct node *nptr,*tptr;
 int con;
 root=NULL;
 printf("Linked List Creation\n");
 while(con!=0)
 {
 nptr = (struct node *)malloc(sizeof(struct node));
 printf("Enter new value");
 scanf("%d",&nptr->info);
 nptr->next=NULL;
 if (root==NULL)
  tptr=root=nptr;
 else
 {
  tptr->next=nptr;
  tptr=tptr->next;
 }
 printf("Want to add more nodes(1 for Yes, 0 for No");
 scanf("%d",&con);
 }
 printf("\n");
 tptr=root;
 while(tptr!=NULL)
 {
  printf("%d\t",tptr->info);
  tptr=tptr->next;
 }
 getch();
}
```

```
/*Operations on  Singly Linked List*/
#include<stdio.h>
#include<conio.h>
struct node
{
int info;
struct node *link;
}*root,*prev,*next;
int tn=0;

void main()
{
 int choice;
 root=NULL;
 while(1)
 {
   clrscr();
   printf("Linklist Operations\n");
   printf("1.Insert Begin\n");
   printf("2.Insert End\n");
   printf("3.Insert Middle\n");
   printf("4.Delete\n");
   printf("5.Display\n");
   printf("6.Quit\n");
   printf("Enter your choice : ") ;
   scanf("%d", &choice);
   switch(choice)
  {
   case 1:
     insbegin();
     break;
   case 2:
     insend();
     break;
```

```
    case 3:
      insmid();
      break;
    case 4:
      del();
      break;
    case 5:
      disp();
      break;
    case 6:
      exit(0);
    default :
      printf("Wrong choice\n");
   }/*End of switch */
   getch();
 }/*End of while */
}/*End of main() */

void insbegin()
{
 struct node *nptr;
 int item;
 nptr = (struct node *)malloc(sizeof(struct node));
 printf("Enter new value");
 scanf("%d",&item);
 nptr->info=item;
 nptr->link=root;
 root=nptr;
 tn++;
}
```

```
void insend()
{
 struct node *nptr,*tptr;
 int item;
 nptr = (struct node *)malloc(sizeof(struct node));
 printf("Enter new value");
 scanf("%d",&item);
 nptr->info=item;
 nptr->link=NULL;
 if (root==NULL)
  root=nptr;
 else
 {
  tptr=root;
  while(tptr!=NULL)
  { prev=tptr;
    tptr=tptr->link;
  }
  prev->link=nptr;
 }
 tn++;
}


void insmid()
{
 struct node *nptr,*tptr;
 int pos,c=0;
 nptr = (struct node *)malloc(sizeof(struct node));
 printf("enter position between 1 and %d ",tn);
 scanf("%d",&pos);
 if (pos>=1 && pos<=tn)
  {
   printf("Enter new value");
```

```
  scanf("%d",&nptr->info);
  nptr->link=NULL;
  tptr=prev=root;
  while(tptr!=NULL)
  { c++;
    if (pos==c)
    {  prev->link=nptr;
          nptr->link=tptr;
          tn++;
          break;
     }
    else
    {
     prev=tptr;
     tptr=tptr->link;
    }
   }
  }
}

void del()
{
 struct node *tptr;
 int item;
 printf("Enter value");
 scanf("%d",&item);
 tptr=prev=root;
 while(tptr!=NULL)
 { if (tptr->info==item)
    {
     prev->link=tptr->link;
     tn--;
     break;
    }
```

```
  prev=tptr;
  tptr=tptr->link;
 }
}


void disp()
{ struct node *nptr;
  nptr=root;
 if(nptr==NULL)
    printf("No elements\n");
 else
 {
   printf("\nElements :\n");
   while(nptr!= NULL)
   {
    printf("%d\n",nptr->info);
    nptr = nptr->link;
  }
 }
}
```

```
/*Stack Operation using Linked List */

#include<stdio.h>
#include<conio.h>

struct node
{
int info;
struct node *link;
} *top=NULL;

void main()
{
 int choice;
 while(1)
 { printf("1.Push\n");
   printf("2.Pop\n");
   printf("3.Display\n");
   printf("4.Quit\n");
   printf("Enter your choice : ") ;
   scanf("%d", &choice);
   switch(choice)
  {
   case 1:
     push();
     break;
   case 2:
     pop();
     break;
   case 3:
    display();
    break;
   case 4:
     exit(0);
```

```
    default :
      printf("Wrong choice\n");
   }/*End of switch */
 }/*End of while */
}/*End of main() */


void push()
{
 struct node *tmp;
 int item;
 tmp = (struct node *)malloc(sizeof(struct node));
 printf("Input the new value to be pushed on the stack");
 scanf("%d",&item);
 tmp->info=item;
 tmp->link=top;
 top=tmp;
}/*End of push()*/


void pop()
{
 struct node *tmp;
 if(top == NULL)
   printf("Stack is empty\n");
 else
 { tmp=top;
    printf("Popped item is %d\n",tmp->info);
    top=top->link;
    free(tmp);
 }
}/*End of pop()*/
```

```
void display()
{ struct node *ptr;
  ptr=top;
 if(top==NULL)
    printf("Stack is empty\n");
 else
 {
   printf("Stack elements :\n");
   while(ptr!= NULL)
   {
    printf("%d\n",ptr->info);
    ptr = ptr->link;
  }/*End of while */
 }/*End of else*/
}/*End of display()*/
```

/*Queue Operation using Linked List */

```
#include<stdio.h>
#include<conio.h>

struct node
{
int info;
struct node* next;
}*front,*rear;

void enqueue(int elt);
int dequeue();
void display();

void main()
{
 int ch,elt;
 rear=NULL;
 front=NULL;
 clrscr();
 while(1)
 {
  printf("\nEnter:\n1->Insert\n2->Delete\n
  3- >Display\n4->Exit\n");
  scanf("%d",&ch);
  switch(ch)
  {
   case 1:
     printf("Enter The Element Value\n");
     scanf("%d",&elt);
     enqueue(elt);
     break;
   case 2:
```

```
      elt=dequeue();
      printf("The deleted element = %d\n",elt);
      break;
   case 3:
     display();
     break;
   default:
     printf("~~~Exit~~~");
     getch();
     exit(0);
  }
 }
}


void enqueue(int elt)
{
 struct node *p;
 p=(struct node*)malloc(sizeof(struct node));
 p->info=elt;
 p->next=NULL;
 if(rear==NULL||front==NULL)
  front=p;
 else
  rear->next=p;
  rear=p;
}


int dequeue()
{
 struct node *p;
 int elt;
 if(front==NULL||rear==NULL)
```

```
 {
  printf("\nUnder Flow");
  getch();
  exit(0);
 }
 else
 {
  p=front;
  elt=p->info;
  front=front->next;
  free(p);
 }
 return(elt);
}


void display()
{
 struct node *t;
 t=front;
 while(front==NULL||rear==NULL)
 {
  printf("\nQueue is empty");
  getch();
  exit(0);
 }
 while(t!=NULL)
 {
  printf("->%d",t->info);
  t=t->next;
 }
}
```

```
/*Circular queue using  Linked List*/

#include<stdio.h>

struct node
{
int info;
struct node *link;
}*root;

void main()
{
 struct node *nptr,*tptr;
 int con;
 root=NULL;

 printf("Create Nodes\n");
 while(con!=0)
 {
 nptr = (struct node *)malloc(sizeof(struct node));
 printf("Enter new value");
 scanf("%d",&nptr->info);
 nptr->link=root;
 if (root==NULL)
  tptr=root=nptr;
 else
 {
  tptr->link=nptr;
  tptr=tptr->link;
 }
 printf("Want to add more nodes 1 for Yes, 0 for No");
 scanf("%d",&con);
 }
```

```
printf("\nLinked List\n");
 tptr=root;
 printf("\n%d\t",tptr->info);
 tptr=tptr->link;
 while(tptr!=root)
 {
  printf("%d\t",tptr->info);
  tptr=tptr->link;
 }
 getch();
}
```

/*Doubly Linked List - Creation, forward and backward display*/

```
#include<stdio.h>

struct node
{
int info;
struct node *prev;
struct node *next;
}*root;

void main()
{
 struct node *nptr,*tptr,*last;
 int con;
 root=NULL;
 printf("Linked List Creation\n");
 while(con!=0)
 {
 nptr = (struct node *)malloc(sizeof(struct node));
 printf("Enter new value");
```

```
scanf("%d",&nptr->info);
nptr->prev=nptr->next=NULL;
if (root==NULL)
 tptr=root=nptr;
else
{
 tptr->next=nptr;
 nptr->prev=tptr;
 tptr=tptr->next;
}
printf("Want to add more nodes 1 for Yes, 0 for No");
scanf("%d",&con);
}

printf("\nForward Display:-  ");
tptr=root;
while(tptr!=NULL)
{
 printf("%d\t",tptr->info);
 last=tptr;
 tptr=tptr->next;
}

printf("\nBackward Display:-  ");
while(last!=NULL)
{
 printf("%d\t",last->info);
 last=last->prev;
}
getch();
}
```