```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void insertf();
void insertl();
void insertpos();
void deletef();
void deletel();
void deletepos();
void display();
struct node
{
    int data;
    struct node *link;
};
struct node *head;
void main()
     {
     int ch;
     clrscr();
     while(ch!=8)
     {
     printf("\n1.Insert First\n2.Insert Last\n3.Insert By Position");
     printf("\n4.Delete First\n5.Delete Last\n6.Delete By Position");
     printf("\n7.Display\n8.Exit");
     printf("\nEnter the choice:");
     scanf("%d",&ch);
     switch(ch)
     {
    case 1:{
         insertf();
         break;
         }
    case 2:{
         insertl();
         break;
         }
    case 3:{
         insertpos();
         break;
         }
    case 4:{
         deletef();
         break;
         }
    case 5:{
         deletel();
         break;
         }
    case 6:{
         deletepos();
         break;
```

```
        }
    case 7:{
        display();
        break;
        }
    case 8:{
        printf("\nExiting");
        break;
        }
    default:printf("\nInvalid");
    }
    };
    getch();
}
void insertf()
{
 struct node *temp,*ptr;
 temp=(struct node*)malloc(sizeof(struct node));
 if(temp==NULL)
 {
  printf("\nUnable to insert:");
 }
 else
 {
 printf("\nEnter the element to insert:");
 scanf("%d",&temp->data);
 temp->data=temp->data;
 temp->link=head;
 head=temp;
 printf("\nElement insert at first");
 }
 }
void insertl()
{
    struct node *temp,*ptr;
     temp=(struct node *)malloc(sizeof(struct node));
     if(temp==NULL)
     {
         printf("Overflow\n");
         exit(0);
     }
     printf("Enter the element to insert:");
     scanf("%d",&temp->data);
     temp->link=NULL;
     if(head==NULL)
     {
         head=temp;
     }
     else
     {
         ptr=head;
         while(ptr->link!=NULL)
```

```
          {
           ptr=ptr->link;
          }
          ptr->link=temp;
      }
     }
void insertpos(int pos)
{
   int i;
   struct node *newNode,*temp;
   newNode=(struct node*)malloc(sizeof(struct node));
   if(newNode==NULL)
   {
    printf("\nUnable to insert");
   }
   printf("\nEnter the element to insert:");
   scanf("%d",&newNode->data);
    newNode->data=newNode->data;
    newNode->link=NULL;
    temp=head;
    printf("\nEnter the position to insert:");
    scanf("%d",&pos);
    for(i=2;i<=pos;i++)
    {
     temp=temp->link;
     if(temp==NULL)
      break;
     }
     if(temp!=NULL)
     {
     newNode->link=temp->link;
     temp->link=newNode;
     printf("\nElement inserted");
     }
     else
      printf("\nUnable to insert");
      }
void deletef()
{
 struct node *dele;
 if(head==NULL)
 {
  printf("\nList is empty");
  }
 else
 {
  dele=head;
  head=head->link;
  printf("\nElement deleted %d",dele->data);
  free(dele);
}
}
```

```
void deletel()
{
 struct node *dele,*prev;
 if(head==NULL)
 {
  printf("\nList is empty");
  }
 dele=head;
 prev=head;
 while(dele->link!=NULL)
 {
   prev=dele;
   dele=dele->link;
   }
   if(dele==head)
   {
   head=NULL;
   }
   else
   {
   prev->link=NULL;
   printf("\nElement deleted %d",dele->data);
   free(dele);
   }
}
void deletepos(int pos)
{
  int i;
  struct node *dele,*prev;
  if(head==NULL)
  {
   printf("\nList is empty");
  }
  dele=head;
  prev=head;
  printf("\nEnter the position to delete:");
  scanf("%d",&pos);
   for(i=2;i<=pos;i++)
   {
   prev=dele;
   dele=dele->link;
   if(dele==NULL)
    break;
   }
   if(dele!=NULL)
   {
    if(dele==head)
     head=head->link;
    prev->link=dele->link;
    dele->link=NULL;
    printf("\nElement deleted is %d",dele->data);
    free(dele);
```

```
    }
    else
      printf("\nInvalid position");
}
void display()
{
      struct node *ptr;
      if(head==NULL)
      {
            printf("Underflow\n");
      }
      else
      {
            ptr=head;
            while(ptr!=NULL)
            {
                  printf("%d\t",ptr->data );
                  ptr=ptr->link;
            }
      }
}
```

```
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:1

Enter the element to insert:3

Element insert at first
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:_
```

```
8.Exit
Enter the choice:1

Enter the element to insert:8

Element insert at first
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:7
8       7       4       3
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:
```

```
6.Delete By Position
7.Display
8.Exit
Enter the choice:2
Enter the element to insert:12

1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:7
8       7       4       3       12
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:
```

```
8.Exit
Enter the choice:2
Enter the element to insert:12

1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:7
8       7       4       3       12
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:3

Enter the element to insert:12_
```

```
Enter the element to insert:12

Enter the position to insert:3

Element inserted
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:7
8       7       4       12      3       12
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:
```

```
6.Delete By Position
7.Display
8.Exit
Enter the choice:7
8       7       4       12      3       12
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:4

Element deleted 8
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:_
```

```
7.Display
8.Exit
Enter the choice:4

Element deleted 8
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:5

Element deleted 12
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:_
```

```
Enter the choice:5

Element deleted 12
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:6

Enter the position to delete:4

Element deleted is 3
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:
```

```
8.Exit
Enter the choice:6

Enter the position to delete:4

Element deleted is 3
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:7
7       4       12
1.Insert First
2.Insert Last
3.Insert By Position
4.Delete First
5.Delete Last
6.Delete By Position
7.Display
8.Exit
Enter the choice:_
```