```
//Create DLL
#include<stdio.h>
#include<stdlib.h>
struct Node{
    int data;
    struct Node * next;
    struct Node * prev;
}*new, *head,*tail, *temp, * prevnode, * prev;

 void linkedListTraversal(struct Node *temp)
 {
      temp= head;
    do{
        printf("Element is %d\n", temp->data);
        temp=temp->next;
    }while(temp!=NULL);
}
  // create list
int main()
 {
 int value;
 char ch = 'y';
 printf("create a doubly  linked list\n");
while(ch =='y')
 {
  new = (struct Node *) malloc(sizeof(struct Node));
  printf("enter value to be inserted in linked list");
  scanf("%d",&value);
  new->data=value;
  if(head==NULL)
 {
    head=new;
    tail=new;
    head->next=NULL;
    head->prev=NULL;
 }
 else
   {
        temp=head;
      while(temp->next!=NULL)
     {
         prevnode=temp;
         temp=temp->next;
     }
        temp->next=new;
        new->next=NULL;
        new->prev=temp;
        tail=new;
       }
          fflush(stdin);
printf("Y- continue, N-exit");
scanf(" %c",&ch);
      }
   printf("Linked list \n");
    linkedListTraversal(head);
   return 0;
}
```

-x-x

Insert at beginning in the list

```
#include<stdio.h>
#include<stdlib.h>
struct Node{
    int data;
    struct Node * next;
```

```
   struct Node * prev;
}*new, *head,*tail, *temp, * prevnode;

 void linkedListTraversal(struct Node *temp)
 {
     temp= head;
   do{
      printf("Element is %d\n", temp->data);
      temp=temp->next;
   }while(temp!=NULL);
}

struct Node * InsertAtBegin( struct Node * head, int data)
  {
         struct Node * new = (struct Node *) malloc(sizeof(struct Node));
     new->data=data;
     new->next=head;
     head->prev=new;
     new->prev=NULL;
     head=new;
     return head;
   }
   // create list
int main()
 {
  int value;
 char ch = 'y';
 printf("create a doubly  linked list\n");
while(ch =='y')
 {
  new = (struct Node *) malloc(sizeof(struct Node));
  printf("enter value to be inserted in linked list");
  scanf("%d",&value);
  new->data=value;

  if(head==NULL)

 {

    head=new;

    tail=new;

    head->next=NULL;

    head->prev=NULL;

 }

 else

   {

        temp=head;

      while(temp->next!=NULL)

     {

         prevnode=temp;

         temp=temp->next;

     }

        temp->next=new;

        new->next=NULL;

        new->prev=temp;

        tail=new;

       }


    fflush(stdin);

printf("Y- continue, N-exit");

scanf(" %c",&ch);

      }

  printf("enetr value to be inserted\n");

      scanf("%d",&value);

      head=InsertAtBegin(head, value);


   printf("Linked list \n");

    linkedListTraversal(head);

   return 0;
```

```
}

-x-x-x


//Insert at end

#include<stdio.h>

#include<stdlib.h>

struct Node{

    int data;

    struct Node * next;

    struct Node * prev;

}*new, *head,*tail, *temp, * prevnode;


 void linkedListTraversal(struct Node *temp)

 {

      temp= head;

    do{

        printf("Element is %d\n", temp->data);

        temp=temp->next;

    }while(temp!=NULL);

}


struct Node * InsertAtEnd( struct Node * head, int data)

  {


      struct Node * new = (struct Node *) malloc(sizeof(struct Node));

      new->data=data;


     temp=head;

      while(temp->next!=NULL)

     {

         prevnode=temp;

         temp=temp->next;
     }
        temp->next=new;
        new->next=NULL;
        new->prev=temp;
        tail=new;
        return head;
       }
    // create list
int main()
 {
 int value;
 char ch = 'y';
 printf("create a doubly  linked list\n");
while(ch =='y')
 {
  new = (struct Node *) malloc(sizeof(struct Node));
  printf("enter value to be inserted in linked list\n ");
  scanf("%d",&value);
  new->data=value;
  if(head==NULL)
 {
    head=new;
    tail=new;
    head->next=NULL;
    head->prev=NULL;
 }
 else
   {
        temp=head;
      while(temp->next!=NULL)
     {
         prevnode=temp;
         temp=temp->next;
     }
        temp->next=new;
        new->next=NULL;
        new->prev=temp;
        tail=new;
       }
     fflush(stdin);
printf("Y- continue, N-exit \n");
scanf(" %c",&ch);
      }
  printf("enetr value to be inserted\n");
      scanf("%d",&value);
      head=InsertAtEnd(head, value);
     printf("Linked list \n");
    linkedListTraversal(head);
   return 0;
 }
```

-x-x-x

Insert at position

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

struct Node{
    int data;
    struct Node * next;
    struct Node * prev;
}*new, *head,*tail, *temp, * prevnode, * prev,  * temp1;

 void linkedListTraversal(struct Node *temp)
 {
      temp= head;
    do{
```

```
    printf("Element is %d\n", temp->data);
    temp=temp->next;
  }while(temp!=NULL);
}
struct Node * InsertAtPosition( struct Node * head, int data, int position)
 {
    int i=1;
   struct Node * new = (struct Node *) malloc(sizeof(struct Node));
   new->data=data;
       temp=head;
    while(i<position-1)
    {

      temp=temp->next;
      i++;
    }
    temp1=temp->next;
    temp->next=new;
    new->next=temp1;
    temp1->prev=new;
    new->prev=temp;
    return head;
 }

 // create list
int main()
 {

 int value, position;
 char ch = 'y';
 printf("create a doubly  linked list\n");
while(ch =='y')
 {
  new = (struct Node *) malloc(sizeof(struct Node));
  printf("enter value to be inserted in linked list\n");
  scanf("%d",&value);
  new->data=value;
  if(head==NULL)
 {
    head=new;
    tail=new;
    head->next=NULL;
    head->prev=NULL;
 }
 else
   {
        temp=head;
      while(temp->next!=NULL)
     {
         prevnode=temp;
         temp=temp->next;
     }
        temp->next=new;
        new->next=NULL;
        new->prev=temp;
        tail=new;
       }

    fflush(stdin);
printf("Y- continue, N-exit\n");
scanf(" %c",&ch);
      }
  printf("enetr position\n");
      scanf("%d",&position);

  printf("enetr value to be inserted\n");
```

```
      scanf("%d",&value);


      head=InsertAtPosition(head, value, position);

   printf("Linked list \n");

    linkedListTraversal(head);

   return 0;


}
```

-x-x-x

Delete at beginning

```
#include<stdio.h>

#include<stdlib.h>


struct Node{

    int data;

    struct Node * next;

    struct Node * prev;

}*new, *head,*tail, *temp, * prevnode, * prev;


 void linkedListTraversal(struct Node *temp)

 {

      temp= head;

    do{

        printf("Element is %d\n", temp->data);

        temp=temp->next;

    }while(temp!=NULL);

}


struct Node * DeleteAtBegin( struct Node * head)

  {

       temp=head;

       head=head->next;
      head->prev=NULL;

       free(temp);

            return head;
  }

  // create list
int main()
 {

 int value;
 char ch = 'y';
 printf("create a doubly  linked list\n");
while(ch =='y')
 {
  new = (struct Node *) malloc(sizeof(struct Node));
  printf("enter value to be inserted in linked list\n");
  scanf("%d",&value);
  new->data=value;
  if(head==NULL)
 {
    head=new;
    tail=new;
    head->next=NULL;
    head->prev=NULL;
 }
 else
   {
        temp=head;
      while(temp->next!=NULL)
     {
         prevnode=temp;
     temp=temp->next;
   }
    temp->next=new;
    new->next=NULL;
    new->prev=temp;
    tail=new;
   }

  fflush(stdin);
printf("Y- continue, N-exit\n");
scanf(" %c",&ch);
   }

   head=DeleteAtBegin(head);

  printf("Linked list \n");
  linkedListTraversal(head);
  return 0;

}
```

-x-x-x

Delete at end

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

struct Node{
  int data;
  struct Node * next;
  struct Node * prev;
}*new, *head,*tail, *temp, * prevnode, * prev;
```

```
 void linkedListTraversal(struct Node *temp)
 {
      temp= head;
    do{
        printf("Element is %d\n", temp->data);
        temp=temp->next;
    }while(temp!=NULL);
}

struct Node * DeleteAtEnd( struct Node * head)
  {



     temp=head;
      while(temp->next!=NULL)
     {
         prevnode=temp;
         temp=temp->next;
     }
        prevnode->next=NULL;
        free(temp);
        return head;
       }









  // create list
int main()
 {
```

```
 int value;
 char ch = 'y';
 printf("create a doubly  linked list\n");
while(ch =='y')
 {
  new = (struct Node *) malloc(sizeof(struct Node));
  printf("enter value to be inserted in linked list\n");
  scanf("%d",&value);
  new->data=value;
  if(head==NULL)
 {
   head=new;
   tail=new;
   head->next=NULL;
   head->prev=NULL;
 }
 else
  {
     temp=head;
    while(temp->next!=NULL)
   {
      prevnode=temp;
      temp=temp->next;
   }
     temp->next=new;
     new->next=NULL;
     new->prev=temp;
     tail=new;
    }

   fflush(stdin);
printf("Y- continue, N-exit\n");
scanf(" %c",&ch);
```

```
      }

      head=DeleteAtEnd(head);

   printf("Linked list \n");
    linkedListTraversal(head);
   return 0;

}
```

-x-x-x

Delete at position

```
struct Node{
    int data;
    struct Node * next;
    struct Node * prev;
}*new, *head,*tail, *temp, * prevnode, * prev;

 void linkedListTraversal(struct Node *temp)
 {
      temp= head;
    do{
        printf("Element is %d\n", temp->data);
        temp=temp->next;
    }while(temp!=NULL);
}

struct Node * DeleteAtPosition( struct Node * head, int position)
  {


     int i=1;
       struct Node *nextnode;
       temp=head;
       while(i<position)
       {
            temp=temp->next;
            i++;
       }

       prevnode=temp->prev;
       nextnode=temp->next;
       prevnode->next=nextnode;
       nextnode->prev=prevnode;
       free(temp);

            return head;
  }

  // create list
int main()
 {

 int value, position;
 char ch = 'y';
 printf("create a doubly  linked list\n");
while(ch =='y')
 {
  new = (struct Node *) malloc(sizeof(struct Node));
  printf("enter value to be inserted in linked list\n");
  scanf("%d",&value);
  new->data=value;
  if(head==NULL)
 {
    head=new;
    tail=new;
    head->next=NULL;
    head->prev=NULL;
 }
 else
   {
        temp=head;
      while(temp->next!=NULL)
     {
         prevnode=temp;
         temp=temp->next;
     }
        temp->next=new;
        new->next=NULL;
        new->prev=temp;
        tail=new;
       }

    fflush(stdin);
printf("Y- continue, N-exit\n");
scanf(" %c",&ch);
      }
  printf("enetr position \n");
      scanf("%d",&position);
      head=DeleteAtPosition(head, position);

   printf("Linked list \n");
    linkedListTraversal(head);
   return 0;

}
```