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

int length (node *head)
{
   int count =0;
   while (head != NULL)
{
    head = head -> next;
    count ++;
}
   return count;
}
void insertAtEnd (node **head, int d)
 {
    node *n,
    *temp = *head;
  if (*head == NULL)
{
    *head =(node *) malloc (sizeof (node));
    (*head) -> prev = NULL;
    (*head) -> data = d;
    (*head) -> next = NULL;
}
   else
```

```
{
    while (temp->next!= NULL)
    { temp = temp-> next;
    }
    n = (node *) malloc ( sizeof (node));
    n-> data =d;
    n->next = NULL;
    n-> prev = temp;
    temp->next = n;
}
printf ( "%d was inserted in the list \n", d);
}

Void insertleft (node **h, int d, int ele)
{ node *head = *h;
    if (head -> data == ele)
{   node * temp1 = NULL;
    temp1 = (node *) malloc (size of (node));
    temp1 -> prev = NULL;
    temp1-> data = d;
    temp1-> next = *h;
    (*h) -> prev = temp1;
    *h = temp1;
printf ("%d was inserted at start \n", d);
return;
}

node * temp;
while (head != NULL)
{
```

```
head = head -> prev;
temp = (node *)malloc (sizeof(node));
temp -> data = d;
temp -> prev = head;
temp -> next = head -> next;
temp -> next -> prev = temp;
printf ("%d was inserted to the left of %d\n",
        d, ele);
    break;
  }
  else
{
  head = head -> next;
  }
}
printf ("Given element is not present in the
        list\n");
}

Void deleteNode (node** head, int d)
{
  node * temp = * head;

  if (*head = NULL)
  { printf ("No elements in the list to delete")
    return;
  }
  while (temp != NULL)
    if (temp -> data == d)
    {
      if (temp -> data == d)
```

```
    if (temp == *head)
{   *head = (*head) -> next;
    (*head) -> prev = NULL;
}
else if (temp -> next == NULL)
  { temp -> prev -> next = NULL;
    free (temp);
  }
else
{
  temp -> prev -> next = temp -> next;
  temp -> next -> prev = temp -> prev;
  free (temp);
}
  printf ("%d was deleted \n", d);
  return;
)
temp = temp -> next;
}
printf ("%d is not present in the list\n" d);
}
void display (node *head)
{ if (head == NULL)
    printf ("Empty List \n");
    return;
}
  while (head != NULL) {
  printf ("<- %d -> ", head -> data);
    head = head -> next; }
  printf ("\n");
```

```
int main()
{
    node *head = NULL;
    int data, pos, opt;
printf ("Insert few elements in the list (Press -1
                to stop): \n");;
    scanf ("%d", &data);
    while (data != -1)
{
    insertAtEnd (&head, data);
    scanf ("%d, &data);
}
printf ("Operations on Doubly Linked List \n");
printf (" 1. Insert At left \n"
        2. Delete specified node \n
        3. Display \n
        4. Insert At End \n
        5. Exit \n" );
    printf ("Your choice: ");
    scanf ("%d", &opt);

    while (opt != 5)
{
    switch (opt) {
        case 1: printf ("Enter element to be inser
                scanf ("%d", &data);
                printf ("Enter the node: ");
                scanf ("%d", &pos);
                insertLeft (&head, data, pos);
                break;
```

```
case 2: printf ("Enter the element to be deleted: ");
        scanf ("%d", &data);
        deleteNode (&head, data);
        break;
case 3:  display (head);
        break;
case 4: printf ("Enter data to be inserted at end
        scanf ("%d", &data);
        insertAtEnd (&head, data);
        }
printf (" 1. Insert At Left \n
        2. Delete specified node \n
        3. Display \n
        4. Insert At End \n
        5. Exit \n");
printf ("Your choice: ");
scanf ("%d", &opt);
    }
}
```