```c
//A Complete Source Code for the Implementation of Double Linked List:
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
struct dlinklist
{
struct dlinklist *left;
int data;
struct dlinklist *right;
};
typedef struct dlinklist node;
node *start = NULL;
node* getnode()
{
node * newnode;
newnode = (node *) malloc(sizeof(node));
printf("\n Enter data: ");
scanf("%d", &newnode -> data);
newnode -> left = NULL;
newnode -> right = NULL;
return newnode;
}
int countnode(node *start)
{
if(start == NULL)
return 0;
else
return 1 + countnode(start -> right);
}
int menu()
{
int ch;
clrscr();
printf("\n 1.Create");
printf("\n------------------------------");
printf("\n 2. Insert a node at beginning ");
printf("\n 3. Insert a node at end");
printf("\n 4. Insert a node at middle");
printf("\n------------------------------");
printf("\n 5. Delete a node from beginning");
printf("\n 6. Delete a node from Last");
printf("\n 7. Delete a node from Middle");
printf("\n------------------------------");
printf("\n 8. Traverse the list from Left to Right ");
printf("\n 9. Traverse the list from Right to Left ");
printf("\n------------------------------");
printf("\n 10.Count the Number of nodes in the list");
printf("\n 11.Exit ");
printf("\n\n Enter your choice: ");
scanf("%d", &ch);
return ch;
}
```

```
void createlist(int n)
{
int i;
node *newnode;
node *temp;
for(i = 0; i < n; i++)
{
newnode = getnode();
if(start == NULL)
start = newnode;
else
{
temp = start;
while(temp -> right)
temp = temp -> right;
temp -> right = newnode;
newnode -> left = temp;
}
}
}

void traverse_left_to_right()
{
node *temp;
temp = start;
printf("\n The contents of List:
"); if(start == NULL )
printf("\n Empty List");
else
{
while(temp != NULL)
{
printf("\t %d ", temp -> data);
temp = temp -> right;
}
}
}
void traverse_right_to_left()
{
node *temp;
temp = start;
printf("\n The contents of List:
"); if(start == NULL)
printf("\n Empty List");
else
{
while(temp -> right != NULL)
temp = temp -> right;
}
while(temp != NULL)
{
printf("\t%d", temp ->
data); temp = temp -> left;
}
}
void dll_insert_beg()
{
node *newnode;
newnode = getnode();
if(start == NULL)
start = newnode;
else
{
newnode -> right = start;
start -> left = newnode;
start = newnode;
}
}
void dll_insert_end()
{
node *newnode, *temp;
newnode = getnode();
if(start == NULL)
else
{
}
}
start = newnode;
temp = start;
while(temp -> right != NULL)
temp = temp -> right;
temp -> right = newnode;
newnode -> left = temp;

void dll_insert_mid()
{
node *newnode,*temp;
int pos, nodectr, ctr = 1;
newnode = getnode();
printf("\n Enter the position: ");
scanf("%d", &pos);
nodectr = countnode(start);
if(pos - nodectr >= 2)
{
printf("\n Position is out of range..");
return;
}
if(pos > 1 && pos < nodectr)
{
temp = start;
while(ctr < pos - 1)
{
temp = temp -> right;
ctr++;
```

```
}
newnode -> left = temp; newnode
-> right = temp -> right; temp ->
right -> left = newnode; temp ->
right = newnode;
}
else
printf("position %d of list is not a middle position ", pos);
}
void dll_delete_beg()
{
node *temp;
if(start == NULL)
{
printf("\n Empty
list"); getch();
return ;
}
else
{
temp = start;
start = start -> right;
start -> left = NULL;
free(temp);
}
}
void dll_delete_last()
{
node *temp;
if(start == NULL)
{
printf("\n Empty
list"); getch();
return ;
}
else
{
temp = start;
while(temp -> right != NULL)

temp = temp -> right;
temp -> left -> right = NULL;
free(temp);
temp = NULL;
}
}
void dll_delete_mid()
{
int i = 0, pos, nodectr;
node *temp;
if(start == NULL)
{
```

```
printf("\n Empty List");
getch();
return;
}
else
{
printf("\n Enter the position of the node to delete: ");
scanf("%d", &pos);
nodectr = countnode(start);
if(pos > nodectr)
{
printf("\nthis node does not
exist"); getch();
return;
}
if(pos > 1 && pos < nodectr)
{
temp =
start; i = 1;
while(i < pos)
{
temp = temp -> right;
i++;
}
temp -> right -> left = temp -> left;
temp -> left -> right = temp -> right;
free(temp);
printf("\n node deleted..");
}
else
{
printf("\n It is not a middle position..");
getch();
}
}
}
void main(void)
{
int ch, n;
clrscr();
while(1)
{
ch = menu();
switch( ch)
{
case 1 :
printf("\n Enter Number of nodes to create: ");
scanf("%d", &n);
createlist(n);

printf("\n List
created.."); break;
:
dll_insert_beg();
break;
:
dll_insert_end();
break;
:
dll_insert_mid();
break;
:
dll_delete_beg();
break;
case 6 : dll_delete_last();
break;
case 7 :
dll_delete_mid();
break;
case 8 :
traverse_left_to_right();
break;
case 9 :
traverse_right_to_left();
break;
case 10 :
printf("\n Number of nodes: %d", countnode(start));
break;
case 11:
exit(0);
}
getch();
}
}
```