**Program:**

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <malloc.h>

struct node

{

    int data;

    struct node \*next;

};

typedef struct node node;

node \*start = NULL;

node \*create\_ll(node \*);

node \*insert\_beg(node \*);

node \*insert\_end(node \*);

node \*delete\_beg(node \*);

node \*delete\_end(node \*);

node \*delete\_after(node \*);

void node\_count(node \*);

void display(node \*);

int main()

{

    int choice, answer;

    printf("\n Press 1 to create Link list");

    printf("\n Press 2 to insert at Beginning ");

    printf("\n Press 3 to insert at end");

    printf("\n Press 4 to delete at Beginning ");

    printf("\n Press 5 to delete at end");

    printf("\n Press 6 to delete after given node");

    printf("\n Press 7 to display Link list");

    printf("\n Press 8 to get the number of nodes");

    do

    {

        printf("\n Enter your choice");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            start = create\_ll(start);

            break;

        case 2:

            start = insert\_beg(start);

            break;

        case 3:

            start = insert\_end(start);

            break;

        case 4:

            start = delete\_beg(start);

            break;

        case 5:

            start = delete\_end(start);

            break;

        case 6:

            start = delete\_after(start);

            break;

        case 7:

            display(start);

            break;

        case 8:

            node\_count(start);

            break;

        default:

            printf("\n Wrong choice");

            break;

        }

        printf("\n Press 1 to choose another option ");

        scanf("%d", &answer);

    } while (answer == 1);

    return 0;

}

node \*create\_ll(node \*start)

{

    node \*new\_node, \*ptr;

    int num, answer;

    do

    {

        printf("Enter data \n");

        scanf("%d", &num);

        new\_node = (node \*)malloc(sizeof(node));

        new\_node->data = num;

        if (start == NULL)

        {

            new\_node->next = new\_node;

            start = new\_node;

        }

        else

        {

            ptr = start;

            while (ptr->next != start)

                ptr = ptr->next;

            ptr->next=new\_node;

            new\_node->next = start;

        }

        printf("\n Press 1 to continue adding nodes ");

        scanf("%d", &answer);

    } while (answer == 1);

    return start;

}

node \*insert\_beg(node \*start)

{

    int num;

    node \*ptr, \*new\_node;

    printf("\n Enter data ");

    scanf("%d", &num);

    new\_node = (node \*)malloc(sizeof(node));

    new\_node->data = num;

    ptr = start;

    while(ptr->next !=start)

        ptr=ptr->next;

    ptr->next = new\_node;

    new\_node->next=start;

    start = new\_node;

    return start;

}

node \*insert\_end(node \*start)

{

    int num;

    node \*ptr, \*new\_node;

    printf("\n Enter data");

    scanf("%d", &num);

    new\_node = (node \*)malloc(sizeof(node));

    new\_node->data = num;

    ptr = start;

    while (ptr->next != start)

        ptr = ptr->next;

    ptr->next = new\_node;

    new\_node->next = start;

    return start;

}

node \*delete\_beg(node \*start)

{

    node \*ptr;

    while(ptr->next!=start)

        ptr=ptr->next;

    ptr->next=start->next;

    free(start);

    start=ptr->next;

    return start;

}

node \*delete\_end(node \*start)

{

    node \*ptr,\*preptr;

    ptr=start;

    while(ptr->next !=start)

    {

        preptr=ptr;

        ptr=ptr->next;

    }

    preptr->next=ptr->next;

    free(ptr);

    return start;

}

node \*delete\_after(node \*start)

{

    int val;

    node \*ptr,\*preptr;

    printf("Enter the value of the node after which deletion will happen \n");

    scanf("%d", &val);

    ptr = start;

    preptr=ptr;

    while (preptr->data != val)

    {

        preptr=ptr;

        ptr = ptr->next;

    }

    preptr->next = ptr ->next;

    if(ptr==start)

        start=preptr->next;

    free(ptr);

    return start;

}

void display(node \*start)

{

    node \*ptr;

    ptr = start;

    while (ptr->next!= start)

    {

        printf("\t %d", ptr->data);

        ptr = ptr->next;

    }

     printf("\t %d", ptr->data);

}

void node\_count(node \*start)

{

node \*ptr;

int count=0;

ptr = start;

    while (ptr->next != start)

    {

        count++;

        ptr = ptr->next;

    }

printf("There are %d nodes in given linked list \n",count+1);

}

**Output:**

Press 1 to create Link list

Press 2 to insert at Beginning

Press 3 to insert at end

Press 4 to delete at Beginning

Press 5 to delete at end

Press 6 to delete after given node

Press 7 to display Link list

Press 8 to get the number of nodes

Enter your choice1

Enter data

12

Press 1 to continue adding nodes 1

Enter data

13

Press 1 to continue adding nodes 1

Enter data

14

Press 1 to continue adding nodes 1

Enter data

15

Press 1 to continue adding nodes 0

Press 1 to choose another option 1

Enter your choice2

Enter data 11

Press 1 to choose another option 1

Enter your choice3

Enter data16

Press 1 to choose another option 1

Enter your choice7

11 12 13 14 15 16

Press 1 to choose another option 1

Enter your choice8

There are 6 nodes in given linked list