**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 \*insert\_before(node \*);

node \*insert\_after(node \*);

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

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

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

node \*delete\_list(node \*);

node \*sort\_list(node \*);

void display(node \*);

void search(node \*);

void node\_count(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 insert before given node");

    printf("\n Press 5 to insert after given node ");

    printf("\n Press 6 to delete at Beginning ");

    printf("\n Press 7 to delete at end");

    printf("\n Press 8 to delete after given node");

    printf("\n Press 9 to delete entire list ");

    printf("\n Press 10 to display Link list");

    printf("\n Press 11 to sort Link list");

    printf("\n Press 12 to search for some element");

    printf("\n Press 13 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 = insert\_before(start);

            break;

        case 5:

            start = insert\_after(start);

            break;

        case 6:

            start = delete\_beg(start);

            break;

        case 7:

            start = delete\_end(start);

            break;

        case 8:

            start = delete\_after(start);

            break;

        case 9:

            start = delete\_list(start);

            break;

        case 10:

            display(start);

            break;

        case 11:

            start = sort\_list(start);

            break;

        case 12:

            search(start);

            break;

        case 13:

            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 = NULL;

            start = new\_node;

        }

        else

        {

            ptr = start;

            while (ptr->next != NULL)

                ptr = ptr->next;

            ptr->next=new\_node;

            new\_node->next = NULL;

        }

        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;

    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;

    new\_node->next = NULL;

    ptr = start;

    while (ptr->next != NULL)

        ptr = ptr->next;

    ptr->next = new\_node;

    return start;

}

node \*insert\_before(node \*start)

{

    int num,val;

    node \*ptr,\*preptr,\*new\_node;

    printf("Enter data  \n");

    scanf("%d", &num);

    printf("Enter value of node before which data is to be inserted  \n");

    scanf("%d", &val);

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

    new\_node->data = num;

    ptr = start;

    while (ptr->data != val)

    {

        preptr=ptr;

        ptr = ptr->next;

    }

    preptr->next = new\_node;

    new\_node ->next=ptr;

    return start;

}

node \*insert\_after(node \*start)

{

    int num,val;

    node \*ptr,\*preptr,\*new\_node;

    printf("Enter data  \n");

    scanf("%d", &num);

    printf("Enter value of node after which data is to be inserted  \n");

    scanf("%d", &val);

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

    new\_node->data = num;

    ptr = start;

    while (preptr->data != val)

    {

        preptr=ptr;

        ptr = ptr->next;

    }

    preptr->next = new\_node;

    new\_node ->next=ptr;

    return start;

}

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

{

    node \*ptr;

    ptr=start;

    start=start->next;

    free(ptr);

    return start;

}

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

{

    node \*ptr,\*preptr;

    ptr=start;

    while(ptr->next !=NULL)

    {

        preptr=ptr;

        ptr=ptr->next;

    }

    ptr->next=NULL;

    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;

    while (preptr->data != val)

    {

        preptr=ptr;

        ptr = ptr->next;

    }

    preptr->next = ptr ->next;

    free(ptr);

    return start;

}

node \*delete\_list(node \*start)

{

    node \*ptr;

    if(start !=NULL)

    {

        ptr=start;

        while(ptr!=NULL)

        {

            start=delete\_beg(ptr);

            ptr=start;

        }

    }

    printf("Entire list is deleted");

    return start;

}

node \*sort\_list(node \*start)

{

 node \*ptr1,\*ptr2;

 int temp;

 ptr1=start;

 while(ptr1 ->next !=NULL)

 {

    ptr2=ptr1->next;

    while(ptr2 !=NULL)

    {

        if(ptr1 ->data > ptr2->data)

        {

            temp=ptr1->data;

            ptr1->data=ptr2->data;

            ptr2->data=temp;

        }

        ptr2=ptr2 ->next;

    }

    ptr1=ptr1 -> next;

 }

 display(start);

return start;

}

void display(node \*start)

{

    node \*ptr;

    ptr = start;

    while (ptr != NULL)

    {

        printf("\t %d", ptr->data);

        ptr = ptr->next;

    }

}

void search(node \*start)

{

    int num, flag;

    node \*ptr;

    ptr = start;

    printf("Enter data to be found \n");

    scanf("%d", &num);

    while (ptr->next != NULL)

    {

        if (ptr->data == num)

            flag = 1;

        else

            flag = 0;

        ptr = ptr->next;

    }

    if (flag == 1)

        printf("Element is present in the link list \n");

}

void node\_count(node \*start)

{

node \*ptr;

ptr=start;

int count=0;

while(ptr !=NULL)

{

    ptr=ptr->next;

    count++;

}

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

}

**Output:**

Press 1 to create Link list

Press 2 to insert at Beginning

Press 3 to insert at end

Press 4 to insert before given node

Press 5 to insert after given node

Press 6 to delete at Beginning

Press 7 to delete at end

Press 8 to delete after given node

Press 9 to delete entire list

Press 10 to display Link list

Press 11 to sort Link list

Press 12 to search for some element

Press 13 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 0

Press 1 to choose another option 1

Enter your choice4

Enter data

15

Enter value of node before which data is to be inserted

14

Press 1 to choose another option 1

Enter your choice2

Enter data 11

Press 1 to choose another option 1

Enter your choice3

Enter data15

Press 1 to choose another option 1

Enter your choice5

Enter data

18

Enter value of node after which data is to be inserted

11

Press 1 to choose another option 1

Enter your choice10

11 18 12 13 15 14 15

Press 1 to choose another option 1

Enter your choice11

The sorted list is

11 12 13 14 15 15 18

Press 1 to choose another option 1

Enter your choice13

There are 7 nodes in given linked list

Press 1 to choose another option 1

Enter your choice9

Entire list is deleted

Press 1 to choose another option0