**Executable Code :**

#include <stdio.h>

#include <stdlib.h>

struct node

{

    int data;

    struct node \*next;

};

struct node \*insert(struct node \*list, int value)

{

    struct node \*help\_ptr;

    help\_ptr = NULL;

    help\_ptr = list;

    struct node \*newnode = (struct node \*)malloc(sizeof(struct node));

    newnode->data = value;

    newnode->next = NULL;

    if (list == NULL || list->data > value)

    {

        newnode->next = list;

        list = newnode;

        return list;

    }

    while (help\_ptr->next != NULL && help\_ptr->next->data < value)

        help\_ptr = help\_ptr->next;

    newnode->next = help\_ptr->next;

    help\_ptr->next = newnode;

    return list;

}

void display(struct node \*list)

{

    struct node \*help\_ptr;

    help\_ptr = NULL;

    help\_ptr = list;

    if (help\_ptr == NULL)

        printf("List is Empty\n");

    while (help\_ptr != NULL)

    {

        printf("%4d", help\_ptr->data);

        help\_ptr = help\_ptr->next;

    }

}

struct node \*delete (struct node \*list, int target)

{

    struct node \*help\_ptr, \*node2delete;

    help\_ptr = NULL;

    help\_ptr = list;

    if (help\_ptr == NULL)

    {

        printf("List is Empty\n");

    }

    if (help\_ptr != NULL)

    {

        if (help\_ptr->data == target)

        {

            list = help\_ptr->next;

            printf("%d Deleted\n", target);

            free(help\_ptr);

            return list;

        }

        while (help\_ptr->next != NULL)

        {

            node2delete = NULL;

            if (help\_ptr->next->data == target)

            {

                node2delete = help\_ptr->next;

                help\_ptr->next = help\_ptr->next->next;

                printf("%d Deleted\n", target);

                free(node2delete);

                return list;

            }

            help\_ptr = help\_ptr->next;

        }

        printf("Element not Found\n");

        return list;

    }

}

void main()

{

    int choice;

    struct node \*prev, \*head, \*p;

    int number = 0;

    struct node \*list = NULL;

    while (choice != 4)

    {

        printf("\nEnter the option between 1,2 and 3 to choose a Linked List Function to be executed\n");

        printf("1: Insert\n");

        printf("2: Delete\n");

        printf("3: Display\n");

        printf("4: Exit\n");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            printf("Insert function is chosen\n");

            number = 0;

            printf("Enter data for the node ");

            scanf("%d", &number);

            if (number != -1)

                list = insert(list, number);

            break;

        case 2:

            printf("Delete function is chosen\n");

            number = 0;

            printf("Enter the data you wish to Delete: ");

            scanf("%d", &number);

            if (number != -1)

                list = delete (list, number);

            break;

        case 3:

            printf("Display function is chosen\n");

            display(list);

            break;

        case 4:

            printf("Exiting the Program\n");

            break;

        default:

            printf("Invalid option\n");

            break;

        }

    }

}

**Output :**

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

1

Insert function is chosen

Enter data for the node 5

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

1

Insert function is chosen

Enter data for the node 8

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

1

Insert function is chosen

Enter data for the node 1

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

1

Insert function is chosen

Enter data for the node 3

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

3

Display function is chosen

1 3 5 8

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

1

Insert function is chosen

Enter data for the node 16

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

3

Display function is chosen

1 3 5 8 16

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

2

Delete function is chosen

Enter the data you wish to Delete: 3

3 Deleted

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

3

Display function is chosen

1 5 8 16

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

2

Delete function is chosen

Enter the data you wish to Delete: 99

Element not Found

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

2

Delete function is chosen

Enter the data you wish to Delete: 1

1 Deleted

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

2

Delete function is chosen

Enter the data you wish to Delete: 5

5 Deleted

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

2

Delete function is chosen

Enter the data you wish to Delete: 8

8 Deleted

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

2

Delete function is chosen

Enter the data you wish to Delete: 16

16 Deleted

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

3

Display function is chosen

List is Empty

Enter the option between 1,2 and 3 to choose a Linked List Function to be executed

1: Insert

2: Delete

3: Display

4: Exit

4

Exiting the Program