

PROGRAM 1

```
Starthere  x  linkedlist2.c  x  linked list insert.c  x  linked list delete.c  x  linkedlist del.c  x

1  #include<stdio.h>
2  #include<conio.h>
3  #include<stdlib.h>
4  struct Node
5  {
6      int data;
7      struct Node *link;
8  };
9  typedef struct Node Node;
10 Node *start=NULL, *new, *curr, *temp;
11 void create();
12 void display();
13 void insert_beg();
14 void insert_end();
15 void insert_pos();
16 void main()
17 {
18     int choice;
19     while(1)
20     {
21         printf("1.Create\n 2.Display\n 3.Insert at begining \n 4.Insert at end\n 5.Insert at required position\n 6.Exit\n");
22         printf("Enter the choice\n");
23         scanf("%d",&choice);
24         switch(choice)
25         {
26             case 1: create();
27             break;
28             case 2: display();
29             break;
30             case 3: insert_beg();
31             break;
32             case 4: insert_end();
33             break;
34             case 5: insert_pos();
35             break;
36             case 6: exit(0);
37             break;
38             default:printf("Invalid choice\n");
39         }
40     }
41     getch();
42 }
43
44 void create()
45 {
46     int choice;
47     Node *new, *curr;
48     start = (Node*)malloc(sizeof(Node));
49     curr = start;
50     printf("Enter element:\n");
```

```

50     printf("Enter element:\n");
51     scanf("%d", &start->data);
52     while(1)
53     {
54         printf("Do you want to add an element? 0 for no\n");
55         scanf("%d", &choice);
56         if(choice!=0)
57         {
58             new = (Node*)malloc(sizeof(Node));
59             printf("Please enter element:\n");
60             scanf("%d", &new->data);
61             curr->link = new;
62             curr = new;
63         }
64         else
65         {
66             curr->link=NULL;
67             break;
68         }
69     }
70 }
71
72 void display()
73 {
74     Node *temp;
75     if(start==NULL)
76     {
77         printf("Linked list is empty");
78         return;
79     }
80     temp=start;
81     while(temp!=NULL)
82     {
83         printf("%d\t",temp->data);
84         temp=temp->link;
85     }
86 }
87
88 void insert_beg()
89 {
90     new=(Node*)malloc(sizeof(Node));
91     printf("Enter element\n");
92     scanf("%d", &new->data);
93     if(start==NULL)
94     {
95         start=new;
96         new->link=NULL;
97         return;
98     }
99     new->link=start;

```

```
Start here  X  Linkedlist2.c  X  *linked list insert.c  X  *linked list delete.c  X  linkedlist del.c  X

100     start=new;
101     }
102
103     void insert_end()
104     {
105         new=(Node*)malloc(sizeof(Node));
106         printf("Enter element\n");
107         scanf("%d",&new->data);
108         if(start==NULL)
109         {
110             start=new;
111             new->link=NULL;
112             return;
113         }
114         temp=start;
115         while(temp->link!=NULL)
116         {
117             temp=temp->link;
118         }
119         temp->link=new;
120         new->link=NULL;
121     }
122
123     void insert_pos()
124     { int pos,i=1;
125         new=(Node*)malloc(sizeof(Node));
126         printf("Enter element\n");
127         scanf("%d",&new->data);
128         printf("Enter pos\n");
129         scanf("%d",&pos);
130         if(pos==1)
131         {
132             new->link=start;
133             start=new;
134             return;
135         }
136         temp=start;
137         while(i<(pos-1) && temp!=NULL)
138         {
139             temp=temp->link;
140             i++;
141         }
142         if(i==(pos-1))
143         {
144             new->link=temp->link;
145             temp->link=new;
146             return;
147         }
148         if(temp==NULL)
149         {
```

"C:\Users\bmsce\Desktop\1BM21CS253\linked list insert.exe"

```
2.Display
3.Insert at begining
4.Insert at end
5.Insert at required position
6.Exit
Enter the choice
1
Enter element:
10
Do you want to add an element? 0 for no
1
Please enter element:
20
Do you want to add an element? 0 for no
0
1.Create
2.Display
3.Insert at begining
4.Insert at end
5.Insert at required position
6.Exit
Enter the choice
3
Enter element
50
1.Create
2.Display
3.Insert at begining
4.Insert at end
5.Insert at required position
6.Exit
Enter the choice
4
Enter element
60
1.Create
2.Display
3.Insert at begining
4.Insert at end
5.Insert at required position
6.Exit
Enter the choice
5
Enter element
70
Enter pos
2
1.Create
2.Display
3.Insert at begining
4.Insert at end
5.Insert at required position
6.Exit
Enter the choice
2
50      70      10      20      60      1.Create
2.Display
3.Insert at begining
4.Insert at end
5.Insert at required position
6.Exit
Enter the choice
```

Program 2

```
Start here  x Linkedlist2.c  x linked list insert.c  x *linked list delete.c  x linkedlist del.c  x
1  #include<stdio.h>
2  #include<conio.h>
3  #include<stdlib.h>
4  struct Node
5  {
6      int data;
7      struct Node *link;
8  };
9  typedef struct Node Node;
10 Node *start=NULL, *new, *curr,*temp;
11 void create();
12 void display();
13 void delete_beg();
14 void delete_end();
15 void delete_ele();
16 void main()
17 {
18     int choice;
19     while(1)
20     {
21         printf("\n 1.Create\n 2.Display\n 3.Delete at beginning \n 4.Delete at end\n 5.Delete the required element\n 6.Exit\n");
22         printf("Enter the choice\n");
23         scanf("%d",&choice);
24         switch(choice)
25         {
26             case 1: create();
27             break;
28             case 2: display();
29             break;
30             case 3: delete_beg();
31             break;
32             case 4: delete_end();
33             break;
34             case 5: delete_ele();
35             break;
36             case 6: exit(0);
37             break;
38             default:printf("Invalid choice\n");
39         }
40     }
41     getch();
42 }
43 void create()
44 {
45     int choice;
46     Node *new, *curr;
47     start = (Node*)malloc(sizeof(Node));
48     curr = start;
49     printf("Enter element:\n");
50     scanf("%d", &start->data);
```

```
Start here X LinkedList2.c X linked list insert.c X *linked list delete.c X linkedlist del.c X
50     scanf("%d", &start->data);
51     while(1)
52     {
53         printf("Do you want to add an element? 0 for no\n");
54         scanf("%d", &choice);
55         if(choice!=0)
56         {
57             new = (Node*)malloc(sizeof(Node));
58             printf("Please enter element:\n");
59             scanf("%d", &new->data);
60             curr->link = new;
61             curr = new;
62         }
63         else
64         {
65             curr->link=NULL;
66             break;
67         }
68     }
69 }
70
71 void display()
72 {
73     Node *temp;
74     if(start==NULL)
75     {
76         printf("Linked list is empty\n");
77         return;
78     }
79     temp=start;
80     while(temp!=NULL)
81     {
82         printf("%d\t",temp->data);
83         temp=temp->link;
84     }
85 }
86
87
88 void delete_beg()
89 {
90     if(start==NULL)
91     {
92         printf("Linked list is empty\n");
93     }
94     else
95     {
96         temp=start;
97         start=start->link;
98         free(temp);
99     }
```



```

98     }
99
100    void delete_end()
101    {
102        Node *prev, *next;
103        if(start==NULL)
104        {
105            printf("Linked list is empty\n");
106            return;
107        }
108        if(start->link==NULL)
109        {
110
111            free(start);
112            start=NULL;
113            return;
114        }
115        prev=start;
116        next=start->link;
117        while(next->link!=NULL)
118        {
119
120            prev=next;
121            next=next->link;
122        }
123        prev->link=NULL;
124        free(next);
125    }
126
127    void delete_ele()
128    {
129        int ele;
130        Node *prev, *next;
131        if(start==NULL)
132        {
133            printf("Linked list is empty");
134            return;
135        }
136        printf("Enter element to be deleted\n");
137        scanf("%d", &ele);
138        if(start->data==ele)
139        {
140            free(start);
141            start=NULL;
142            printf("Element deleted\n");
143            return;
144        }
145        prev=start;
146        next=start->link;
147        while(next->data!=ele && next!=NULL)

```

```
,
prev=start;
next=start->link;
while(next->data!=ele && next!=NULL)
{
    prev=next;
    next=next->link;
}
if(next->data==ele)
{
    prev->link=next->link;
    free(next);
    printf("Element deleted");
}
printf("Element not found");
}
```



```

1.Create
2.Display
3.Delete at begining
4.Delete at end
5.Delete the required element
6.Exit
Enter the choice
1
Enter element:
10
Do you want to add an element? 0 for n
1
Please enter element:
20
Do you want to add an element? 0 for n
1
Please enter element:
30
Do you want to add an element? 0 for n
0
1.Create
2.Display
3.Delete at begining
4.Delete at end
5.Delete the required element
6.Exit
Enter the choice
3
1.Create
2.Display
3.Delete at begining
4.Delete at end
5.Delete the required element
6.Exit
Enter the choice
2
20      30      1.Create
2.Display
3.Delete at begining
4.Delete at end
5.Delete the required element
6.Exit
Enter the choice
4
1.Create
2.Display
3.Delete at begining
4.Delete at end
5.Delete the required element
6.Exit
Enter the choice
2
20      1.Create
2.Display
3.Delete at begining
4.Delete at end
5.Delete the required element
6.Exit
Enter the choice
5
Enter element to be deleted
20
Element deleted

```