

EXP:5

SANSKAR SRIVASTAVA SY-IT 57

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <malloc.h>
```

```
// Defining Structure
```

```
typedef struct node
```

```
{
```

```
    int data;
```

```
    struct node *next;
```

```
} node;
```

```
node *createList();
```

```
node *Insert_beg(node *head, int x);
```

```
node *Insert_end(node *head, int x);
```

```
node *Insert_mid(node *head, int x);
```

```
node *Delete_beg(node *head);
```

```
node *Delete_end(node *head);
```

```
node *Delete_mid(node *head);
```

```
void PrintList(node *head);
```

```
// Main Function
```

```
void main()
```

```
{
```

```
    int choice, insert_option, delete_option, x;
```

```
    node *head = NULL;
```

```
    printf("Welcome to the implementation of the singly linked list ! \n");
```

```
    do
```

```
    {
```

```
        printf("Please select an operation to perform from the below list \n");
```

```
        printf(" 1. Create a List \n 2. Insert a node \n 3. Delete a node \n 4. Print the existing list \n
```

```
5. Exit \n");
```

```
        printf("Enter your choice: ");
```

```
        scanf("%d", &choice);
```

```
        printf("\n \n");
```

```
        switch (choice)
```

```
        {
```

```
        case 1:
```

```
            head = createList();
```

```
            break;
```

```
        case 2:
```

```
            do
```

```
            {
```

```
                printf("Select a position where you to want to insert new node \n");
```

```

    printf(" 1. Beginning of the List \n 2. At the end of the list \n 3. Insert in between \n 4.
Exit the insert operation \n");
    printf("Enter your choice: ");
    scanf("%d", &insert_option);
    switch (insert_option)
    {
    case 1:
        printf("Enter the data to be inserted: ");
        scanf("%d", &x);
        head = Insert_beg(head, x);
        break;
    case 2:
        printf("Enter the data to be inserted: ");
        scanf("%d", &x);
        head = Insert_end(head, x);
        break;
    case 3:
        printf("Enter the data to be inserted: ");
        scanf("%d", &x);
        head = Insert_mid(head, x);
        break;
    case 4:
        printf("Insert operation Exit");
        break;
    default:
        printf("Please enter a valid choide: 1, 2, 3, 4");
    }
} while (insert_option != 4);
printf("\n \n");
break;
case 3:
do
{
    printf("Select a position from where you to want to delete the element \n");
    printf(" 1. Beginning of the List \n 2. At the end of the list \n 3. Somewhere in between
\n 4. Exit the delete operation \n");
    printf("Enter your choice: ");
    scanf("%d", &delete_option);
    switch (delete_option)
    {
    case 1:
        head = Delete_beg(head);
        break;
    case 2:

```

```

        head = Delete_end(head);
        break;
    case 3:
        head = Delete_mid(head);
        break;
    case 4:
        printf("Delete Operation Exit");
        break;
    default:
        printf("Please enter a valid choide: 1, 2, 3, 4");
    }
} while (delete_option != 4);
printf("\n \n");
break;
case 4:
    PrintList(head);
    break;
case 5:
    printf("Exit: Program Finished !!");
    break;
default:
    printf("Please enter a valid choide: 1, 2, 3, 4, 5");
}
} while (choice != 5);
}

```

// Function to create List

```

node *createList()
{
    node *head, *p;
    int i, n;
    head = NULL;
    printf("Enter the number of nodes: ");
    scanf("%d", &n);
    printf("Enter the data: ");
    for (i = 0; i <= n - 1; i++)
    {
        if (head == NULL)
        {
            p = head = (node *)malloc(sizeof(node));
        }
        else
        {
            p->next = (node *)malloc(sizeof(node));

```

```

        p = p->next;
    }
    p->next = NULL;
    scanf("%d", &(p->data));
}
printf("\n \n");
return (head);
}

```

// Function to insert element

```

node *Insert_beg(node *head, int x)
{
    node *p;
    p = (node *)malloc(sizeof(node));
    p->data = x;
    p->next = head;
    head = p;
    return (head);
}

```

```

node *Insert_end(node *head, int x)
{
    node *p, *q;
    p = (node *)malloc(sizeof(node));
    p->data = x;
    p->next = NULL;
    if (head == NULL)
        return (p);
    for (q = head; q->next != NULL; q = q->next)
        ;
    q->next = p;
    return (head);
}

```

```

node *Insert_mid(node *head, int x)
{
    node *p, *q;
    int y;
    p = (node *)malloc(sizeof(node));
    p->data = x;
    p->next = NULL;
    printf("After which element you want to insert the new element ?");
    scanf("%d", &y);
    for (q = head; q != NULL && q->data != y; q = q->next)
        ;
    if (q != NULL)

```

```

{
    p->next = q->next;
    q->next = p;
}
else
    printf("ERROR !! Data Not Found");
return (head);
}

// Function to delete element
node *Delete_beg(node *head)
{
    node *p, *q;
    if (head == NULL)
    {
        printf("Empty Linked List");
        return (head);
    }
    p = head;
    head = head->next;
    free(p);
    return (head);
}

node *Delete_end(node *head)
{
    node *p, *q;
    if (head == NULL)
    {
        printf("Empty Linked List");
        return (head);
    }
    p = head;
    if (head->next == NULL)
    {
        head = NULL;
        free(p);
        return (head);
    }
    for (q = head; q->next->next != NULL; q = q->next)
        p = q->next;
    q->next = NULL;
    free(p);
    return (head);
}

```

```

node *Delete_mid(node *head)
{
    node *p, *q;
    int x, i;
    if (head == NULL)
    {
        printf("Empty Linked List");
        return (head);
    }
    printf("Enter the data to be deleted: ");
    scanf("%d", &x);
    if (head->data == x)
    {
        p = head;
        head = head->next;
        free(p);
        return (head);
    }
    for (q = head; q->next->data != x && q->next != NULL; q = q->next)
        if (q->next == NULL)
        {
            printf("ERROR !! Data Not Found");
            return (head);
        }
    p = q->next;
    q->next = q->next->next;
    free(p);
    return (head);
}

```

```

// Function to print the existing list
void PrintList(node *head)
{
    node *p;
    printf("[ ");
    for (p = head; p != NULL; p = p->next)
    {
        printf("%d \t", p->data);
    }
    printf(" ]");
    printf("\n \n");
}

```

```
Activities Terminal Aug 21 14:30 dl416@itadmin: ~
dl416@itadmin:~$ gedit sans.c
dl416@itadmin:~$ gcc sans.c
dl416@itadmin:~$ ./a.out
Welcome to the implementation of the singly linked list !
Please select an operation to perform from the below list
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
5. Exit
Enter your choice: 1
Enter the number of nodes: 12
Enter the data: 12
45
23
89
47
54
12
56
45
18
Please select an operation to perform from the below list
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
5. Exit
Enter your choice: 2
Select a position where you want to insert new node
1. Beginning of the List
2. At the end of the list
3. Insert in between
4. Exit the Insert operation
Enter your choice: 1
Enter the data to be inserted: 78

Documents sans.c
1#include <stdio.h>
2#include <stdlib.h>
3#include <malloc.h>
4
5// Defining Structure
6typedef struct node
7{
8    int data;
9    struct node *next;
10} node;
11node *createList();
12node *Insert_beg(node *head, int x);
13node *Insert_end(node *head, int x);
14node *Insert_mid(node *head, int x);
15node *Delete_beg(node *head);
16node *Delete_end(node *head);
17node *Delete_mid(node *head);
18void PrintList(node *head);
19
20// Main Function
21void main()
22{
23    int choice, insert_option, delete_option, x;
24    node *head = NULL;
25    printf("Welcome to the implementation of the
singly linked list ! \n");
26    do
27    {
28        printf("Please select an operation to perform
from the below list \n");
29        printf(" 1. Create a List \n 2. Insert a node
\n 3. Delete a node \n 4. Print the existing list \n
5. Exit \n");
30        printf("Enter your choice: ");
31        scanf("%d", &choice);
32        printf("\n \n");
33    }
34    while(choice != 5);
35    head = createList();
36    break;
37    case 2:
38    do
39    {
40        printf("Select a position where you
to want to insert new node \n");
41        printf(" 1. Beginning of the List \n
2. At the end of the list \n 3. Insert in between \n
4. Exit the Insert operation \n");
42        printf("Enter your choice: ");
43        scanf("%d", &insert_option);
44        switch (insert_option)
45        {
46        case 1:
47            printf("Enter the data to be
inserted: ");
48            scanf("%d", &x);
49            head = Insert_beg(head, x);
50            break;
51        case 2:
52            printf("Enter the data to be
inserted: ");
53            scanf("%d", &x);
54            head = Insert_end(head, x);
55            break;
56        case 3:
57            printf("Enter the data to be
inserted: ");
58            scanf("%d", &x);
59            head = Insert_mid(head, x);
60            break;
61        case 4:
62            printf("Insert operation Exit");
63            break;
64        }
```

```
45
23
89
47
54
12
56
45
18
Please select an operation to perform from the below list
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
5. Exit
Enter your choice: 3
After which element you want to insert the new element 748
ERROR !! Data Not FoundSelect a position where you want to insert new node
1. Beginning of the List
2. At the end of the list
3. Insert in between
4. Exit the Insert operation
Enter your choice: 4
Insert operation Exit
Please select an operation to perform from the below list
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
5. Exit
Enter your choice: 3
Select a position from where you want to delete the element
1. Beginning of the List
2. At the end of the list
3. Somewhere in between
4. Exit the delete operation
Enter your choice: 3
Enter the data to be deleted: 56
Select a position from where you want to delete the element
1. Beginning of the List
2. At the end of the list
3. Somewhere in between
4. Exit the delete operation
Enter your choice: 4
Delete Operation Exit
Please select an operation to perform from the below list
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
5. Exit
Enter your choice: 4
Print the existing list
45
23
89
47
54
12
56
45
18

Documents sans.c
36    head = createList();
37    break;
38    case 2:
39    do
40    {
41        printf("Select a position where you
to want to insert new node \n");
42        printf(" 1. Beginning of the List \n
2. At the end of the list \n 3. Insert in between \n
4. Exit the Insert operation \n");
43        printf("Enter your choice: ");
44        scanf("%d", &insert_option);
45        switch (insert_option)
46        {
47        case 1:
48            printf("Enter the data to be
inserted: ");
49            scanf("%d", &x);
50            head = Insert_beg(head, x);
51            break;
52        case 2:
53            printf("Enter the data to be
inserted: ");
54            scanf("%d", &x);
55            head = Insert_end(head, x);
56            break;
57        case 3:
58            printf("Enter the data to be
inserted: ");
59            scanf("%d", &x);
60            head = Insert_mid(head, x);
61            break;
62        case 4:
63            printf("Insert operation Exit");
64            break;
65        }
```

Activities Terminal Aug 21 14:31

dl416@itadmin: ~

```
4. Print the existing list
5. Exit
Enter your choice: 3

Select a position from where you to want to delete the element
1. Beginning of the List
2. At the end of the list
3. Somewhere in between
4. Exit the delete operation
Enter your choice: 3
Enter the data to be deleted: 56
Select a position from where you to want to delete the element
1. Beginning of the List
2. At the end of the list
3. Somewhere in between
4. Exit the delete operation
Enter your choice: 4
Delete Operation Exit

Please select an operation to perform from the below list
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
5. Exit
Enter your choice: 4

[ 78 12 45 23 89 47 54 12 56 45
45 18 ]

Please select an operation to perform from the below list
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
5. Exit
Enter your choice: 5

Exit: Program Finished !!dl416@itadmin:~$ gedit sans.c
dl416@itadmin:~$ gedit sans.c
^[[20~
```

Documents sans.c

```
67
68     } while (insert_option != 4);
69     printf("\n \n");
70     break;
71     case 3:
72     do
73     {
74         printf("Select a position from where
you to want to delete the element \n");
75         printf(" 1. Beginning of the List \n
2. At the end of the list \n 3. Somewhere in between
\n 4. Exit the delete operation \n");
76         printf("Enter your choice: ");
77         scanf("%d", &delete_option);
78         switch (delete_option)
79         {
80             case 1:
81                 head = Delete_beg(head);
82                 break;
83             case 2:
84                 head = Delete_end(head);
85                 break;
86             case 3:
87                 head = Delete_mid(head);
88                 break;
89             case 4:
90                 printf("Delete Operation Exit");
91                 break;
92             default:
93                 printf("Please enter a valid
choide: 1, 2, 3, 4");
94         }
95     } while (delete_option != 4);
96     printf("\n \n");
97     break;
98     case 4:
```

C Tab Width: 6 Ln 257, Col 2 INS