Experiment No 1(A):- Linked List

Program:-

Input:-

```
#include<stdio.h>
#include<malloc.h>
typedef struct NODE
{
int data;
struct NODE *next;
}node;

node *start=NULL;

void insert_beg();
void insert_end();
void insert_before();
void insert_after();
void delete_beg();
void delete_end();
void delete_before();
void delete_after();
void display();

void main()
{
int ch;
do
 {
  printf("1.Insert at the Beginning");
  printf("\n2.Insert at the End");
  printf("\n3.Insert before a given Node");
  printf("\n4.Insert after a given Node");
  printf("\n5.Delete from the Beginning");
  printf("\n6.Delete from the End");
  printf("\n7.Delete before a given Node");
  printf("\n8.Delete after a given Node");
  printf("\n9.Display");
  printf("\n10.Exit");
  printf("\nEnter Your Choice:- ");
  scanf("%d",&ch);
  switch(ch)
   {
     case 1 : insert_beg();
              display();
              break;
     case 2 : insert_end();
              display();
              break;
```

```
      case 3 : insert_before();
               display();
               break;
      case 4 : insert_after();
               display();
               break;
      case 5 : delete_beg();
               display();
               break;
      case 6 : delete_end();
               display();
               break;
      case 7 : delete_before();
               display();
               break;
      case 8 : delete_after();
               display();
               break;
      case 9 : display();
               break;
      case 10 : break;
      default: printf("\nInvalid Choice");
    }
  }while(ch!=10);
}

void insert_beg()
{
int i;
node *p;
p=(node*)malloc(sizeof(node));
printf("\nEnter the information:-");
scanf("%d",&i);
p->data=i;
p->next=NULL;
if(start==NULL)
   start=p;
else
   {
     p->next=start;
     start=p;
    }
}

void insert_end()
{
int i;
node *p,*ptr;
```

```
p=(node*)malloc(sizeof(node));
printf("\nEnter the information:- ");
scanf("%d",&i);
p->data=i;
p->next=NULL;
if(start==NULL)
   start=p;
else
   {
     ptr=start;
     while(ptr->next!=NULL)
          ptr=ptr->next;
      ptr->next=p;
    }
}

void insert_before()
{
int i,val;
node *p,*ptr,*pre_ptr;
p=(node*)malloc(sizeof(node));
printf("\nEnter the information:- ");
scanf("%d",&i);
p->data=i;
p->next=NULL;
printf("\nEnter the value before which the Node has to be inserted: ");
scanf("%d",&val);
if(start->data==val)
  {
   p->next=start;
   start=p;
   }
else
  {
   ptr=start;
   while(ptr->data!=val)
      {
       pre_ptr=ptr;
       ptr=ptr->next;
      }
   pre_ptr->next=p;
   p->next=ptr;
  }
}

void insert_after()
{
int i,val;
```

```
node *p,*ptr;
p=(node*)malloc(sizeof(node));
printf("\nEnter the information:- ");
scanf("%d",&i);
p->data=i;
p->next=NULL;
printf("\nEnter the value after which the Node has to be inserted:- ");
scanf("%d",&val);
ptr=start;
while(ptr->data!=val)
 ptr=ptr->next;
p->next=ptr->next;
ptr->next=p;
}

void delete_beg()
{
node *temp;
if(start==NULL)
  {
   printf("\nUnderflow\n");
   return;
   }
temp=start;
start=start->next;
free(temp);
}

void delete_end()
{
node *ptr,*pre_ptr;
if(start==NULL)
  {
   printf("\nUnderflow\n");
   return;
   }
ptr=start;
pre_ptr=ptr;
while(ptr->next!=NULL)
  {
   pre_ptr=ptr;
   ptr=ptr->next;
   }
pre_ptr->next=NULL;
free(ptr);
}

void delete_before()
```

```
{
int val;
node *temp,*pre_ptr,*ptr;
if(start==NULL)
 {
  printf("\nUnderflow\n");
  return;
  }
printf("\nEnter the value before which the Node has to be deleted:- ");
scanf("%d",&val);
if(start->next->data==val)
 {
  temp=start;
  start=start->next;
  free(temp);
  }
else
  {
   ptr=start;
   pre_ptr=ptr;
   while(ptr->next->data!=val)
      {
       pre_ptr=ptr;
       ptr=ptr->next;
       }
   pre_ptr->next=ptr->next;
   free(ptr);
   }
}

void delete_after()
{
int val;
node *temp,*ptr;
if(start==NULL)
 {
  printf("\nUnderflow\n");
  return;
  }
printf("\nEnter the value after which the Node has to be deleted:- ");
scanf("%d",&val);
ptr=start;
while(ptr->data!=val)
   ptr=ptr->next;
temp=ptr->next;
ptr->next=temp->next;
free(temp);
}
```

```
void display()
{
node *ptr;
if(start==NULL)
 {
  printf("\nUnderflow\n");
  return;
  }
ptr=start;
printf("\nThe Linked List is :-\n");
while(ptr!=NULL)
   {
    printf("%d->",ptr->data);
    ptr=ptr->next;
   }
printf("NULL\n");
}
```

Output:-

```
student@CE4PC-12:~$ cc -c linkedlist.c
student@CE4PC-12:~$ cc -o linkedlist linkedlist.c
student@CE4PC-12:~$ ./linkedlist
1.Insert at the Beginning
2.Insert at the End
3.Insert before a given Node
4.Insert after a given Node
5.Delete from the Beginning
6.Delete from the End
7.Delete before a given Node
8.Delete after a given Node
9.Display
10.Exit
Enter Your Choice:- 1

Enter the information:-12

The Linked List is :-
12->NULL
1.Insert at the Beginning
2.Insert at the End
3.Insert before a given Node
4.Insert after a given Node
5.Delete from the Beginning
6.Delete from the End
7.Delete before a given Node
8.Delete after a given Node
9.Display
10.Exit
Enter Your Choice:- 2

Enter the information:- 14
```

```
The Linked List is :-
12->14->NULL
1.Insert at the Beginning
2.Insert at the End
3.Insert before a given Node
4.Insert after a given Node
5.Delete from the Beginning
6.Delete from the End
7.Delete before a given Node
8.Delete after a given Node
9.Display
10.Exit
Enter Your Choice:- 3

Enter the information:- 16

Enter the value before which the Node has to be inserted: 14

The Linked List is :-
12->16->14->NULL
1.Insert at the Beginning
2.Insert at the End
3.Insert before a given Node
4.Insert after a given Node
5.Delete from the Beginning
6.Delete from the End
7.Delete before a given Node
8.Delete after a given Node
9.Display
10.Exit
Enter Your Choice:- 4

Enter the information:- 25

Enter the value after which the Node has to be inserted:- 14

The Linked List is :-
12->16->14->25->NULL
1.Insert at the Beginning
2.Insert at the End
3.Insert before a given Node
4.Insert after a given Node
5.Delete from the Beginning
6.Delete from the End
7.Delete before a given Node
8.Delete after a given Node
9.Display
10.Exit
Enter Your Choice:- 5

The Linked List is :-
16->14->25->NULL
1.Insert at the Beginning
2.Insert at the End
3.Insert before a given Node
4.Insert after a given Node
```

5.Delete from the Beginning
6.Delete from the End
7.Delete before a given Node
8.Delete after a given Node
9.Display
10.Exit
Enter Your Choice:- 6

The Linked List is :-
16->14->NULL
1.Insert at the Beginning
2.Insert at the End
3.Insert before a given Node
4.Insert after a given Node
5.Delete from the Beginning
6.Delete from the End
7.Delete before a given Node
8.Delete after a given Node
9.Display
10.Exit
Enter Your Choice:- 7

Enter the value before which the Node has to be deleted:- 14

The Linked List is :-
14->NULL
1.Insert at the Beginning
2.Insert at the End
3.Insert before a given Node
4.Insert after a given Node
5.Delete from the Beginning
6.Delete from the End
7.Delete before a given Node
8.Delete after a given Node
9.Display
10.Exit
Enter Your Choice:- 1

Enter the information:-23

The Linked List is :-
23->14->NULL
1.Insert at the Beginning
2.Insert at the End
3.Insert before a given Node
4.Insert after a given Node
5.Delete from the Beginning
6.Delete from the End
7.Delete before a given Node
8.Delete after a given Node
9.Display
10.Exit
Enter Your Choice:- 8

Enter the value after which the Node has to be deleted:- 23

```
The Linked List is :-
23->NULL
1.Insert at the Beginning
2.Insert at the End
3.Insert before a given Node
4.Insert after a given Node
5.Delete from the Beginning
6.Delete from the End
7.Delete before a given Node
8.Delete after a given Node
9.Display
10.Exit
Enter Your Choice:- 9

The Linked List is :-
23->NULL
1.Insert at the Beginning
2.Insert at the End
3.Insert before a given Node
4.Insert after a given Node
5.Delete from the Beginning
6.Delete from the End
7.Delete before a given Node
8.Delete after a given Node
9.Display
10.Exit
Enter Your Choice:- 10
student@CE4PC-12:~$
```