lab 9

Dimyashree.K
1BM19CS054

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   int data;
   struct node* next;
   struct node *prev;
};
struct node *head=NULL;
void insert_beg()
{
   struct node *new_node;
   new_node=(struct node*)malloc(sizeof
                                 (struct node));
   printf("Enter the item\n");
   scanf("%d",&new_node->data);
   new_node->next=NULL;
   new_node->prev=NULL;
   if(head==NULL)
   {
      head=new_node;
   }
   else
   {
      new_node->next=head;
      head->prev=new_node;
      head=new_node;
   }
}
void insert_end()
{
   struct node *new_node,*temp;
   new_node=(Struct node*)malloc(size of
                                 (struct node));
   printf("Enter the item\n");
```

```
scanf ("%d", &new_node->data);
new_node->next = NULL;
new_node->prev = NULL;
if (head == NULL)
{
  head = new_node;
}
else
{
  temp = head;
  while (temp->next != NULL)
  temp = temp->next;
  temp->next = new_node;
  new_node->prev = temp;
}
}
void insert_between()
{
int listele;
struct node *new_node, *temp;
printf ("Enter the element in the list \n");
scanf ("%d", &listele);
new_node = (struct node*) malloc (sizeof
                    (struct node));
printf ("Enter the new node node data \n");
scanf ("%d", &new_node->data);
new_node->next = NULL;
new_node->prev = NULL;
if (head == NULL)
{
  printf ("Empty list \n"); return;
}
temp = temp->next;
if (temp == NULL)
{
```

```
printf("Element is not in the list");
return;
}
}
if (temp->next==NULL)
{
new_node->next=temp->next;//
  becomes null
temp->next=new_node;
new_node->prev=temp;
return;
}
new_node->next=temp->next;//
temp->next=new_node;   becomes null
new_node->prev=temp;
new_node->next->prev=new_node;
}
void del()
{
struct node *temp;
int ele;
if (head==NULL)
{
printf("Empty list\n");
return;
}
printf("Enter the element to be
                deleted\n");
scanf("%d",&ele);
temp=head;
while(temp->data!=ele)
{
temp=temp->next;
if(temp==NULL)
{
```

```
printf ("Element is not in the list \n");
break;
}
}
if (temp == head)
{
head = head -> next;
}
else if (temp -> next == NULL)
{
temp = temp -> prev;
temp -> next = NULL;
}
else
{
temp -> prev -> next = temp -> next;
temp -> next -> prev = temp -> prev;
}
}
void display ()
{
struct node *temp;
temp = head;
while (temp != NULL)
{
printf ("%d \t", temp -> data);
temp = temp -> next
}
print ("\n");
}
int main
while (1)
{
printf ("1. Insert at the beg \n");
printf ("2. Insert at the end \n");
printf ("3. Insert after a given node \n");
```

```
printf("4. Delete\n");
printf("5. Display\n");
printf("6. Exit\n");
printf("Enter your choice\n");
scanf("%d", &choice);
switch (choice)
{
case 1: insert_beg();  break;
case 2: insert_end();  break;
case 3: insert_between();  break
case 4: del();  break
case 5: display();  break.
case 6: exit(0);
}
}
return 0;
}
```