week-6

```
#include<stdio.h>
#include<stdlib.h>
 Void create();
 void display();
 void delete_front();
 void delete_last();
 void delete_at_node(int);
 struct node
{
 int data;
 struct node *next;
};
struct node *head=NULL;
 int main()
{
 int choice, ele;
 do
 {
printf("\n1. Create \n2. Display \3. Delete at first \n4. Delete
        from end \n5. Delete particular element \n6. Exit \n");
printf("\n Enter your choice :");
scanf("%d", &choice);
switch(choice)
{
case 1: create(); break;
case 2: disply(); break;
case 3: delete_front();
 break;
case 4: delete_last();
 break;
case5: printf("Enter he element to be deleted \n");
```

```
scanf("%d",&ele);
delete_at_node(ele);
break;
default: exit(0);
}
}while(choice==1||choice==2||choice==3||choice==4||
                                        choice==5);
return 0;
}
void create()
{
struct node *newnode, *temp;
int item;
newnode=(struct node *)malloc(sizeof(struct node));
printf("Enter the data:");
scanf("%d",&item);
newnode->data=item;
if(head==NULL)
{
newnode->next=NULL;
head=newnode;
printf("Node Created\n");
}
else
{
temp=head;
while(temp->next!=NULL)
{
temp->next=newnode;
newnode->next=NULL;
printf("Node Created\n");
}
}
void delete_front()
{
```

```
if (head==NULL)
{
printf("Empty list. can't delete\n"); return;
}
else
head=head->next;
}
Void delete_last()
{
Struct node *temp;
if (head==NULL)
{
printf("Empty list. can't delete\n"); return;
}
else
{
temp=head;
while (temp->next->next!=NULL)
{
printf("%d", temp->data);
temp=temp->next;
}
temp->next=NULL;
}
}
void delete_at_node (int ele)
{
struct node *temp, *del=NULL;
if (head==NULL)
{
printf("Empty list. can't delete\n");
return;
}
temp=head;
```

```
if ( head->data = ele)
{
head = head -> next;
return;
}
while ( temp -> next ! = NULL)
{
if ( temp-> next -> data == ele )
{
del = temp ->next;
if (del-> next == NULL)
temp -> next = NULL;
else
temp ->next; del -> next;
}
else
temp = temp -> next;
}
if(del== NULL)
{
printf(" element not found in the list \n");
return;
}
}
void display()
{
struct node *ptr=NULL;
ptr = head;
if (ptr == NULL)
{
printf (" list empty!!!\n");
}
```

```
else
{
while (ptr != NULL)
{
printf("%d", ptr->data);
ptr = ptr->next;
}
}
printf("\n");
}
```