Expt. No. 9

Aditi Akarsh
1BM19CS007

Date

Page No. 1

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node
{ int data;
  struct node* prev;
  struct node* next;
} Node;
Node *head=NULL;
void doublyll();
void insertnode(int);
void insertnodetoleft();
void deletespecificvalue();
void displaylist();

int main()
{ doublyll();
  return 0; }
void doublyll()
{ int ch=0;
printf("1. Enter node 2. Enter node to left 3. delete
        4. display 5. exit Choice:");
  scanf("%d", &ch);
  switch(ch)

{
case 1: insertnode(0);
        break;
        case 2: insertnode(1);
case 3: deletespecificvalue();
        break;
```

```
case 4: displaylist();
        break;
case 5: exit(0);
default: printf("Error");
         doublyll();
}
doublyll();
}

void insert(int flag)
{
Node *newnode;
newnode=(Node*)malloc(sizeof(Node
printf("Enter element");
scanf("%d",&newnode->data);
if(head==NULL)
{
head==NULL)
{head=newnode;
 newnode->next=NULL;
 newnode->prev=NULL;
 printf("First node");
 doublyll();
}
if(flag==0)
{Node *temp=head;
for(temp;(temp->next)!=NULL;
 temp=temp->next);
```

```
        temp -> next = newnode;
        newnode -> prev = temp;
        newnode -> next = NULL;
    }
    else if (flag == 1)
        insertnodetoleft(newnode);
    }

void insertnodetoleft(node *tempNew)
{   int ele; char ch;
    printf("Enter node who's left");
    scanf("%d", &ele);
    node *temphead;
    if(head -> data == ele)
    {
        tempNew -> next = head;
        tempNew -> prev = NULL;
        head = tempNew;
        printf("Node created");
        doublyll();
    }
    for(temp; temp != NULL; temp = temp -> next)
    {   if(temp -> data == ele)
        {   tempNew -> next = temp;
            tempNew -> prev = temp -> prev;
            (temp -> prev) -> next = tempNew;
            temp -> prev = tempNew;
            printf("Node created");
            doublyll();
        }
    }
```

Teacher's Signature

```
fflush(stdin);
scanf("%c",&ch);
if(ch=='Y'||ch=='y')
insertnodetoleft(tempNew);
else
{ free(tempNew);
  printf("Node creation failed");
  doublyll();
}}

void deletespecificvalue()
{ if(head==NULL)
  { doublyll(); }
int ele;
scanf("%d",&ele);
  Node *temp=head;
  if(head->next=NULL)
{ if(head->data==ele)
  {free(temp);
   head=NULL;
   doublyll(); }}
else { for(temp; temp!=NULL; temp=temp->next)
  {
  if(temp->data==ele)
    { if(temp->next==NULL)
    {(temp->prev)->next=NULL;
     free(temp);
     doublyll();
    }
```

```
if (temp -> prev == NULL)
{   (temp -> next) -> prev = NULL;
    head = head -> next;
    doublyll();
}
else { (temp -> prev) -> next = temp -> next;
       (temp -> next) -> prev = temp -> prev;
       free (temp);
       doublyll();
     }
   }
}

void displaylist()
{ if (head == NULL)
  { printf ("Empty list");
    doublyll();
  }
node * temp2 head;
for (temp; temp != NULL; temp2 temp -> next
  {
    printf ("%d", temp -> data);
  }
doublyll();
}
```