LAB 9

Doubly Linked List

① Inserting

```
void insertNode()
{
    if (head == NULL)
    {   head = newnode;
        newnode->next = newnode->prev = NULL;
        return;
    }
    Node *temp = head;
    for (temp; (temp->next) != NULL; temp = temp->next);
    temp->next = newnode;
    newnode->prev = temp;
    newnode->next = NULL;
}
```

② Inserting to right of a node

```
void insertNodeToRight(Node *tempnew)
{
    int ele;
    char choice;
    printf("\n Enter the ele whose right you want
            to insert to : ");
    scanf("%d", &ele);
    Node *temp = head;
    for (temp; templ = NULL; temp = temp->next)
    {
        if (temp->data == ele)
        {
            if (temp->next != NULL)
            {
                tempNew->next = temp->next;
                tempNew->prev = temp;
                temp->next->prev = tempNew;
                temp->next = tempNew;
            }
            return;
```

```
        else
        {
            tempNew ->next =NULL;
            tempNew  ->prev = temp;
            temp->next=tempNew;
            return;
        }
    }
}
printf("In Element Not found! press Y to Try again: ");
scanf("%c", &choice);
if( choice =='y' || choic = ='Y')
insertNodeForRight (tempNew);
else
free(tempNew);
}
```

③ void displaylist()

```
{
    if (head ==NULL)
    {
        printf("\n Empty list! \n");
        return;
    }
    Node *temp=head;
    printf("\n the list contains: ");
    for (temp; temp!= NULL; temp=temp->next)
    printf("%d", temp->data);
    return;
}
```

④ delebion:

if last node

```
temp->prev->next=NULL;
free(temp);
```

if first node:

```
temp->next->prev=NULL;
head=head->next;
```

if in-between:

```
temp->prev->next = temp->next;
temp->next->prev = temp->prev;
free(temp)
```