Name - Nitin Sikarwar
Section - C (38)
Uni. Roll no - 2015004S1

DATE:

(Singly LL)

```
impat java.util.*;
public class SinglyLL {
    class Node {
        int data
        Node next;
        Node( int data) {
            this.data = data;
            this.next = next;
        }
    }
    static Node head;
    public void insertatbeg (int data) {
        Node n = new Node (data);
        if (head == null) {
            head = n;
            return;
        }
        n.next = head;
        head = n;
    }
    public void insertatlast (int data) {
        Node n = new Node (data);
        if (head == null) {
            head = n;
            return;
        }
        Node curr = head;
        while( curr.next != null) {
            curr = curr.next;
        }
        curr.next = n;
    }
```

Name - Nitin Sikarwar
Section - C (38)
Uni Rollno - 2015005151

Dhoom
PAGE NO.:
DATE: / /

```
public void deleteatbeg() {
    if(head == null) {
        System.out.println("LL is Empty");
        return;
    }
    head = head.next;
}
public void deleteatlast() {
    if(head == null) {
        System.out.println("LL is Empty");
        return;
    }
    Node
    else if(head.next == null) {
        head = null;
        return;
    }
    else {
        Node prev = head;
        Node curr = head;
        while(curr.next != null) {
            prev = curr;
            curr = curr.next;
        }
        prev.next = null;
    }
}
public static void main(String args[]) {
```

Nitin Sikarwar
Sec- C (38)

```
public int count() {
    Node temp = head;
    if (head == null)
        return 0;
    else {
        int c = 1
        while (temp.next != null) {
            c++;
            temp = temp.next;
        }
        return c;
    }
}
public void insertAtpos (int pos, int data);
    Node n = new Node (data);
    if (head == null) {
        if (pos != 1) {
            s.o.p(" Invalid pos");
        }
        else {
            head = n;
        }
    }
    else {
        if (pos == 1) {
            n.next = head;
            head = n;
        }
        else if (pos > count() + 1) {
            s.o.p(" Invalid pos");
        }
```

```
else {
    Node temp = head;
    int c=1;
    while (temp.next != null) {
        if (pos-1 == c)
            break;
        c++;
        temp = temp.next;
    }
    n.next = temp.next;
    temp.next = n;

    }
  }
}

public void deleteatpos (int pos) {
    if (head == null) {
        return;
    else {
        if (pos==1) {
            if (head.next == head) {
                head = null;
            }
            else {
                Node temp = head
                head = head.next;
            }
        } }
        else if (pos > count()) {
```

Nitin Sikarwar
Sec-C(38)

```
        s.o.pl("Invalid pos");
    }
    else {
        Node temp = head;
            int c=1;
        while (temp.next != null) {
            if(pos-1 == c) {
                    break;
            }
            c++;
            temp = temp.next;
        }
        temp.next = temp.next.next;
    }
}
```

Nitin Sikarwar
Section (c)

```
public void printll() {
        Node curr = head;
        while( curr != null) {
                System.out.println(" curr.data +
                curr = curr.next;
        }
}
public static void main (String args[]) {
        Singlyll L = new Singlyll();
        L.insertatbeg(2);
        L.insertatbeg(1);
        L.insertatlast(3);
        L.deleteatlast();
        L.print();
    }
}
```

Singly Linkedlist using Tail :-

```
import java.util.*;
public class Singly {
    class Node {
        int data;
        Node next;
        Node(int data) {
            this.data = data;
            this.next = next;
        }
    }
    static Node head;
    static Node tail;
```

Nitin Sikarwar
Section - C(30)


```
public void insertAtBeg(int data){
    Node n = new Node(data);
    if(head==null){
        head = n;
        tail = n;
        return;
    }
    n.next = head;
    head = n;
}
public void insertAtLast(int data){
    Node n = new Node(data);
    if(head==null){
        head = n;
        tail = n;
        return;
    }
    tail.next = n;
    tail = n;
}
public void deleteAtBeg(){
    if(head==null){
        System.out.println("List is Empty");
        return;
    }
    head = head.next;
}
public void deleteAtLast(){
public static void main(String args[]){
    Singly L = new Singly();
```

```
l.insertAtBeg(2);
l.insertAtBeg(1);
l.deleteAtBeg();
}
}
```

3. Circular Linked List :-

```
import java.util.*;
public class CLL {
    class Node {
        int data;
        Node next;
        Node(int data) {
            this.data = data;
            this.next = next;
        }
    }
    static Node head;
    public void insertAtBeg(int data) {
        Node n = new Node(data);
        if(head == null) {
            head = n;
            n.next = head;
            return;
        }
        Node curr = head;
        while(curr.next != head) {
            curr = curr.next;
        }
```

```
        n.next = head;
        head = n;
        curr.next = n;
    }
    public void insertAtLast(int data){
        Node n = new Node(data);
        if(head == null){
            head = n;
            n.next = head;
            return;
        }
        Node curr = head;
        while(curr.next != head){
            curr = curr.next;
        }
        curr.next = n;
        n.next = head;
    }
    public void deleteAtBegin(){
        if(head == null){
            System.out.println("List is Empty");
            return;
        }
        Node temp = head;
        while(temp.next != head){
            temp = temp.next;
        }
        temp.next = head.next;
        head = head.next;
}
```

Nitin Sikarwar
Sec - L (38)

PAGE NO.:
DATE:

```
public void deleteatlast() {
    if (head == null) {
        return;
    }
    Node temp = head;
    while (temp.next.next != head) {
        temp = temp.next;
    }
    else if (head.next == head) {
        head = null;
        return;
    }
    else {
        Node temp = head;
        while (temp.next.next != head) {
            temp = temp.next;
        }
        temp.next = head;
    }
}
public void print() {
    Node curr = head;
    ~~while (curr != ~~
    do {
        System.out.println(curr.data);
        curr = curr.next;
    } while (curr != head);
}
public static void main (String args []) {
    CLL list = new CLL();
    list.insertatlast();
    list.print();
}
```

Nitin Sikarwar
Sec- CC38)

Dhoom
PAGE NO.:
DATE: / /

Circular Linkedlist using Tail :-

```
import java.util.*;
public class CLLTail{
    class Node{
        int data;
        Node next;
        Node(int data){
            this.data=data;
            this.next=next;
        }
    }
    static Node head;
    static Node tail;
    public void insertatbeg(int data){
        Node n=new Node(data);
        if(head==null){
            head=n;
            tail=n;
            n.next=head;
            return;
        }
        tail.next=n;
        n.next=head;
        head=n;
    }
    public void insertatlast(int data){
        Node n=new Node(data);
```

Nitin Sikarwar
Sec - C(30

PAGE NO
DATE :

```
if(head == null) {
    head = newnode;
    tail = newnode;
}

if (head == null) {
    head = n;
    tail = n;
    n.next = head;
    return;
}
tail.next = newnode;
n.next = head;
tail = n;
}
public void deletefirst() {
    if (head == null)
    {   return;
    }
    tail.next = head.next;
    head = head.next;
}
public void deletelast() {
    if (head == null) {
        return;
    }
    Node
    else if (head.next == null) {
        tail.next = null;
    }   head = null;
```

```
else {
    Node curr = head;
    while(curr.next.next != head) {
        curr = curr.next;
    }
    curr = curr.nx
    curr.next = head;
    tail = curr;
}
}

public void print() {
    Node curr = head;
    do{
        System.out.println(curr.data);
    } while(curr = curr.next;
    } while (curr != head);
}

public static void main (String args[]) {
    CLLTail obj = new CLLTail();
    obj.insertatbeg(3);
    obj.deletelast();
    obj.print();
}
}
```

Nitin Sikarwar
Sec - C (38)

5: Doubly Linked List :-

```
import java.util.*;
public class DLL {
    class Node {
        int data;
        Node prev;
        Node next;
        Node(int data) {
            this.data = data;
            this.prev = prev;
            this.next = next;
        }
    }
    static Node head;
    public void insertatbeg(int data) {
        Node n = new Node(data);
        if(head == null) {
            head = n;
            return;
        }
        n.next = head;
        head.prev = n;
        head = n;
    }
    public void insertatlast(int data) {
        Node n = new Node(data);
        if(head == null) {
            head = n;
            return;
        }
```

Nitin Sikarwar
Sec - C (38)

Dhoom
PAGE NO.:
DATE: / /

```
        Node curr = head;
        while(curr.next != null) {
            curr = curr.next;
        }
        curr.next = n;
        n.prev = curr;
    }
    public void deleteFirst() {
        if(head == null) {
            return
        }
        head = head.next;
        head.prev = null;
    }
    public void deleteLast() {
        if(head == null) {
            return;
        }
        Node curr = head;
        while(curr.next != null)
        {
            curr = curr.next;
        }
        curr.prev.next = null;
    }
    public void print() {
        Node curr = head;
        while (curr != null) {
            System.out.println(curr.data);
            curr = curr.next;
        }
}
```

Nitin Sikarwar
Sec - C(30)

```
public static void main(String args[]){
    DLL list = new DLL();
    list.insertatbeg(2);
    list.insertatlast(3);
    list.deletefirst();
    list.print();
  }
}
```

6 Doubly LinkedList Using Tail :—

```
import java.util.*;
public class DLLTail {
    class Node {
        int data;
        Node prev;
        Node next;
        Node(int data) {
            this.data = data;
            this.prev = prev;
            this.next = next;
        }
    }
    static Node head;
    static Node tail;
    public void insertatbeg(int data) {
        Node n = new Node(data);
        if(head == null) {
            head = n;
```

```
        tail = n;
        return;
    }
    n.next = head;
    head.prev = n;
    head = n;
}
public void insertatlast(int data){
    Node n = new Node(data);
    if(head == null){
        head = n;
        tail = n;
        return;
    }
    tail.next = n;
    n.prev = tail;
    tail = n;
}
public void deletefirst(){
    if(head == null){
        return;
    }
    head = head.next;
    head.prev = null;
}
public void deletelast(){
    if(head == null){
        return;
    }
    tail.prev.next = null;
}
```

Nitin Sikarwar
Sec - C (38)

```
public void print() {
    Node curr = head;
    while(curr != null) {
        System.out.println(curr.data);
        curr = curr.next;
    }
}
public static void main (String args[]) {
    DLLtail List = new DLLtail();
    List.insertatfirst(2);
    List.insertatlast(3);
    List.insertatlast(4);
    List.deletelast();
    List.print();
}
}
```

Thank you