SATYAM TRIPATHI
202151141

CS ASSIGNMENT 5

CODE

```
class Node{
    int data;
    Node next;
    public Node(int v){
        data=v;
        next=null;
    }
    public Node(){}
}
class LinkedList {
    protected Node first;
    protected int size;
    public LinkedList(){
        first=null;
        size=0;
    }
    public boolean isEmpty(){
        if(size==0)
        return true;
        return false;
    }
    public int size(){
        return size;
    }
    public void checkIndex(int index){
        if(index<0 || index>size){
            throw new IndexOutOfBoundsException("index = "+index+" for
size = "+size);
        }
    }
    public int get(int index){
        checkIndex(index);
        Node current=first;
        for(int i=0;i<index;i++)
        {
            current=current.next;
```

SATYAM TRIPATHI
202151141

```
        }
        return current.data;

    }
    public int indexOf(int element){
        Node current=first;
        int c=0;
        while(current != null){
            if(current.data==element)
            break;
            current=current.next;
            c++;
        }
        if(current!=null)
        return c;
        return -1;
    }
    public void add(int element,int index){
        checkIndex(index);

        size++;
        Node temp=first;
        if(index==0)
        {
            first=new Node(element);

            return;
        }
       for(int i=1;i<index;i++)
       {
            temp=temp.next;
       }
       Node n1=new Node(element);
       n1.next=temp.next;
       temp.next=n1;

    }
    public void addFront(int element){
        add(element, 0);
        size++;
```

SATYAM TRIPATHI
202151141

```
    }
    public void addRear(int element){
        add(element, size);
        size++;
    }
    public int remove(int index){
        checkIndex(index);
        Node temp=first;
        size--;
        for(int i=1;i<index;i++){
            temp=temp.next;
        }
        int v=temp.next.data;
        temp.next=temp.next.next;
        return v;
    }
    public void show(){
        Node temp=first;
        while(temp!=null){
            System.out.print(temp.data+" ");
            temp=temp.next;
        }
        System.out.println();
    }
    public void reverse(){
        Node current=first;
        Node prev=null;
        Node nxt;
        while(current!=null){
            nxt=current.next;
            current.next=prev;
            prev=current;
            current=nxt;

        }
        first=prev;
    }
    public int search(int find){
        Node current=first;
        int c=0;
```

SATYAM TRIPATHI
202151141

```
        while(current!=null)
        {
            if(current.data==find){
                break;
            }
            c++;
            current=current.next;
        }
        if(current==null)
        return -1;
        return c;
    }

}
public class LinkedListImplement{
    public static void main(String[] args) {
        LinkedList list=new LinkedList();

        list.add(1,0);
        list.add(2,1);
        list.add(2,2);
        list.add(3,3);
        list.add(4,4);
        list.add(5,5);
        list.addRear(6);
        System.out.print("Size of the list => "+list.size()+"\n");
        list.show();
        System.out.println("Index of 4 => "+list.indexOf(4));
        System.out.println("Get element at index 2 => "+list.get(2));
        System.out.print("Deleting element at index 2 => ");
        list.remove(2);
        list.show();
        System.out.println("Element 4 found at index => "+list.search(4));
        System.out.print("Reversing the Linked List => ");
        list.reverse();
        list.show();

    }
}
```

SATYAM TRIPATHI
202151141

OUTPUT

```
Size of the list => 8
1 2 2 3 4 5 6
Index of 4 => 4
Get element at index 2 => 2
Deleting element at index 2 => 1 2 3 4 5 6
Element 4 found at index => 3
Reversing the Linked List => 6 5 4 3 2 1
```

SATYAM TRIPATHI
202151141

CODE

```
class Node{
    Node next;
    int data;
    public Node(int v){
        next=null;
        data=v;
    }
}
class CircularLinkedList {
    Node first,current;
    void add(int element){
        if(first==null){
            first=new Node(element);
            current=first;
            first.next=first;
            return;
        }
        else{
            Node n1=new Node(element);
            current.next=n1;
            current=current.next;
            current.next=first;
        }
    }
    void show(){
        Node temp=first.next;
        System.out.print(first.data+" ");
        while(temp!=first){
            System.out.print(temp.data+" ");
            temp=temp.next;
        }
        System.out.println();
    }
    void remove(int element){        //removes all matching elements
        Node temp=first.next,prev=first;
        if(first.data==element)
        {
            current.next=first.next;
```

SATYAM TRIPATHI
202151141

```
            first=first.next;
            return;
        }
        while(temp!=first){

            if(temp.data==element){
                prev.next=temp.next;
                temp=temp.next;
                continue;
            }
            temp=temp.next;
            prev=prev.next;
        }
    }

}
public class ImplementationCircular{
    public static void main(String[] args) {
        CircularLinkedList list=new CircularLinkedList();
        list.add(1);
        list.add(2);
        list.add(3);
        list.add(4);
        list.add(5);
        list.add(5);
        list.show();
        list.remove(5);
        list.show();
    }
}
```

SATYAM TRIPATHI
202151141

OUTPUT

```
1 2 3 4 5 5
1 2 3 4
```

SATYAM TRIPATHI
202151141

CODE

```
class Node{
    public Node next;
    public Node prev;
    int data;
    public Node(int v){
        next=null;
        prev=null;
        data=v;
    }
}
class DoubleLL{
    protected Node first;
    protected int size;
    public DoubleLL(){
        first=null;
        size=0;
    }
    public boolean isEmpty(){
        if(size==0)
        return true;
        return false;
    }
    public int size(){
        return size;
    }
    public void checkIndex(int index){
        if(index<0 || index>size){
            throw new IndexOutOfBoundsException("index = "+index+" for
size = "+size);
        }
    }
    public int indexOf(int element){
        Node current=first;
        int c=0;
        while(current != null){
            if(current.data==element)
            break;
            current=current.next;
```

SATYAM TRIPATHI
202151141

```
            c++;
        }
        if(current!=null)
        return c;
        return -1;
    }
    public void add(int element,int index){
        checkIndex(index);

        size++;
        Node temp=first;
        if(index==0)
        {
            first=new Node(element);

            return;
        }
       for(int i=1;i<index;i++)
       {
            temp=temp.next;
       }
       Node n1=new Node(element);
       n1.next=temp.next;
       n1.prev=temp;
       temp.next=n1;
       n1=n1.next;
       temp=temp.next;
       try{
       n1.prev=temp;
       }
       catch(NullPointerException E){

       }


    }
    public void addFront(int element){
        add(element,0);
        size++;
    }
```

SATYAM TRIPATHI
202151141

```
    public void addRear(int element){
        add(element,size-1);
        size++;
    }
    public int remove(int index){
        checkIndex(index);
        Node temp=first;
        size--;
        for(int i=1;i<index;i++){
            temp=temp.next;
        }
        int v=temp.next.data;
        temp.next=temp.next.next;
        temp.next.next.prev=temp;
        return v;
    }
    public void show(){
        Node temp=first;
        while(temp!=null){
            System.out.print(temp.data+" ");
            temp=temp.next;
        }
        System.out.println();
    }
    public void reverse(){
        Node current=first;
        Node prev=null;
        Node nxt;
        while(current!=null){
            nxt=current.next;
            current.next=prev;
            prev=current;
            current=nxt;

        }
        first=prev;
    }

}
public class ImplementationDoubleLL {
```

SATYAM TRIPATHI
202151141

```
    public static void main(String[] args) {
        DoubleLL list=new DoubleLL();
        list.addFront(1);
        list.add(1,1);
        list.add(2, 2);
        list.add(3, 3);
        list.add(4, 4);
        list.addRear(5);
        list.show();
        list.remove(1);
        list.show();

    }
}
```

OUTPUT

```
1 1 2 3 4 5
1 2 3 4 5
```