Name: ZOHAIB HASSAN SOOMRO

RollNo#: 19SW42

Subject: DSA

```
class Node{

        private int data;
        private Node next;

        Node(int data){
                this.data=data;
        }

        Node(int data ,Node list){
    this.data=data;
    this.next= list;
        }

/////////Sum function returns sum of all elements of a list
        public static int sum(Node list){
                int sumOfList=0;
                while(list!=null){
                        sumOfList+=list.data;
                        list=list.next;
                }
```

```
        return sumOfList;
    }

//////////////CopyList function returns a copy of source list///////////
    public static Node copyList(Node sourcelist){
        Node newList= new Node(sourcelist.data);
        Node p=newList;
        while((sourcelist=sourcelist.next)!=null){
            p=p.next=new Node(sourcelist.data);

        }
        return newList;
    }


//////////////Insert Function//////////////////
    public static Node insert(Node p1,int x){
        if (x<p1.data || p1==null) {        ///Condition for inserting number before first element if number is less than first element
            p1=new Node(x,p1);
            return p1;
        }
        Node p=p1;
```

```
            while(p.next!=null){
                if (p.next.data>x)
                    break;
                p=p.next;   //iterate the linkedlist
            }
            p.next= new Node(x,p.next);
            return p1;
        }



//////////method to delete///
        public static Node delete(Node p,int val){
            if (val<p.data || p==null) return p;        //val is not in list
            if (val==p.data)        return p.next;    //val is the first element

            for (Node i=p; i!=null;i=i.next ) {
                if (val< i.next.data) break;  //x is not in list
                if (val==i.next.data) {
                    i.next=i.next.next;
                    break;
                }
            }
```

```
        return p;
    }

//////////////////Delete last element method
    public static Node deleteLast(Node p){
        if (p==null) return p;
        if (p.next==null) return p=null;

        for (Node i=p;i!=null;i=i.next) {
            if (i.next==null) return i=null;
            if (i.next.next==null) i.next=null;
        }
        return p;
    }

//////////////////////Concat method to concatinate 2 lists ////
    public static Node concat(Node list1,Node list2){
        Node newList=new Node(list1.data);
        Node p=newList;
        while((list1=list1.next)!=null){
            p=p.next=new Node(list1.data);
        }
```

```
        while(list2!=null){
                p=p.next=new Node(list2.data);
                list2=list2.next;
        }

        return newList;
    }

//////////////////////////swap method to swap two values
    public static void swap(Node list,int i,int j){
        if(i==j || i<1 || j<1)              return;
        Node temp1=null,temp2=null;
        int var;
            for (Node x=list;x!=null ;x=x.next) {
                if(i--==1)      temp1=x;
                if(j--==1)   {temp2=x;        break;}
            }
        var=temp1.data;
        temp1.data=temp2.data;
        temp2.data=var;
```

```
    }


//////////main method starts
    public static void main(String[] args) {
        Node start = new Node(22);
        Node p=start;

        for (int i=1;i<3;i++) {
            p=p.next=new Node(22+11*i);
        }
        start=insert(start,23);
        start=insert(start,8);
        start=insert(start,10);
        start=insert(start,9);
        start=insert(start,35);

        //int counter=0;
        System.out.print("1st List:");
        for (p=start;p!=null;p=p.next) {
            System.out.print(" "+p.data);
            //System.out.println("Address: "+p);
            //counter++;
```

```
        }
        //System.out.println("\nThere are "+counter+" elements in list");
/*
//////////////////////////calculating sum
        System.out.println("Sum of list is: "+sum(start));

////////////coping start list into anotherlist*/
        Node anotherList= copyList(start);
        anotherList=insert(anotherList,5);   //inserting an extra element
        System.out.print("\n2nd List:");
        for (p=anotherList;p!=null;p=p.next ) {
            System.out.print(" "+p.data);
        }

        /*System.out.println("\nAfter deleting");
        start=delete(start,10);
        for (p=start;p!=null;p=p.next)
            System.out.println(p.data);

        System.out.println("\nAfter deleting last element");
        start=deleteLast(start);
        for (p=start;p!=null;p=p.next)
```

```
                System.out.println(p.data);*/


                Node list= concat(start,anotherList);
                System.out.print("\nConcatinated list: ");
                for (p=list;p!=null;p=p.next ) {
                        System.out.print(" "+p.data);
                }


                swap(start,1,5);
                System.out.println("\nAfter Swapping 1st and 5th element of 1st list: ");
                for (p=start;p!=null;p=p.next ) {
                        System.out.print(" "+p.data);
                }
                System.out.println();
        }   //main method ends here
}       //class ended
```