```
package org.singlelist.example1.cw;

import java.util.Scanner;

class SinglyList{

	static class Node{
		private int data;
		private Node next;

		public Node() {
			data =0;
			next = null;
		}

		public Node(int val) {
			data = val;
			next = null;
		}

	}

	private Node head; //head ka pointer bana daala

	public SinglyList() {
		head = null;
	}

	public void disp() {

		System.out.println("******List elements are :******");
```

```
        Node trav = head ;
        while(trav!=null) {
        System.out.println(trav.data);
        trav = trav.next;
        }

        System.out.println(""); //runs when head == null;
}

public void addLast(int val) {
        //created new node
        Node nn = new Node(val);
        //trav ko head ka matlab first element ka address do.
        Node trav = head;

        //find until trav.next == null jaise null mila
        //to usme hi trav.next ko nn ka address do.
        if(head == null)
        {
                head = nn; //head main hi replace karo value ko.
        }
        else {
        //traverse
        while(trav.next!=null) {
                trav = trav.next;
        }
        trav.next = nn;

        }
}
```

```
	public void addFirst(int val) {
		//create a new node
		Node nn = new Node(val);
		//add first mhnjhe apn ko  nn ka address main next jo current first node
		//hai uska address to dalna hi padega na.
		nn.next = head;
		head = nn;

	}

	public void addElement(int val , int pos) {
		if(head == null || pos<=1) {
			addFirst(val);
		}
		else {
			//create a new node
			Node nn = new Node(val);
			Node trav = head; //head == null declared in constructor
			//abhi traverse karna padega na apn ko
			for(int i = 1; i<pos-1;i++)//first new node main jo previous node ke next ka
add hai vo daalna padega
			trav = trav.next;
			//ye hai jisme apn previous main nn ka address dal rahe hai
			nn.next = trav.next ;
			trav.next = nn;//iska trav.next ka pointer point karega nn ko.
	}

}

	public void delFirst() {
		if(head == null)
```

```
                throw new RuntimeException("List is empty");
        head = head.next;

}

public void delLast() {

        if(head == null)
                throw new RuntimeException("Invalid hai baba");
        else {
                Node temp = null;//this will be behind the trav
                //ie temp... trav = temp . trav.next bhi hona chahiye.
                Node trav = head;
        while(trav.next !=null)
        {
                temp=trav;//ye piche piche
                trav = trav.next; //ye aage aage.
        }
        temp.next = null;

}
}

public void delElement(int pos) {
        if(pos ==1)
                delFirst();

        if(head == null)
                throw new RuntimeException("kaise delete kare hai hi nahi");
        else {
                Node temp = null;
```

```
                    Node trav = head;
                    for(int i= 0; i<pos ; i++ )
                    {
                        temp = trav;
                        trav = trav.next;
                    }

                }

        }

        public void reverse() {
    Node prev = null;//before curr
    Node current = head; //now this is the one who has head.
    Node next = null;//after curr

    while (current != null) {
        next = current.next;
        current.next = prev;
        prev = current;
        current = next;
    }
    head = prev;
}

        public void middleEle() {

            //hare and tortoise algo....
            //car vala example ki 100km/hr vali car kabhi bhi 50km/hr vale car se jyada tej
            // ie 2 times jyada tej hi bhagegei na
            if (head == null) {
```

```
            System.out.println("The list is empty.");
            return;
        }
            Node slowPtr = head;
            Node fastPtr = head;

            while(fastPtr!= null && fastPtr.next!= null)
            {
                slowPtr = slowPtr.next;
                fastPtr = fastPtr.next.next; //two times.
            }

            //head = slowPtr.data;
            System.out.println("the last ele is : " +slowPtr.data);
    }


}
public class SingleListMain {

    public static void main(String[] args) {
            int choice, val, pos ;
            SinglyList list = new SinglyList();
            Scanner sc = new Scanner(System.in);
            do {
                System.out.println("0.Exit");
                System.out.println("1.Add First element");
                System.out.println("2.Add Last element");
                System.out.println("3.Add between element");
                System.out.println("4.del first element");
                System.out.println("5.del last element");
```

```
System.out.println("6.del between  element");
System.out.println("7.display list");
System.out.println("8. Reverse the list");
System.out.println("9. find  middle of the list");
System.out.println("enter the choice : ");

choice = sc.nextInt();
switch(choice)
{
case 1://add first
        System.out.print("Enter the first element : ");
        val = sc.nextInt();
        list.addFirst(val);
        break;
case 2: //add last elements
        System.out.print("enter the last element ");
        val = sc.nextInt();
        list.addLast(val);
        break;
case 3://add between elements.
        System.out.println("enter the val");
        val = sc.nextInt();
        System.out.println("enter the pos");
        pos = sc.nextInt();
        list.addElement(val,pos);
        break;
case 4://delete first element
        list.delFirst();
        System.out.println("element is deleted");
        break;
case 5://delete last element.
```

```
                    System.out.println("");
                    list.delLast();
                    break;
                case 6://delete between elements.
                    System.out.println("Del this position");
                    pos = sc.nextInt();
                    list.delElement(pos);
                    break;
                case 7: //display
                    list.disp();
                    break;
                case 8:
                    System.out.println("the reversed list is as follows");
                    list.reverse();
                    list.disp();
                    break;
                case 9:
                    System.out.println("the middle element is ");
                    list.middleEle();
                    break;
                }
            }while(choice!=0);
            sc.close();

        }

}
```