```
public class Implementation{
    Node head;

    class Node{
        int data;
        Node next;

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

    Implementation(){
        head = null;
    }
```

```
public void insertAtBeginning(int val){
    Node newNode = new Node(val);

    if(head == null)
    {
        head=newNode;
    }
    else
    {
        newNode.next=head;
        head=newNode;
    }
}

public void insertAtposotion(int pos,int val)
{
    if (pos==0)
    {
        insertAtBeginning(val);
```

```
        return;
    }
    Node newNode= new Node(val);
    Node temp = head;

    for(int i=0 ; i<pos-1;i++)
    {
        temp=temp.next;
        if (temp==null) {
        //
        throw new IllegalArgumentException("invalid
position : " +pos);
        }
    }
    newNode.next=temp.next;
    temp.next=newNode;
  }
```

```
  public void deleteAtPosition(int pos)
  {

      if(pos==0)
      {
          deleteAtBeginning();
          return;
      }

      Node temp = head;
      Node prev = null;

      for(int i=0;i<=pos-1;i++)
      {
          prev=temp;
          temp=temp.next;
      }
      prev.next=temp.next;
}
```

```
public void deleteAtBeginning()
{
    if(head == null)
    {
        //Syssout;
        throw new IndexOutOfBoundsException("index");
    }
    head=head.next;
}

public void deleteAtLast() {
    if (head == null) {
        throw new IndexOutOfBoundsException("Index");
    }

    Node temp = head;
    Node prev = null;
```

```
    while (temp.next != null) {
        prev = temp;
        temp = temp.next;
    }

    if (prev == null) {
        // The list had only one element
        head = null;
    } else {
        prev.next = null;
    }
}
```

```
public void display()
{
    Node temp =head;
    while (temp != null)
    {
        System.out.print(temp.data +" ");
        temp=temp.next;
    }
}
public void reverse()
{
    Node prev=null;
    Node current=head;
    Node next=null;
```

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

```
public class Demo{
    public static void main(String[] args) {

        Implementation i = new Implementation();
        i.insertAtBeginning(4);
        i.insertAtBeginning(2);
        i.insertAtBeginning(1);
        i.display();
        System.out.println();
        i.insertAtposotion(2,3);
        i.insertAtposotion(3,5);
        i.insertAtposotion(0,0);
       //  System.out.println();
        i.display();
       // i.insertAtposotion(9,9);
       i.deleteAtPosition(2);
       i.deleteAtPosition(4);
       System.out.println();
        i.display();

        i.deleteAtBeginning();
        System.out.println();
        i.display();
```

```
        System.out.println();
       // i.deleteAtPosition(8);
        i.display();
        System.out.println();
        i.deleteAtLast();
        i.display();

        System.out.println();
        i.reverse();
        i.display();
    }
}
```

1 2 4

0 1 2 3 5 4

0 1 3 5

1 3 5

1 3 5

1 3

3 1