Doubly Linked List (DLL)

Code:

```
class DoublyLinkedList {
    class Node {
        int data;
        Node prev, next;

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

    Node head;

```

```
    // Insert at beginning
    public void insertAtBeg(int data) {
        Node newNode = new Node(data);
        newNode.next = head;
        if (head != null)
            head.prev = newNode;
        head = newNode;
    }
```

```java
    // Insert at end
    public void insertAtEnd(int data) {
        Node newNode = new Node(data);
        if (head == null) {
            head = newNode;
            return;
        }
        Node temp = head;
        while (temp.next != null)
            temp = temp.next;
        temp.next = newNode;
        newNode.prev = temp;
    }

    // Insert at position
    public void insertAtPos(int data, int pos) {
        if (pos == 0) {
```

```
            insertAtBeg(data);
            return;
        }
        Node newNode = new Node(data);
        Node temp = head;
        for (int i = 0; i < pos - 1 && temp != null; i++)
            temp = temp.next;
        if (temp == null) {
            System.out.println("Position out of bounds.");
            return;
        }
        newNode.next = temp.next;
        newNode.prev = temp;
        if (temp.next != null)
            temp.next.prev = newNode;
        temp.next = newNode;
    }
```

```java
    // Delete at beginning
    public void deleteAtBeg() {
        if (head == null) return;
        head = head.next;
        if (head != null)
            head.prev = null;
    }

    // Delete at end
    public void deleteAtEnd() {
        if (head == null) return;
        if (head.next == null) {
            head = null;
            return;
        }
        Node temp = head;
        while (temp.next != null)
```

```
            temp = temp.next;
        temp.prev.next = null;
    }

    // Delete at position
    public void deleteAtPos(int pos) {
        if (head == null) return;
        if (pos == 0) {
            deleteAtBeg();
            return;
        }
        Node temp = head;
        for (int i = 0; i < pos && temp != null; i++)
            temp = temp.next;
        if (temp == null) {
            System.out.println("Position out of bounds.");
            return;
        }
```

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

    // Display
    public void display() {
        Node temp = head;
        while (temp != null) {
            System.out.print(temp.data + " <-> ");
            temp = temp.next;
```

```
        }
        System.out.println("null");
    }
```

```
    public static void main(String[] args) {
        DoublyLinkedList dll = new DoublyLinkedList();
        dll.insertAtEnd(1);
        dll.insertAtBeg(0);
        dll.insertAtPos(2, 2);
        dll.display();

        dll.deleteAtBeg();
        dll.display();

        dll.deleteAtEnd();
        dll.display();

        dll.deleteAtPos(0);
        dll.display();
    }
}
```

Output:

```
0 <-> 1 <-> 2 <-> null
1 <-> 2 <-> null
1 <-> null
null
```