```
public class Forward {
        class Node {
            int data;
            Node prev;
            Node next;

            public Node(int data) {
                this.data = data;
                this.prev = null;
                this.next = null;
            }
        }

        class DoublyLinkedList {
            private Node head;

            public DoublyLinkedList() {
                this.head = null;
            }

            public void insert(int data) {
                Node newNode = new Node(data);

                if (head == null) {
                    head = newNode;
                } else {
                    Node current = head;
                    while (current.next != null) {
                        current = current.next;
                    }
```

```
            current.next = newNode;
            newNode.prev = current;
        }
    }

    public void traverseForward() {
        Node current = head;
        System.out.print("Doubly Linked List (Forward): ");
        while (current != null) {
            System.out.print(current.data + " ");
            current = current.next;
        }
        System.out.println();
    }

    public void traverseBackward() {
        Node current = head;
        while (current != null && current.next != null) {
            current = current.next;
        }

        System.out.print("Doubly Linked List (Backward): ");
        while (current != null) {
            System.out.print(current.data + " ");
            current = current.prev;
        }
        System.out.println();
    }
}

public class DoublyLinkedListTraversal {
```

```
    public void main(String[] args) {
        DoublyLinkedList list = new DoublyLinkedList();



        list.insert(90);
        list.insert(80);
        list.insert(70);
        list.insert(60);
        list.insert(50);



        list.traverseForward();


        list.traverseBackward();
    }
}
```