Program7: Traverse a doubly linked list in the forward and backward directions

```java
package mounika7;
public class DLL
{
        Node head;
class Node
{
                        int data;
                        Node prev;
                        Node next;
Node(int d)
{
data = d;
}
        }
public void push(int new_data)
        {
Node new_Node = new Node(new_data);
new_Node.next = head;
                        new_Node.prev = null;
if (head != null)
                                head.prev = new_Node;
head = new_Node;
        }
public void InsertAfter(Node prev_Node, int new_data)
        {
if (prev_Node == null)
{
                                System.out.println("The given previous node cannot
be NULL ");
                                return;
                }
Node new_node = new Node(new_data);
new_node.next = prev_Node.next;
prev_Node.next = new_node;
new_node.prev = prev_Node;
if (new_node.next != null)
                                new_node.next.prev = new_node;
        }
        void append(int new_data)
        {
Node new_node = new Node(new_data);
                Node last = head;
new_node.next = null;
if (head == null)
{
                                new_node.prev = null;
                                head = new_node;
                                return;
                }
while (last.next != null)
```

```java
                        last = last.next;
        last.next = new_node;
        new_node.prev = last;
        }
    public void printlist(Node node)
        {
                    Node last = null;
                    System.out.println("Traversal in forward Direction");
                    while (node != null)
    {
                            System.out.print(node.data + " ");
                            last = node;
                            node = node.next;
                    }
                    System.out.println();
                    System.out.println("Traversal in reverse direction");
                    while (last != null)
    {
                            System.out.print(last.data + " ");
                            last = last.prev;
                    }
        }
    public static void main(String[] args)
        {
DLL dll = new DLL
dll.append(6);
dll.push(7);
dll.push(1);
dll.append(4);
dll.InsertAfter(dll.head.next, 8);
                System.out.println("Created DLL is: ");
                    dll.printlist(dll.head);
        }
}
```