1. Implement ‘Singly Linked List’ to store set of integer elements

```
package dsa;

public class LinkedList {
	// Represent a node of the singly linked list
	class Node {
		int data;
		Node next;

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

	// Represent the head and tail of the singly linked list
	public Node head = null;
	public Node tail = null;

	// addNode() will add a new node to the list
	public void addNode(int data) {
		// Create a new node
		Node newNode = new Node(data);

		// Checks if the list is empty
		if (head == null) {
			// If list is empty, both head and tail will point to new node
			head = newNode;
			tail = newNode;
		} else {
			// newNode will be added after tail such that tail's next will point to newNode

			tail.next = newNode;
			// newNode will become new tail of the list
			tail = newNode;
		}
	}

	// display() will display all the nodes present in the list
	public void display() {
		// Node current will point to head
		Node current = head;

		if (head == null) {
			System.out.println("List is empty");
			return;
		}
		System.out.println("Nodes of singly linked list: ");
		while (current != null) {
			// Prints each node by incrementing pointer
			System.out.print(current.data + " ");
			current = current.next;
		}
		System.out.println();
	}

	public static void main(String[] args) {

		LinkedList sList = new LinkedList();

		// Add nodes to the list
		sList.addNode(1);
		sList.addNode(2);
		sList.addNode(3);
		sList.addNode(4);
		sList.addNode(5);
		sList.addNode(6);
		sList.addNode(7);

		// Displays the nodes present in the list
		sList.display();
	}
}
```

Output:

```
Nodes of singly linked list:
1 2 3 4 5 6 7
```