2. Write a program to check if given linked list has a loop or not.

Description:  
Sometimes a linked list get corrupt, and two nodes point to the same node, which forms the loop or cycle in the linked list.

public class LinkedList{

private Node head;

private static class Node {

private int value;

private Node next;

Node(int value) {

this.value = value;

}

}

public void addToTheLast(Node node) {

if (head == null) {

head = node;

} else {

Node temp = head;

while (temp.next != null)

temp = temp.next;

temp.next = node;

}

}

public void printList() {

Node temp = head;

while (temp != null) {

System.out.format("%d ", temp.value);

temp = temp.next;

}

System.out.println();

}

public boolean ifLoopExists() {

Node fastPtr = head;

Node slowPtr = head;

while (fastPtr != null && fastPtr.next != null) {

fastPtr = fastPtr.next.next;

slowPtr = slowPtr.next;

if (slowPtr == fastPtr)

return true;

}

return false;

}

public static void main(String[] args) {

LinkedList list = new LinkedList();

list.addToTheLast(new Node(5));

list.addToTheLast(new Node(6));

list.addToTheLast(new Node(7));

list.addToTheLast(new Node(1));

list.addToTheLast(new Node(2));

list.printList();

System.out.println("Loop existed-->" + list.ifLoopExists());

}

}

**Output:**

