/\*\*

\* Class LL describes linked list of int type

\*

\* @author (Erik Marrero)

\* @version (Mar 26, 2020)

\*/

public class LL

{

private Node list;

//Constructor for objects of class LL

public LL()

{

list = null;

}

public void addFront(int num)

{

Node node = new Node(num);

node.next = list;

list = node;

}

public void addRear(int num)

{

Node node = new Node(num);

if (list==null)

list = node;

else

{

Node curr = list;

while (curr.next!= null)

{

curr = curr.next;

}

curr.next = node;

}

}

public void printList()

{

Node curr = list;

while (curr != null)

{

System.out.println(curr.data);

curr=curr.next;

}

}

public int sumData()

{

Node curr = list;

int sum = 0;

while (curr != null)

{

sum += curr.data;

curr=curr.next;

}

return sum;

}

public int countNodes()

{

Node curr = list;

int count = 0;

while (curr != null)

{

curr = curr.next;

count++;

}

return count;

}

public int countLargest(int num)

{

Node curr = list;

int count = 0;

while (curr != null)

{

if (curr.data > num)

count++;

curr = curr.next;

}

return count;

}

public int largest()

{

Node curr = list;

int largest = 0;

while (curr != null)

{

if (curr.data > largest)

largest = curr.data;

curr = curr.next;

}

return largest;

}

private class Node

{

public int data;

public Node next;

public Node(int num)

{

data = num;

next = null;

}

}// end Node

}// end LL

/\*\*

\* Class Driver uses class LL.

\*

\* @author (Erik Marrero)

\* @version (Mar 26, 2019)

\*/

public class Driver

{

public static void main(String[] args)

{

LL obj = new LL();

obj.addFront(1);

obj.addFront(3);

obj.addFront(5);

System.out.println("Printing linked list of integers.");

obj.printList();

System.out.println("Sum of data in all nodes = " +

obj.sumData());

LL obj1 = new LL();

obj1.addRear(1);

obj1.addRear(3);

obj1.addRear(5);

System.out.println("Printing second linked list of integers.");

obj1.printList();

System.out.println("Sum of data in all nodes = " +

obj1.sumData());

System.out.println("\nThe number of nodes in the list are:" +

obj1.sumData());

System.out.println("The number of nodes larger than 2 are: " +

obj1.countLargest(2));

System.out.println("The largest number in the list is: " +

obj1.largest());

}

}

