Задание 1.

import java.util.ArrayList;  
/\*  
Задание 1. Удаление дубликатов в односвязном списке  
Напишите метод, который удаляет все дубликаты из односвязного списка.  
Пример:  
Input: 1 -> 2 -> 3 -> 2 -> 4 -> 1  
Output: 1 -> 2 -> 3 -> 4  
 \*/  
class ListNode {  
 Node head;  
 class Node{  
 int value;  
 Node next;  
 }  
  
 public void print() {  
 while (head != null) {  
 System.*out*.print(head.value);  
 head = head.next;  
 if (head != null)  
 System.*out*.print(" -> ");  
 }  
 }  
  
 public void push(int value){  
 Node node = new Node();  
 node.value = value;  
  
 if (head == null){  
 head = node;  
 }else {  
 Node currunt = head;  
 while (currunt.next != null){  
 currunt = currunt.next;  
 }  
 currunt.next = node;  
 }  
 }  
  
 public void removeDuplicates() {  
 if (head == null) return;  
  
 ArrayList<Integer> list = new ArrayList<>();  
  
 Node current = head;  
  
 list.add(current.value);  
  
 while (current.next != null){  
 if(list.contains(current.next.value)){  
 current.next = current.next.next;  
 } else {  
 list.add(current.next.value);  
 current = current.next;  
 }  
 }  
 }  
}  
  
public class Main {  
 public static void main(String[] args) {  
 ListNode list = new ListNode();  
  
 list.push(1);  
 list.push(2);  
 list.push(3);  
 list.push(2);  
 list.push(4);  
 list.push(1);  
  
 list.removeDuplicates();  
 list.print();  
 }  
}

Задача 2

/\*  
Задача 2. Поиск среднего элемента в односвязном списке  
Реализуйте метод, который находит средний элемент односвязного списка за  
один проход по нему.  
Пример:  
Input: 1 -> 2 -> 3 -> 4 -> 5  
Output: 3  
 \*/  
class ListNode {  
 ListNode head;  
 int value;  
 ListNode next;  
  
 public void print() {  
 while (head != null) {  
 System.*out*.print(head.value);  
 head = head.next;  
 if (head != null)  
 System.*out*.print(" -> ");  
 }  
 }  
  
 public void push(int value) {  
 ListNode node = new ListNode();  
 node.value = value;  
  
 if (head == null) {  
 head = node;  
 } else {  
 ListNode currunt = head;  
 while (currunt.next != null) {  
 currunt = currunt.next;  
 }  
 currunt.next = node;  
 }  
 }  
  
 public void findMiddle() {  
 if (head != null) {  
 ListNode slow = head;  
 ListNode fast = head;  
 while (fast != null && fast.next != null){  
 slow = slow.next;  
 fast = fast.next.next;  
 }  
 System.*out*.println("Средний элемент " + slow.value);  
 }  
 }  
}  
  
public class Main {  
 public static void main(String[] args) {  
 ListNode list = new ListNode();  
  
 list.push(1);  
 list.push(2);  
 list.push(3);  
 list.push(4);  
 list.push(5);  
 list.findMiddle();  
 list.print();  
 }  
}

Задание 3

/\*  
Задача 3. Слияние двух отсортированных односвязных списков  
Реализуйте метод, который сливает два отсортированных односвязных списка  
в один отсортированный список.  
Пример:  
Input: 1 -> 3 -> 5 и 2 -> 4 -> 6  
Output: 1 -> 2 -> 3 -> 4 -> 5 -> 6  
 \*/  
  
class ListNode {  
 ListNode head;  
 int value;  
 ListNode next;  
 ListNode(int value) {this.value = value;}  
  
 public void print(ListNode list) {  
 while (list != null) {  
 System.*out*.print(list.value);  
 list = list.next;  
 if (list != null)  
 System.*out*.print(" -> ");  
 }  
 }  
  
 public static ListNode mergeTwoList(ListNode list1, ListNode list2){  
 ListNode dummy = new ListNode(-1);  
 ListNode merge = dummy;  
 while (list1 != null && list2 != null){  
 if (list1.value < list2.value){  
 merge.next = list1;  
 list1 = list1.next;  
 } else {  
 merge.next = list2;  
 list2 = list2.next;  
 }  
 merge = merge.next;  
 }  
  
 merge.next = (list1 != null) ? list1 : list2;  
  
 return dummy.next;  
 }  
}  
  
public class Main {  
 public static void main(String[] args) {  
 ListNode list1 = new ListNode(1);  
 list1.next = new ListNode(3);  
 list1.next.next = new ListNode(5);  
  
 ListNode list2 = new ListNode(2);  
 list2.next = new ListNode(4);  
 list2.next.next = new ListNode(6);  
  
 System.*out*.println("Список 1");  
 list1.print(list1);  
 System.*out*.println();  
 System.*out*.println("Список 2");  
 list2.print(list2);  
 System.*out*.println();  
 System.*out*.println("Список после слияния двух списков");  
  
 ListNode list = ListNode.*mergeTwoList*(list1, list2);  
 list.print(list);  
  
 }  
}