Problem 4

class ListNode{  
 int data;  
 ListNode next;  
 ListNode(int data){  
 this.data=data;  
 }  
}  
public class problem4 {  
 public static ListNode addTwoNumber(ListNode l1,ListNode l2){  
 ListNode dummy=new ListNode(0);  
 ListNode curr=dummy;  
 int carry=0;  
 while(l1!=null||l2!=null||carry!=0){  
 int sum=0;  
 if(l1!=null){  
 sum+=l1.data;  
 l1=l1.next;  
 }  
 if(l2!=null){  
 sum+=l2.data;  
 l2=l2.next;  
 }  
 sum+=carry;  
 carry=sum/10;  
 curr.next=new ListNode(sum%10);  
 curr=curr.next;  
 }  
 return dummy.next;  
 }  
 public static void printList(ListNode head){  
 while(head!=null){  
 System.*out*.print(head.data);  
 if(head.next!=null) {  
 System.*out*.print("->");  
 }  
 head = head.next;  
 }  
 System.*out*.println();  
 }  
 public static void main(String[] args) {  
 ListNode l1=new ListNode(2);  
 l1.next=new ListNode(4);  
 l1.next.next=new ListNode(3);  
  
 ListNode l2=new ListNode(5);  
 l2.next=new ListNode(6);  
 l2.next.next=new ListNode(4);  
 ListNode ans=*addTwoNumber*(l1,l2);  
 *printList*(ans);  
 }  
  
}

A black screen with text

AI-generated content may be incorrect.

Problem 5

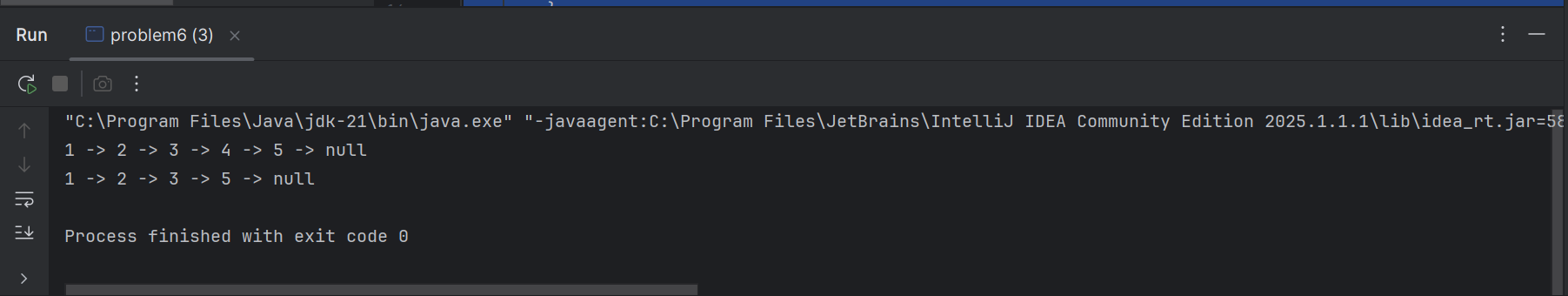
class Node{  
 int data;  
 Node next;  
 Node(int data){  
 this.data=data;  
 }  
}  
public class problem5 {  
 static Node reverse(Node head){  
 Node prev=null;  
 Node curr=head;  
 Node next;  
 while(curr!=null){  
 next=curr.next;  
 curr.next=prev;  
 prev=curr;  
 curr=next;  
 }  
 return prev;  
 }  
 static void reorder(Node head){  
 if(head==null||head.next==null) return;  
  
 Node slow=head,fast=head;  
 while(fast!=null&&fast.next!=null){  
 slow=slow.next;  
 fast=fast.next.next;  
 }  
 Node revHalf=*reverse*(slow);  
 slow.next=null;  
  
 Node first=head,second=revHalf;  
 while(second!=null){  
 Node l1=first.next;  
 Node l2=second.next;  
  
 first.next=second;  
 second.next=l1;  
 first=l1;  
 second=l2;  
 }  
  
 }  
 static void printList(Node head){  
 while(head!=null){  
 System.*out*.print(head.data+" -> ");  
 head=head.next;  
 }  
 System.*out*.println("null");  
 }  
 public static void main(String[] args) {  
 Node head=new Node(1);  
 head.next=new Node(2);  
 head.next.next=new Node(3);  
 head.next.next.next=new Node(4);  
 head.next.next.next.next=new Node(5);  
  
 *printList*(head);  
 *reorder*(head);  
 *printList*(head);  
 }  
}

A screen shot of a computer

AI-generated content may be incorrect.

Problem 6

class Node{  
 int data;  
 Node next;  
 Node(int data){  
 this.data=data;  
 }  
}  
  
public class problem6 {  
 static Node remove(Node head,int idx){  
 int size=0;  
 Node temp=head;  
 while(temp!=null){  
 size++;  
 temp=temp.next;  
 }  
 if(size==idx){  
 head=head.next;  
 return head;  
 }  
 size=size-idx;  
 Node prev=head;  
 for(int i=1;i<size;i++){  
 prev=prev.next;  
 }  
 prev.next=prev.next.next;  
 return head;  
 }  
 static void printlist(Node head){  
 while(head!=null){  
 System.*out*.print(head.data+" -> ");  
 head=head.next;  
 }  
 System.*out*.println("null");  
 }  
 public static void main(String[] args) {  
 Node head=new Node(1);  
 head.next=new Node(2);  
 head.next.next=new Node(3);  
 head.next.next.next=new Node(4);  
 head.next.next.next.next=new Node(5);  
 *printlist*(head);  
 *remove*(head,2);  
 *printlist*(head);  
  
 }  
}



Problem 7

import java.util.HashSet;  
  
public class problem7 {  
 static int longest(String s){  
 int n=s.length();  
 int left=0,right=0,maxLen=Integer.*MIN\_VALUE*;  
 HashSet<Character> set=new HashSet<>();  
 while(right<n){  
 char c=s.charAt(right);  
 if(!set.contains(c)){  
 set.add(c);  
 maxLen=Math.*max*(right-left+1,maxLen);  
 right++;  
 }else{  
 set.remove(s.charAt(left));  
 left++;  
 }  
 }  
 return maxLen;  
 }  
 public static void main(String[] args) {  
 String s="abcdaa";  
 System.*out*.println(*longest*(s));  
 }  
}

A black screen with white text

AI-generated content may be incorrect.

Problem 8

import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.HashMap;  
import java.util.List;  
  
public class problem8 {  
 static List<List<String>> anagram(String[] str){  
 HashMap<String,List<String>> map=new HashMap<>();  
 for(String s:str){  
 char[] charArr=s.toCharArray();  
 Arrays.*sort*(charArr);  
 String sorted=new String(charArr);  
  
 if(!map.containsKey(sorted)){  
 map.put(sorted,new ArrayList<>());  
 }  
  
 map.get(sorted).add(s);  
 }  
 return new ArrayList<>(map.values());  
 }  
 public static void main(String[] args) {  
 String[] str={"eat","tea","tan","ate","nat","bat"};  
 List<List<String>> result=*anagram*(str);  
  
 for(List<String> s:result) System.*out*.println(s);  
 }  
}

A computer screen shot of a computer

AI-generated content may be incorrect.

Problem 9

public class problem9 {  
 public static void main(String[] args) {  
 int[] nums={5,0,2,2,1};  
 int[] count=new int[10];  
 for(int i:nums){  
 count[i]++;  
 }  
  
 int idx=0;  
 for(int i=0;i<10;i++){  
 while(count[i]-->0){  
 nums[idx++]=i;  
 }  
 }  
  
 for(int i:nums){  
 System.*out*.print(i);  
 }  
 System.*out*.println();  
  
 }  
}

A screenshot of a computer

AI-generated content may be incorrect.