1. soru cevabı:

class OgrenciNode {  
 String ad;  
 int not;  
 OgrenciNode sonraki;  
  
  
 public OgrenciNode(String ad, int not) {  
 this.ad = ad;  
 this.not = not;  
 this.sonraki = null;  
 }  
}  
  
  
class OgrenciListesi {  
 OgrenciNode bas;  
  
 public void ogrenciEkle(String ad, int not) {  
 OgrenciNode yeniNode = new OgrenciNode(ad, not);  
 if (bas == null) {  
 bas = yeniNode;  
 } else {  
 OgrenciNode temp = bas;  
 while (temp.sonraki != null) {  
 temp = temp.sonraki;  
 }  
 temp.sonraki = yeniNode;  
 }  
 }  
  
 public void basariliOgrencileriYazdir(int gecmeNotu) {  
 OgrenciNode temp = bas;  
 while (temp != null) {  
 if (temp.not >= gecmeNotu) {  
 System.*out*.println("Adı: " + temp.ad + ", Not: " + temp.not);  
 }  
 temp = temp.sonraki;  
 }  
 }  
}  
  
  
public class Main {  
 public static void main(String[] args) {  
 OgrenciListesi liste = new OgrenciListesi();  
  
  
 liste.ogrenciEkle("Ayşe", 45);  
 liste.ogrenciEkle("Ali", 78);  
 liste.ogrenciEkle("Ahmet", 56);  
 liste.ogrenciEkle("Veli", 65);  
 liste.ogrenciEkle("Kemal", 59);  
  
 System.*out*.println("Başarılı Öğrenciler:");  
 liste.basariliOgrencileriYazdir(50);  
 }  
}

,

2. soru cevabı:

class VucutKitleIndeksi {  
 String ad;  
 double kilo;  
 double boy;  
 VucutKitleIndeksi sonraki;  
  
 VucutKitleIndeksi(String ad, double kilo, double boy) {  
 this.ad = ad;  
 this.kilo = kilo;  
 this.boy = boy;  
 this.sonraki = null;  
 }  
}  
  
class LinkedList {  
 VucutKitleIndeksi bas;  
  
 public void ekle(String ad, double kilo, double boy) {  
 VucutKitleIndeksi yeniVucutKitleIndeksi = new VucutKitleIndeksi(ad, kilo, boy);  
 if (bas == null) {  
 bas = yeniVucutKitleIndeksi;  
 } else {  
 VucutKitleIndeksi temp = bas;  
 while (temp.sonraki != null) {  
 temp = temp.sonraki;  
 }  
 temp.sonraki = yeniVucutKitleIndeksi;  
 }  
 }  
 public void obezleriYazdir() {  
 VucutKitleIndeksi temp = bas;  
 System.*out*.println("Obez Kişiler:");  
 while (temp != null) {  
 double vki = temp.kilo / (temp.boy \* temp.boy);  
 if (vki >= 30) {  
 System.*out*.println("Ad: " + temp.ad + ", VKİ: " + vki);  
 }  
 temp = temp.sonraki;  
 }  
 }  
}  
  
public class Main {  
 public static void main(String[] args) {  
 LinkedList liste = new LinkedList();  
 liste.ekle("Ayşe", 45, 1.70);  
 liste.ekle("Ali", 78, 1.90);  
 liste.ekle("Ahmet", 56, 1.50);  
 liste.ekle("Veli", 65, 1.69);  
 liste.ekle("Kemal", 59, 1.63);  
 liste.obezleriYazdir();  
  
 }  
}

3. soru cevabı:

public class CiftYonluNode {  
 int data;  
 CiftYonluNode onceki;  
 CiftYonluNode sonraki;  
  
 public CiftYonluNode(int data) {  
 this.data = data;  
 this.onceki = null;  
 this.sonraki = null;  
 }  
}  
  
  
public class CiftYonluListe {  
 CiftYonluNode bas;  
  
 public void arayaEkle(int data) {  
 CiftYonluNode yeniNode = new CiftYonluNode(data);  
  
 if (bas == null) {  
 bas = yeniNode;  
 return;  
 }  
  
 CiftYonluNode temp = bas;  
  
 while (temp != null && temp.data < data) {  
 temp = temp.sonraki;  
 }  
  
 if (temp == null) {  
  
 CiftYonluNode son = bas;  
 while (son.sonraki != null) {  
 son = son.sonraki;  
 }  
 son.sonraki = yeniNode;  
 yeniNode.onceki = son;  
 } else if (temp == bas) {  
  
 yeniNode.sonraki = bas;  
 bas.onceki = yeniNode;  
 bas = yeniNode;  
 } else {  
  
 yeniNode.sonraki = temp;  
 yeniNode.onceki = temp.onceki;  
 if (temp.onceki != null) {  
 temp.onceki.sonraki = yeniNode;  
 }  
 temp.onceki = yeniNode;  
 }  
 }  
  
  
 public void yazdir() {  
 CiftYonluNode temp = bas;  
 while (temp != null) {  
 System.*out*.print(temp.data + " ");  
 temp = temp.sonraki;  
 }  
 System.*out*.println();  
 }  
}  
  
  
public class MainSoru3 {  
 public static void main(String[] args) {  
 CiftYonluListe liste = new CiftYonluListe();  
  
 CiftYonluNode n1 = new CiftYonluNode(69);  
 CiftYonluNode n2 = new CiftYonluNode(80);  
 CiftYonluNode n3 = new CiftYonluNode(88);  
 CiftYonluNode n4 = new CiftYonluNode(93);  
  
 n1.sonraki = n2;  
 n2.onceki = n1;  
  
 n2.sonraki = n3;  
 n3.onceki = n2;  
  
 n3.sonraki = n4;  
 n4.onceki = n3;  
  
  
 liste.bas = n1;  
  
 System.*out*.println("83 eklenmeden önce:");  
 liste.yazdir();  
   
 liste.arayaEkle(83);  
  
 System.*out*.println("83 eklendikten sonra:");  
 liste.yazdir();  
 }  
}

4. soru cevabı:

public class MainSoru4 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Dizi uzunluğu: ");  
 int n = scanner.nextInt();  
  
 int[] dizi = new int[n];  
 System.*out*.println("Dizi elemanları: ");  
 for (int i = 0; i < n; i++) {  
 dizi[i] = scanner.nextInt();  
 }  
  
 int toplam = 0;  
 for (int i = 0; i < n; i++) {  
 if (dizi[i] > 0) {  
 toplam += dizi[i];  
 } else {  
 break;  
 }  
 }  
  
 System.*out*.println("Pozitif ardışık sayıların toplamı: " + toplam);  
 }  
}

5. sorunun cevabı:

class Node {  
 int data;  
 Node next;  
  
  
 public Node(int data) {  
 this.data = data;  
 this.next = null;  
 }  
}  
  
  
class TekYonluListe {  
 Node head;  
  
  
 public void ekle(int data) {  
 Node yeniNode = new Node(data);  
 if (head == null) {  
 head = yeniNode;  
 } else {  
 Node temp = head;  
 while (temp.next != null) {  
 temp = temp.next;  
 }  
 temp.next = yeniNode;  
 }  
 }  
  
 public TekYonluListe tersAktar() {  
 TekYonluListe yeniListe = new TekYonluListe();  
 Node temp = head;  
  
  
 java.util.Stack<Integer> stack = new java.util.Stack<>();  
 while (temp != null) {  
 stack.push(temp.data);  
 temp = temp.next;  
 }  
  
  
 while (!stack.isEmpty()) {  
 yeniListe.ekle(stack.pop());  
 }  
  
 return yeniListe;  
 }  
  
  
 public void yazdir() {  
 Node temp = head;  
 while (temp != null) {  
 System.*out*.print(temp.data + " ");  
 temp = temp.next;  
 }  
 System.*out*.println();  
 }  
}  
  
  
public class Main {  
 public static void main(String[] args) {  
 TekYonluListe orijinalListe = new TekYonluListe();  
  
 orijinalListe.ekle(10);  
 orijinalListe.ekle(20);  
 orijinalListe.ekle(30);  
 orijinalListe.ekle(40);  
  
 System.*out*.println("Orijinali:");  
 orijinalListe.yazdir();  
  
 TekYonluListe tersListe = orijinalListe.tersAktar();  
  
 System.*out*.println("Tersi:");  
 tersListe.yazdir();  
 }  
}