package Assignment11;

public class MergeSortList {

static class Node{

int data;

Node next;

Node(int data){

this.data=data;

this.next=null;

}

}

static Node ConvertToNode(int[] arr ) {

Node head= new Node(arr[0]);

Node ptr=head;

for(int i=1;i<arr.length;i++) {

Node temp=new Node(arr[i]);

ptr.next=temp;

ptr=ptr.next;

}

return head;

}

static void display( Node head) {

Node ptr=head;

while(ptr!=null) {

System.***out***.print(ptr.data + "-->");

ptr=ptr.next;

}

}

static Node mergesort(Node head,Node head1,Node head2) {

Node dummy =new Node(0);

Node current=dummy;

Node temp1=head;

Node temp2=head1;

Node temp3=head2;

while(temp1!=null && temp2!=null && temp3!=null ) {

if(temp1.data<=temp2.data && temp1.data<=temp3.data) {

Node temp=new Node(temp1.data);

current.next=temp;

current=current.next;

temp1=temp1.next;

}

else if(temp2.data<temp3.data) {

Node temp=new Node(temp2.data);

current.next=temp;

current=current.next;

temp2=temp2.next;

}

else {

Node temp=new Node(temp3.data);

current.next=temp;

current=current.next;

temp3=temp3.next;

}

}

if(temp1 != null) {

current.next = temp1;

current = current.next;

}

if(temp2 != null) {

current.next = temp2;

current = current.next;

}

if(temp3 != null){

current.next = temp3;

current = current.next;

}

return dummy.next;

}

public static void main(String[] args) {

int[] arr= {1,4,5};

int[] arr1= {1,3,4};

int[] arr2= {2,6};

Node head=*ConvertToNode*(arr);

Node head1=*ConvertToNode*(arr1);

Node head2=*ConvertToNode*(arr2);

Node node1 =*mergesort*(head,head1,head2);

*display*(node1);

}

}

package Assignment11;

public class DeleteNode {

static class Node{

int data;

Node next;

Node(int data){

this.data=data;

this.next=null;

}

}

static Node ConvertToNode(int[] arr ) {

Node head= new Node(arr[0]);

Node ptr=head;

for(int i=1;i<arr.length;i++) {

Node temp=new Node(arr[i]);

ptr.next=temp;

ptr=ptr.next;

}

return head;

}

static void display( Node head) {

Node ptr=head;

while(ptr!=null) {

System.***out***.print(ptr.data + "-->");

ptr=ptr.next;

}

System.***out***.println("null");

}

static Node delete(Node head,int n) {

if (head == null)

return null;

if (head.data == n)

return head.next;

Node ptr=head;

while(ptr.next!=null) {

if(ptr.next.data==n)

ptr.next=ptr.next.next;

ptr=ptr.next;

}

return head;

}

public static void main(String[] args) {

int[] arr= {1,4,5,6};

Node head=*ConvertToNode*(arr);

*display*(head);

*delete*(head,5);

*display*(head);

}

}

package Assignment11;

public class RemoveDuplicates {

static class Node{

int data;

Node next;

Node(int data){

this.data=data;

this.next=next;

}

}

static Node ConvertToNode(int[] arr ) {

Node head= new Node(arr[0]);

Node ptr=head;

for(int i=1;i<arr.length;i++) {

Node temp=new Node(arr[i]);

ptr.next=temp;

ptr=ptr.next;

}

return head;

}

static Node remove(Node head) {

Node ptr=head;

while(ptr.next!=null) {

if(ptr.data == ptr.next.data)

ptr.next=ptr.next.next;

else

ptr=ptr.next;

}

return head;

}

static void display( Node head) {

Node ptr=head;

while(ptr!=null) {

System.***out***.print(ptr.data + "-->");

ptr=ptr.next;

}

}

public static void main(String[] args) {

int[] arr= {1,1,2,3,4,4,5};

Node head=*ConvertToNode*(arr);

//display(head);

Node head1=*remove*(head);

*display*(head1);

}

}