**PJ 4 Report Your Name: Francisco Valadez**

**A. The following is my Java program:**

**// Please copy your Java program into here from your Eclipse window. The code must be colored.**

**// You must not copy Java program from your .java file since the code over there is not colored at all.**

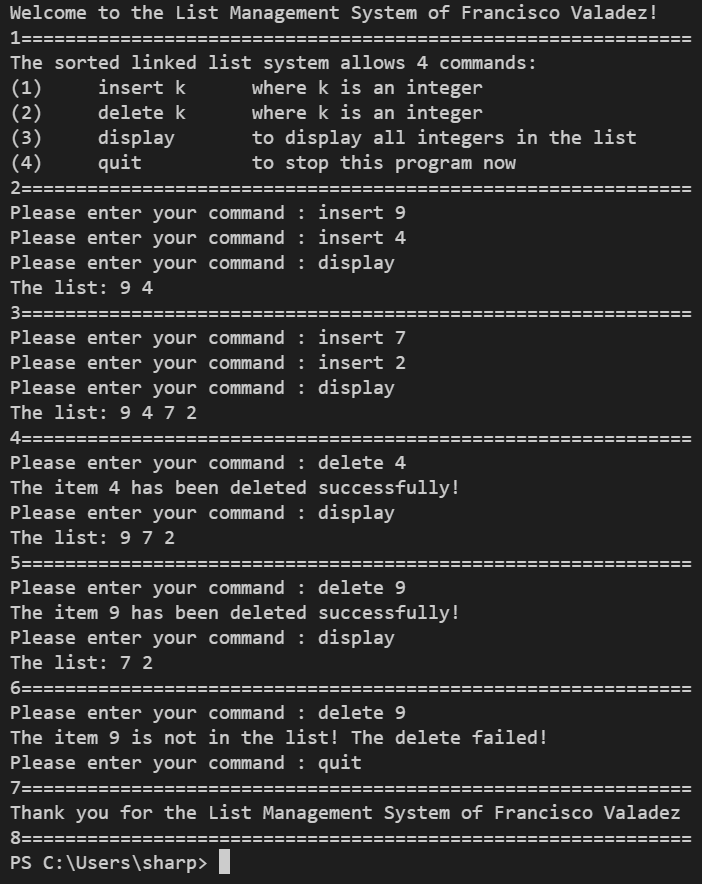
**// You must not show screen prints here.**

// Author: Francisco Valadez  
// Date: 5/21/2021  
// Purpose: CS136 PJ 4 - Manage a Sorted Linked List  
  
import java.util.\*;   
import java.util.Scanner;  
  
public class IntNode // Integer Node with link  
{ // begin class IntNode  
 private int item ; // field 1: item for integer value  
 private IntNode next ; // field 2: next for link to next IntNode  
  
 // list is the static global variable for this program  
 public static IntNode list = new IntNode(-999, null); //The list points to null.   
 // In the beginning, list is empty, so it points to null.   
 // -999 means the integer data is not applicable for this list head pointer.  
  
 public IntNode( int newItem ) // constructor #1 for IntNode class  
 {   
 item = newItem;  
 next = null;  
 }  
  
 public IntNode( int newItem , IntNode nextNode ) // constructor #2 for IntNode class  
 {   
 item = newItem;  
 next = nextNode;  
 }  
   
 public void insert(int i) // insert integer i into the list as a new node   
 { // begin insert( )  
 IntNode n = new IntNode(i, null); // create a new node for integer i  
 if (list.next == null) // list is empty, insert it as the first node  
 {   
 list.next = n; //Points to head of new node  
  
 }   
 else // list is not empty. Insert it into the sorted list at the right position.  
 { // begin else  
  
 IntNode temp = list.next;  
 while(temp.next != null)  
 {  
 temp = temp.next;   
 }  
 temp.next = n;  
  
 } // end else   
 } // end insert( )  
  
 public void delete(int i) // delete integer i from the sorted list   
 { // begin delete( )  
 if (list.next == null) // the list is empty  
 {System.out.println("Empty list! You cannot delete item from an empty list!");return;}  
  
 IntNode prevnode = list ; // previous node is the list node   
 IntNode currnode = list.next ; // get the first node of the list  
 int curritem = currnode.item ; // get the item of the first node, also current node  
  
 while ( currnode != null ) // while current node is not null  
 { // begin while not null   
 curritem = currnode.item ; // get the item while currnode is not null  
 if(curritem == i)  
 {  
 prevnode.next = currnode.next;  
 System.out.println("The item " + i + " has been deleted successfully!");  
 return;  
 }  
 // to delete item successfully & return   
 prevnode = currnode ; // current node becomes previous node  
 currnode = currnode.next ; // go to next node, which can be null   
 } // end while not null   
 // Here means the above loop does not find the item to be deleted.  
 System.out.println("The item " + i + " is not in the list! The delete failed!") ;  
 } // end delete( )  
  
 public void display() // to display all integers in the list  
 {   
 IntNode n = list.next;  
 System.out.print("The list: ");  
 while(n != null)  
 {  
 System.out.print(n.item + " ");  
 n = n.next;  
 }  
 System.out.println();  
 }  
  
 ////////////////////=================== MAIN () ===================////////////////////  
 public static void main(String[] args)   
 { // begin main( )  
 String command;   
 int item = -999, counter = 2; // a virtual item only  
 Scanner scan = new Scanner(System.in);  
 System.out.println("Welcome to the List Management System of Francisco Valadez!");  
 System.out.println(counter - 1 + "=============================================================");  
 System.out.println("The sorted linked list system allows 4 commands:");  
 System.out.println("(1) insert k where k is an integer ");  
 System.out.println("(2) delete k where k is an integer ");   
 System.out.println("(3) display to display all integers in the list ");   
 System.out.println("(4) quit to stop this program now ");   
 System.out.println(counter + "=============================================================");  
 System.out.print("Please enter your command : ");  
 command = scan.next();  
   
 while ( !command.equals("quit") ) // while command is not “quit”  
 { // begin while ( not quit )  
 switch (command)  
 { // begin switch(command)  
 case "display" :   
 list.display();  
 ++counter;  
 System.out.println(counter + "=============================================================");  
 break; // display all integers items  
 case "insert" : item = scan.nextInt(); // get the item to be inserted  
 list.insert( item ); break;  
 case "delete" : item = scan.nextInt(); // get the item to be deleted   
 list.delete( item );   
 break;  
 case "quit":   
 ++counter;  
 System.out.println(counter + "=============================================================");  
 break ; // to quit now   
 default: System.out.println("\""+ command + "\"" + " is not a valid command. ");  
 } // end switch(command)  
   
 System.out.print("Please enter your command : "); // get next command  
 command = scan.next(); // get the command being entered  
 } // end while ( not quit )  
 counter++;  
 System.out.println(counter + "=============================================================");  
 System.out.println("Thank you for the List Management System of Francisco Valadez");  
 counter++;  
 System.out.println(counter + "=============================================================");  
 } // end main( )  
} // end class IntNode ///////////////////////////////////////////////////////////

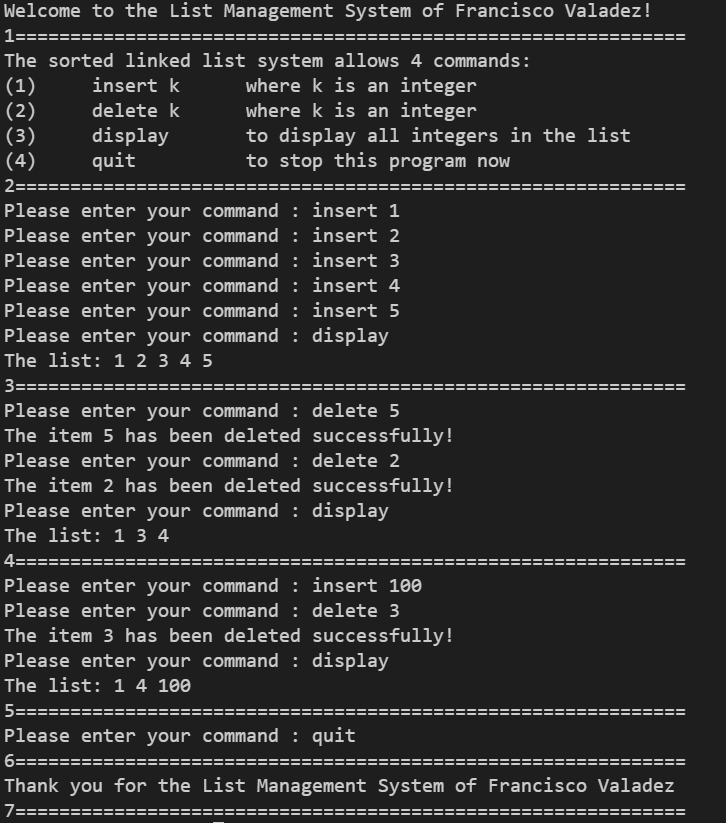
**B. The following is the complete output of my 3 test cases: [You must show 3 test cases.]**

**// Please copy your Eclipse console output into here.**

**Test Case 1:**



**Test Case 2:**



**Test Case 3:**

