```java
// Filename Fread_Program3.2.java
// Written by Justin Fread
// Written on 3/31/19
import javax.swing.JOptionPane;

public class Main {

  public static void main(String[] args) {

    boolean endProgram = false;
    char addItems;
    int selection;
    List nums = new List(" ");

    while(!endProgram) {
      // Add numbers entered by the user to the linked list
      StringBuilder userInputAdd = new StringBuilder
          (JOptionPane.showInputDialog(null,
           "Enter numbers seperated by a space"));
      StringBuilder addNumbers = new StringBuilder();
      int cCount = 0;

      for(int i = 0; i < userInputAdd.length(); i++) {
        addItems = userInputAdd.charAt(i);
        if(!Character.isWhitespace(addItems)) {
          addNumbers.append(addItems);
          cCount++;
        }
        if(Character.isWhitespace(addItems) && cCount > 0) {
          nums.insertItem(addNumbers.toString());
          addNumbers.delete(0, addNumbers.length());
        }
      }
      // Add last number typed by user to the list, delete whitespace,
      // and clear addNumbers
      nums.insertItem(addNumbers.toString());
      nums.deleteItem(" ");
      addNumbers.delete(0, addNumbers.length());
      // Sort list ascending
      selection = JOptionPane.showConfirmDialog(null,
        "Would you like to sort numbers\n" +
        "in ascending order?",
        "Sort Ascending",
        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
      if(selection == JOptionPane.YES_NO_OPTION) {
        nums.sortAscending();
        nums.printList();
      }
      // Sort list descending
      selection = JOptionPane.showConfirmDialog(null,
        "Would you like to sort numbers\n" +
        "in descending order?",
        "Sort Descending",
        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
      if(selection == JOptionPane.YES_NO_OPTION) {
        nums.sortDescending();
        nums.printList();
      }
      // Shuffle list
      selection = JOptionPane.showConfirmDialog(null,
        "Would you like to Shuffle numbers",
        "Shuffle",
        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
      if(selection == JOptionPane.YES_NO_OPTION) {
        nums.shuffle();
      }
      // Delete a number from the list
      selection = JOptionPane.showConfirmDialog(null,
          "Would you like to delete a number\nCurrent List\n" +
               nums.printList2(),
          "Delete Numbers",
          JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
      StringBuilder userInputDel = new StringBuilder
          (JOptionPane.showInputDialog(null,
           "Enter a number to remove from list"));
      if(selection == JOptionPane.YES_NO_OPTION) {
          nums.deleteItem(userInputDel.toString());
          nums.printList();
      }
      userInputDel.delete(0, userInputDel.length());
      // Add more numbers to list or terminate program
      selection = JOptionPane.showConfirmDialog(null,
        "Would you like to enter more numbers?",
        "Enter Additional Numbers",
        JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
          endProgram = (selection != JOptionPane.YES_NO_OPTION);

    }

  }

}
```

===

```java
import javax.swing.JOptionPane;

public class List {
  private Node head;
  public List(String item) {
    head = new Node();
    head.value = item;
    head.link = null;
  }

  public void insertItem(String item) {
    Node n = new Node();
    n.value = item;
    n.link = head;
    head = n;
  }

  public boolean deleteItem(String item) {
    if(head.value.equals(item)) {
       head = head.link;
       return true;
    }
    else {
      Node h = head;
      Node l = head.link;
      while(true) {
        if(l == null || l.value.equals(item)) {
          break;
        }
        else {
          h = l;
          l = l.link;
        }
      }
      if(l != null) {
        h.link = l.link;
        return true;
      }
      else {
        return false;
      }
    }
  }

  public void sortAscending() {
    Node n = head;
    int count = 0;
    while(n != null) {
      n = n.link;
      count++;
    }
    for(int i = 0; i < count; i++) {
      Node current = head;
      Node next = head.link;
      for(int j = 0; j < count - 1; j++) {
        if(Integer.parseInt(current.value) > Integer.parseInt(next.value)) {
          String temp = next.value;
          next.value = current.value;
          current.value = temp;
        }
        current = next;
        next = current.link;
      }
    }
  }

  public void sortDescending() {
    Node n = head;
    int count = 0;
    while(n != null) {
      n = n.link;
      count++;
    }
    for(int i = 0; i < count; i++) {
      Node current = head;
      Node next = head.link;
      for(int j = 0; j < count - 1; j++) {
        if(Integer.parseInt(current.value) < Integer.parseInt(next.value)) {
          String temp = next.value;
          next.value = current.value;
          current.value = temp;
        }
        current = next;
        next = current.link;
      }
    }
  }

  public void shuffle() {
    Node n = head;
    int i = 0;
    StringBuilder s1 = new StringBuilder();
    StringBuilder s2 = new StringBuilder();
    while(n != null) {

      if(i % 2 != 0) {
        s1.append(n.value);
        s1.append("  ");
      }
      else {
        s2.append(n.value);
        s2.append("  ");
      }
      i++;
      n = n.link;
    }
    JOptionPane.showMessageDialog(null, s1.toString() + s2);
  }

  public void printList() {
    Node n = head;
    StringBuilder nums = new StringBuilder();
    while(n != null) {
      nums.append(n.value);
      nums.append("  ");
      n = n.link;
    }
    JOptionPane.showMessageDialog(null, nums.toString());
  }

  public String printList2() {
    Node n = head;
    StringBuilder nums = new StringBuilder();
    while(n != null) {
      nums.append(n.value);
      nums.append("  ");
      n = n.link;
    }
    return nums.toString();
  }

  class Node {
    private String value;
    private Node link;
  }

}
```