01)

Source Code:

List class:

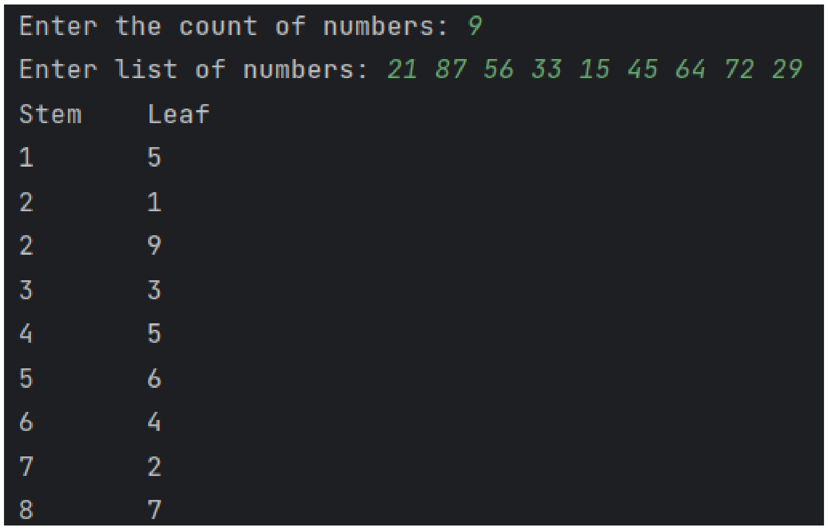
```java
import java.util.Arrays;

public class List<E> {
    private int maxSize;
    private int position;
    private E[] ListEntry;

    List(int size){
        maxSize = size;
        ListEntry = (E[]) new Object[maxSize];
        position = -1;
    }

    @Override
    public String toString() {
        return "List{" +
                "maxSize=" + maxSize +
                ", position=" + position +
                ", ListEntry=" + Arrays.toString(ListEntry) +
                '}';
    }

    //    check if list is empty
    public boolean isListEmpty(){
        return position == -1;
    }

    //     check if list is full
    public boolean isListFull(){
        return position == maxSize -1;
    }

    //    returns the list size
    public int listSize(){
        return ++position;
    }

    //    insert an item to last position of list
    public void insertLast(E value){
        if (isListFull()){
            System.out.println("List is full\n");
        } else {
            ListEntry[++position] = value;
        }
    }

    //    insert an item to given position of list
    public void insertList(E value, int index){
        if (isListFull()){
            System.out.println("List is full\n");
        } else if (index < 0 || index > listSize()){
            System.out.println("Out of list size. Enter a valid index.");
        } else {
            for (int i = listSize(); i > index ; i--) {
                ListEntry[index] = ListEntry[index-1];
                ListEntry[index] = value;
            }
            position++;
        }
    }

//    delete last item of list
    public E deleteList(int index){
        E element;
        if(isListEmpty()){
            System.out.println("List is empty");
        } else if (index < 0 || index >= listSize()) {
            System.out.println("Out of list size. Enter a valid index.");
        } else {
            element = ListEntry[index];
            for (int i = index; i < listSize()-1 ; i++) {
                ListEntry[index] = ListEntry[index+1];
            }
            position--;
            return element;
        }
        return null;
    }

//    retrieve an item from the list
    public E retrieveList(int index){
        E element;
        if (isListEmpty()){
            System.out.println("List is empty");
            return null;
        } else if(index < 0 || index >= listSize()){
            System.out.println("Out of list size. Enter a valid index.");
            return null;
        } else {
            element = ListEntry[index];
            return element;
        }
    }

//    replace an item in a list with a given value
    public void replaceList(int index, E value){
        if(isListEmpty()){
            System.out.println("List is empty");
        } else if (index < 0 || index >= listSize()) {
            System.out.println("Out of list size. Enter a valid index.");
        } else {
            ListEntry[index] = value;
        }
    }

    public void traverselList(){
```

```java
        for (int i = 0; i < position+1; i++) {
            System.out.println(ListEntry[i]);
        }
    }

    public void sortList(){
        Arrays.sort(ListEntry);
    }
}
```

StemLeaf class:

```java
import java.util.Scanner;

public class StemLeaf {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the count of numbers: ");
        String numberCount = scanner.nextLine();

        System.out.print("Enter list of numbers: ");
        String numSeq = scanner.nextLine();

        List<Integer> list = new List<>(Integer.parseInt(numberCount));
        for (String num : numSeq.split(" ")){
            list.insertLast(Integer.parseInt(num));
        }
        list.sortList();

        System.out.println("Stem\tLeaf");
        for (int i = 0; i < list.listSize(); i++) {
                int number = list.retrieveList(i);
                int stem = number / 10;
                int leaf = number % 10;
                System.out.print(stem + "\t\t" + leaf);
                System.out.println();
        }
    }
}
```

Output:



```
Enter the count of numbers: 9
Enter list of numbers: 21 87 56 33 15 45 64 72 29
Stem    Leaf
1       5
2       1
2       9
3       3
4       5
5       6
6       4
7       2
8       7
```