CSCI 323.25

Project 1: Linked List & Middle Node

Name: Gangjun Jiang

Language: Java

Project Due date: 2/8/2023

```
Student: Gangjun Jiang
Submitting Project 1
1) Does your project compile? yes
2) Does your project execute? yes
3) Does your project produce correct results? yes
4) Did you write the code yourself? yes
5) Submission date and time: yes
```

Algorithm Step for this Implementation Step 1: inFile <- open with args[0] outFile <- open with args[1] deBugFile <- open with args[2] Step 2: listHead <- get a new listNode with (“dummy”), as the dummy node for listHead to point to. Step 3: constructLL (listHead, inFile, deBugFile)

Step 4: printList (listHead, outFile) // Print the complete list to outFile

Step 5: middleNode <- findMiddleNode (listHead, deBugFile)

Step 6: if middleNode != null // in case the list is empty outFile <- middleNode’s data // with caption “the word in the middle of list is” Step 7: Close all files

Source Code:

ListNode.java

```java
class ListNode {
    String data;

    ListNode next;

    ListNode(String data) {

        this.data = data;

        this.next = null;
    }


}
```

LList.java

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.LinkedList;
import java.util.Scanner;
/*
@Name :Gangjun Jiang
@Student ID :23795633
@date 2/8/2023
@file name : JiangG_Project1_Main
* */

public class JiangG_Project1_LList {
    private static ListNode listHead = null;
    private static ListNode middleNode = null;
    private static String arrow = "-> ";
    //private static String arrow1 = ",";
    //private static String arrow2= ") ->";

    public static void main(String[] args) throws FileNotFoundException {
        Scanner inFile = new Scanner(new File(args[0]));
        PrintWriter outFile = new PrintWriter(args[1]);
        PrintWriter deBugFile = new PrintWriter(args[2]);

        listHead = new ListNode("Dummy");
        constructLL(listHead,inFile,deBugFile);
        printList(listHead,outFile);
        middleNode = findMiddleNode(listHead,deBugFile);
        if(middleNode != null){
            outFile.print("The word in the middle of list is: "+ middleNode.data );
        }
        inFile.close();
        outFile.close();
        deBugFile.close();

        /*ListNode cur = listHead;
        int n = 0;
        while (cur.next != null) {
            System.out.println(n + ": " + cur.data);
            cur = cur.next;
            n++;
        }*/
    }
    private static void constructLL(ListNode listHead,Scanner inFile, PrintWriter
deBugFile) {
        deBugFile.println("In constructLL method");
        String word = "";

        while (inFile.hasNext()) {
            word = inFile.next();
            ListNode newNode = new ListNode(word);
            listInsert(listHead,newNode,deBugFile);
        }
    }
    private static void listInsert(ListNode listHead, ListNode newNode, PrintWriter
deBugFile){
        deBugFile.println("In listInsert method");

        ListNode Spot = findSpot(listHead, newNode);

        deBugFile.println("Returns from findSpot where Spot.data is: " + Spot.data);

        newNode.next = Spot.next;
        Spot.next = newNode;
    }
    private static ListNode findSpot(ListNode listHead, ListNode newNode){

        ListNode Spot = listHead;
        while (Spot.next != null && Spot.next.data.compareToIgnoreCase(newNode.data) <
0) {
            Spot = Spot.next;
        }
        return Spot;
    }

    private static void printList(ListNode listHead, PrintWriter writeFile){
        ListNode current = listHead;
        writeFile.print("listHead ");

        while (current.next != null) {
            int count = 0;
            while (count < 5 && current.next != null) {
                if(count == 4 ){
                    writeFile.write("(" + current.data + "," + current.next.data + ")"
);
                    current = current.next;
                    count++;
                }else {
                    writeFile.write("(" + current.data + "," + current.next.data + ")"
+ arrow);
                    current = current.next;
                    count++;
                }
            }
            if(current.next == null){
                writeFile.println("NULL");
            }
            writeFile.write("\n");
        }
    }

    private static ListNode findMiddleNode(ListNode listHead,PrintWriter deBugFile){
        deBugFile.println("\n\n\nIn findMiddleNode method");

        ListNode walker1 = listHead.next;
        ListNode walker2 = listHead.next;

        while(walker2 != null && walker2.next != null){
            walker1 = walker1.next;
            walker2 = walker2.next.next;
            deBugFile.println("walker1's data is " + walker1.data);
        }
        return walker1;
    }

}
```

Program Output

outFile from LLMiddleNode_Data1

```
listHead -> (Dummy,a)-> (a,ago)-> (ago,all)-> (all,and)-> (and,and)
-> (and,are)-> (are,brought)-> (brought,conceived)-> (conceived,continent)->
(continent,created)
-> (created,dedicated)-> (dedicated,equal)-> (equal,fathers)-> (fathers,forth)->
(forth,Four)
-> (Four,in)-> (in,liberty)-> (liberty,men)-> (men,nation)-> (nation,new)
-> (new,on)-> (on,our)-> (our,proposition)-> (proposition,score)-> (score,seven)
-> (seven,that)-> (that,the)-> (the,this)-> (this,to)-> (to,years)
-> NULL
the word in the middle of list is: in
```

debugFile from LLMiddleNode_Data1

```
In constructLL method
In listInsert method
Returns from findSpot where Spot.data is: Dummy
In listInsert method
Returns from findSpot where Spot.data is: Four
In listInsert method
Returns from findSpot where Spot.data is: Dummy
In listInsert method
Returns from findSpot where Spot.data is: score
In listInsert method
Returns from findSpot where Spot.data is: seven
In listInsert method
Returns from findSpot where Spot.data is: Dummy
In listInsert method
Returns from findSpot where Spot.data is: Four
In listInsert method
Returns from findSpot where Spot.data is: and
```

```
In listInsert method
Returns from findSpot where Spot.data is: and
In listInsert method
Returns from findSpot where Spot.data is: fathers
In listInsert method
Returns from findSpot where Spot.data is: Four
In listInsert method
Returns from findSpot where Spot.data is: seven
In listInsert method
Returns from findSpot where Spot.data is: brought
In listInsert method
Returns from findSpot where Spot.data is: Dummy
In listInsert method
Returns from findSpot where Spot.data is: Four
In listInsert method
Returns from findSpot where Spot.data is: Four
In listInsert method
Returns from findSpot where Spot.data is: brought
In listInsert method
Returns from findSpot where Spot.data is: Four
In listInsert method
Returns from findSpot where Spot.data is: in
In listInsert method
Returns from findSpot where Spot.data is: ago
In listInsert method
Returns from findSpot where Spot.data is: continent
In listInsert method
Returns from findSpot where Spot.data is: this
In listInsert method
Returns from findSpot where Spot.data is: seven
In listInsert method
Returns from findSpot where Spot.data is: our
In listInsert method
Returns from findSpot where Spot.data is: seven
In listInsert method
Returns from findSpot where Spot.data is: ago
In listInsert method
Returns from findSpot where Spot.data is: liberty
In listInsert method
Returns from findSpot where Spot.data is: and
In listInsert method
Returns from findSpot where Spot.data is: continent
In listInsert method
Returns from findSpot where Spot.data is: dedicated



In findMiddleNode method
walker1's data is ago
walker1's data is all
walker1's data is and
walker1's data is and
```

```
walker1's data is are
walker1's data is brought
walker1's data is conceived
walker1's data is continent
walker1's data is created
walker1's data is dedicated
walker1's data is equal
walker1's data is fathers
walker1's data is forth
walker1's data is Four
walker1's data is in
```

outFile from LLMiddleNode_Data2

```
listHead (Dummy,84)-> (84,a)-> (a,a)-> (a,a)-> (a,about)
(about,aging)-> (aging,American)-> (American,an)-> (an,and)-> (and,and)
(and,and)-> (and,and)-> (and,and)-> (and,apprentice)-> (apprentice,as)
(as,baseball)-> (baseball,battle)-> (battle,been)-> (been,been)-> (been,being)
(being,between)-> (between,boy)-> (boy,by)-> (by,catching)-> (catching,confident)
(confident,Cuba)-> (Cuba,day)-> (day,days)-> (days,each)-> (each,end)
(end,experienced)-> (experienced,far)-> (far,favorite)-> (favorite,fish)-> (fish,fish)
(fish,fish)-> (fish,fisherman)-> (fisherman,fishermen)-> (fishermen,fishing)->
(fishing,Florida)
(Florida,food)-> (food,forbidden)-> (forbidden,form)-> (form,gear)-> (gear,gone)
(gone,Gulf)-> (Gulf,has)-> (has,has)-> (has,hauling)-> (hauling,having)
(having,he)-> (he,He)-> (He,him)-> (him,his)-> (his,his)
(his,his)-> (his,his)-> (his,his)-> (his,in)-> (in,instead)
(instead,into)-> (into,is)-> (is,is)-> (is,its)-> (its,large)
(large,Man)-> (Man,Manolin)-> (Manolin,Manolin)-> (Manolin,marlin)-> (marlin,near)
(near,next)-> (next,night)-> (night,north)-> (north,now)-> (now,of)
(of,of)-> (of,of)-> (of,of)-> (of,Old)-> (Old,on)
(on,opens)-> (opens,out)-> (out,parents)-> (parents,player)-> (player,preparing)
(preparing,sail)-> (sail,salao)-> (salao,Santiago)-> (Santiago,Santiago)->
(Santiago,Santiago)
(Santiago,Santiagos)-> (Santiagos,Sea)-> (Sea,seen)-> (seen,shack)-> (shack,so)
(so,story)-> (story,story)-> (story,Straits)-> (Straits,streak)-> (streak,Stream)
(Stream,successful)-> (successful,talking)-> (talking,tells)-> (tells,tells)->
(tells,that)
(that,that)-> (that,that)-> (that,the)-> (the,the)-> (the,the)
(the,The)-> (The,the)-> (the,The)-> (The,the)-> (the,the)
(the,The)-> (The,to)-> (to,to)-> (to,to)-> (to,told)
(told,unluckiness)-> (unluckiness,unlucky)-> (unlucky,unlucky)-> (unlucky,venture)->
(venture,visits)
(visits,will)-> (will,with)-> (with,with)-> (with,with)-> (with,without)
(without,worst)-> (worst,young)-> NULL

The word in the middle of list is: Manolin
```

debugFile from LLMiddleNode_Data2

```
In constructLL method
In listInsert method
Returns from findSpot where Spot.data is: Dummy
In listInsert method
Returns from findSpot where Spot.data is: Dummy
In listInsert method
Returns from findSpot where Spot.data is: Dummy
In listInsert method
Returns from findSpot where Spot.data is: Dummy
In listInsert method
Returns from findSpot where Spot.data is: Old
In listInsert method
Returns from findSpot where Spot.data is: Old
In listInsert method
Returns from findSpot where Spot.data is: Sea
In listInsert method
Returns from findSpot where Spot.data is: tells
In listInsert method
Returns from findSpot where Spot.data is: Sea
In listInsert method
Returns from findSpot where Spot.data is: Man
In listInsert method
Returns from findSpot where Spot.data is: Dummy
In listInsert method
Returns from findSpot where Spot.data is: and
In listInsert method
Returns from findSpot where Spot.data is: battle
In listInsert method
Returns from findSpot where Spot.data is: a
In listInsert method
Returns from findSpot where Spot.data is: a
In listInsert method
Returns from findSpot where Spot.data is: between
In listInsert method
Returns from findSpot where Spot.data is: experienced
In listInsert method
Returns from findSpot where Spot.data is: Old
In listInsert method
Returns from findSpot where Spot.data is: an
In listInsert method
Returns from findSpot where Spot.data is: Dummy
In listInsert method
Returns from findSpot where Spot.data is: fisherman
In listInsert method
Returns from findSpot where Spot.data is: Man
In listInsert method
Returns from findSpot where Spot.data is: tells
In listInsert method
Returns from findSpot where Spot.data is: Sea
In listInsert method
Returns from findSpot where Spot.data is: Old
In listInsert method
```

```
Returns from findSpot where Spot.data is: The
In listInsert method
Returns from findSpot where Spot.data is: opens
In listInsert method
Returns from findSpot where Spot.data is: fisherman
In listInsert method
Returns from findSpot where Spot.data is: fisherman
In listInsert method
Returns from findSpot where Spot.data is: Dummy
In listInsert method
Returns from findSpot where Spot.data is: between
In listInsert method
Returns from findSpot where Spot.data is: with
In listInsert method
Returns from findSpot where Spot.data is: between
In listInsert method
Returns from findSpot where Spot.data is: 84
In listInsert method
Returns from findSpot where Spot.data is: experienced
In listInsert method
Returns from findSpot where Spot.data is: an
In listInsert method
Returns from findSpot where Spot.data is: marlin
In listInsert method
Returns from findSpot where Spot.data is: battle
In listInsert method
Returns from findSpot where Spot.data is: Sea
In listInsert method
Returns from findSpot where Spot.data is: and
In listInsert method
Returns from findSpot where Spot.data is: opens
In listInsert method
Returns from findSpot where Spot.data is: tells
In listInsert method
Returns from findSpot where Spot.data is: without
In listInsert method
Returns from findSpot where Spot.data is: fisherman
In listInsert method
Returns from findSpot where Spot.data is: now
In listInsert method
Returns from findSpot where Spot.data is: The
In listInsert method
Returns from findSpot where Spot.data is: having
In listInsert method
Returns from findSpot where Spot.data is: He
In listInsert method
Returns from findSpot where Spot.data is: seen
In listInsert method
Returns from findSpot where Spot.data is: unluckiness
In listInsert method
Returns from findSpot where Spot.data is: tells
In listInsert method
```

```
Returns from findSpot where Spot.data is: He
In listInsert method
Returns from findSpot where Spot.data is: worst
In listInsert method
Returns from findSpot where Spot.data is: and
In listInsert method
Returns from findSpot where Spot.data is: Man
In listInsert method
Returns from findSpot where Spot.data is: gone
In listInsert method
Returns from findSpot where Spot.data is: battle
In listInsert method
Returns from findSpot where Spot.data is: fisherman
In listInsert method
Returns from findSpot where Spot.data is: between
In listInsert method
Returns from findSpot where Spot.data is: He
In listInsert method
Returns from findSpot where Spot.data is: opens
In listInsert method
Returns from findSpot where Spot.data is: The
In listInsert method
Returns from findSpot where Spot.data is: parents
In listInsert method
Returns from findSpot where Spot.data is: unlucky
In listInsert method
Returns from findSpot where Spot.data is: He
In listInsert method
Returns from findSpot where Spot.data is: an
In listInsert method
Returns from findSpot where Spot.data is: gone
In listInsert method
Returns from findSpot where Spot.data is: battle
In listInsert method
Returns from findSpot where Spot.data is: to
In listInsert method
Returns from findSpot where Spot.data is: his
In listInsert method
Returns from findSpot where Spot.data is: The
In listInsert method
Returns from findSpot where Spot.data is: experienced
In listInsert method
Returns from findSpot where Spot.data is: unlucky
In listInsert method
Returns from findSpot where Spot.data is: story
In listInsert method
Returns from findSpot where Spot.data is: fisherman
In listInsert method
Returns from findSpot where Spot.data is: that
In listInsert method
Returns from findSpot where Spot.data is: between
In listInsert method
```

```
Returns from findSpot where Spot.data is: unlucky
In listInsert method
Returns from findSpot where Spot.data is: Santiago
In listInsert method
Returns from findSpot where Spot.data is: seen
In listInsert method
Returns from findSpot where Spot.data is: days
In listInsert method
Returns from findSpot where Spot.data is: marlin
In listInsert method
Returns from findSpot where Spot.data is: has
In listInsert method
Returns from findSpot where Spot.data is: him
In listInsert method
Returns from findSpot where Spot.data is: fishermen
In listInsert method
Returns from findSpot where Spot.data is: form
In listInsert method
Returns from findSpot where Spot.data is: parents
In listInsert method
Returns from findSpot where Spot.data is: fishing
In listInsert method
Returns from findSpot where Spot.data is: successful
In listInsert method
Returns from findSpot where Spot.data is: a
In listInsert method
Returns from findSpot where Spot.data is: aging
In listInsert method
Returns from findSpot where Spot.data is: as
In listInsert method
Returns from findSpot where Spot.data is: an
In listInsert method
Returns from findSpot where Spot.data is: him
In listInsert method
Returns from findSpot where Spot.data is: experienced
In listInsert method
Returns from findSpot where Spot.data is: parents
In listInsert method
Returns from findSpot where Spot.data is: salao
In listInsert method
Returns from findSpot where Spot.data is: talking
In listInsert method
Returns from findSpot where Spot.data is: Man
In listInsert method
Returns from findSpot where Spot.data is: tells
In listInsert method
Returns from findSpot where Spot.data is: Old
In listInsert method
Returns from findSpot where Spot.data is: that
In listInsert method
Returns from findSpot where Spot.data is: marlin
In listInsert method
```

```
Returns from findSpot where Spot.data is: catching
In listInsert method
Returns from findSpot where Spot.data is: having
In listInsert method
Returns from findSpot where Spot.data is: visits
In listInsert method
Returns from findSpot where Spot.data is: unlucky
In listInsert method
Returns from findSpot where Spot.data is: experienced
In listInsert method
Returns from findSpot where Spot.data is: opens
In listInsert method
Returns from findSpot where Spot.data is: instead
In listInsert method
Returns from findSpot where Spot.data is: that
In listInsert method
Returns from findSpot where Spot.data is: gone
In listInsert method
Returns from findSpot where Spot.data is: story
In listInsert method
Returns from findSpot where Spot.data is: night
In listInsert method
Returns from findSpot where Spot.data is: now
In listInsert method
Returns from findSpot where Spot.data is: catching
In listInsert method
Returns from findSpot where Spot.data is: his
In listInsert method
Returns from findSpot where Spot.data is: that
In listInsert method
Returns from findSpot where Spot.data is: story
In listInsert method
Returns from findSpot where Spot.data is: now
In listInsert method
Returns from findSpot where Spot.data is: fishing
In listInsert method
Returns from findSpot where Spot.data is: The
In listInsert method
Returns from findSpot where Spot.data is: favorite
In listInsert method
Returns from findSpot where Spot.data is: catching
In listInsert method
Returns from findSpot where Spot.data is: tells
In listInsert method
Returns from findSpot where Spot.data is: him
In listInsert method
Returns from findSpot where Spot.data is: unluckiness
In listInsert method
Returns from findSpot where Spot.data is: Straits
In listInsert method
Returns from findSpot where Spot.data is: into
In listInsert method
```

```
Returns from findSpot where Spot.data is: marlin
In listInsert method
Returns from findSpot where Spot.data is: is
In listInsert method
Returns from findSpot where Spot.data is: each



In findMiddleNode method
walker1's data is a
walker1's data is a
walker1's data is a
walker1's data is about
walker1's data is aging
walker1's data is American
walker1's data is an
walker1's data is and
walker1's data is and
walker1's data is and
walker1's data is and
walker1's data is and
walker1's data is apprentice
walker1's data is as
walker1's data is baseball
walker1's data is battle
walker1's data is been
walker1's data is been
walker1's data is being
walker1's data is between
walker1's data is boy
walker1's data is by
walker1's data is catching
walker1's data is confident
walker1's data is Cuba
walker1's data is day
walker1's data is days
walker1's data is each
walker1's data is end
walker1's data is experienced
walker1's data is far
walker1's data is favorite
walker1's data is fish
walker1's data is fish
walker1's data is fish
walker1's data is fisherman
walker1's data is fishermen
walker1's data is fishing
walker1's data is Florida
walker1's data is food
walker1's data is forbidden
walker1's data is form
walker1's data is gear
```

```
walker1's data is gone
walker1's data is Gulf
walker1's data is has
walker1's data is has
walker1's data is hauling
walker1's data is having
walker1's data is he
walker1's data is He
walker1's data is him
walker1's data is his
walker1's data is his
walker1's data is his
walker1's data is his
walker1's data is his
walker1's data is in
walker1's data is instead
walker1's data is into
walker1's data is is
walker1's data is is
walker1's data is its
walker1's data is large
walker1's data is Man
walker1's data is Manolin
```