CS 2401 – Elementary data structures and algorithms

Fall 2024

Lab 4

Due Date: October 3, Thursday, 11:59PM.

Objective:

The goal of this assignment is to practice file reading, input processing, and implement recursive functions to manipulate and explore properties of strings and linked lists.

The main method will contain test code in Lab4_Lastname.java meant to be used after the completion of each method. You can also add/edit code in main for testing purposes.

You may add helper methods if it helps you find a solution. Results from methods should not make a difference to method calls in main, i.e. expected results should consist of same type and format with or without any helper method(s).

Things to consider

- Are there any base cases or edge cases that weren't included in the lab?
- If so, does your solution address those cases as well?
- Common cases to consider are if input is empty/null, positives/negatives (both?), 0, very large input size, very small input size, ordered/unordered.

Assignment – Timeline:

Milestone 1 – Sep 26 – Sep 29

Task 0 – Store words data

- You will be given a WordsList.txt file where you will find a long list of words separated by commas. The list is stored in a single line.
- Finish the buildArray method by reading input form the text file and create a 1D String array. **NOTE: a try-catch statement is provided**
    - Once we have correctly created a 1D array from the input file, the main method will print random words and the displayed in terminal.
- **Test your code and compare with the Screenshots Below.**

```
Task 0 ########################
Word #0         the
Word #27                last
Word #180               me
Word #253               born
Word #258               speeds
Word #249               far
Word #585               shows
Word #22                dancer
Word #1267              acts
Word #645               flesh
Word #508               pablo
Word #650               worth
Word #2050              starter
Word #1305              filled
```

Task 1 – Reversing a string

- Complete the reverseString recursive method to reverse a given string.

Instructions:

- The **base case** is when the string is empty or has one character (already reversed).
- The **recursive case** should take the first character and append it to the reversed remaining string.
- **Test your code and compare with the Screenshots Below**

```
Task 1 ########################
Original  :  Reversed
bases   sesab
drc     crd
pinch   hcnip
walker  reklaw
aceh    heca
posts   stsop
raw     war
logo    ogol
cent    tnec
fra     arf
glenn   nnelg
role    elor
```

Milestone 2 – Sep 29 – Oct 1

Task 2 – Count number of char, c, in a string

- Finish the countChar recursive method to count the number of times a specific character appears in a string.

Instructions:

- The **base case** is when the string is empty.

- **For each recursive call**, check the first character of the string.
    - If it matches the character c, add 1 to the result of the recursive call with the rest of the string.
- **Test your code and compare with the Screenshots Below**

```
Task 2 ######################
Occurrences of 'e' in mice:     1
Occurrences of 'r' in actors:   1
Occurrences of 'b' in brand:    1
Occurrences of 'e' in leon:     1
Occurrences of 'l' in lies:     1
Occurrences of 's' in houses:   2
Occurrences of 'l' in lap:      1
Occurrences of 'r' in mars:     1
Occurrences of 's' in stream:   1
Occurrences of 'o' in moves:    1
Occurrences of 'a' in pat:      1
Occurrences of 'k' in linked:   1
```

Task 3 – Is this word a palindrome?

- Finish the isPalindrome recursive method to count the number of times a specific character appears in a string.

Instructions:

- The **base case** is when the string is empty or has one character (both are palindromes).
- The **recursive case** compares the first and last characters of the string.
    - If they are the same, the method calls itself with the substring that excludes the first and last characters.
        - Input: “dread” -> recursive call: “rea”
    - Otherwise, the string is not a palindrome
        - Input: “rea” -> return False
- **Test your code and compare with the Screenshots Below**

```
Task 3 ########################
Is 'rotator' a palindrome?   true
Is 'noon' a palindrome?   true
Is 'last' a palindrome?   false
Is 'bare' a palindrome?   false
Is 'time' a palindrome?   false
Is 'est' a palindrome?   false
Is 'refer' a palindrome?   true
Is 'level' a palindrome?   true
Is 'stats' a palindrome?   true
Is 'noon' a palindrome?   true
Is 'spite' a palindrome?   false
Is 'est' a palindrome?   false
```

Milestone 3 – Oct 1 – Oct 3

Task 4 – Compute the n^{th} Fibonacci number

- In mathematics, the Fibonacci sequence is a sequence in which **each number is the sum of the two preceding ones**. The first two numbers in the sequence are 0 and 1.
- Finish the fibonacci recursive method to count the number of times a specific character appears in a string.

Instructions:

- The **base cases** are when n is 0 or n is 1
- The **recursive case** compares the first and last characters of the string.
    - If they are the same, the method calls itself with the substring that excludes the first and last characters.
        - Input: “dread” -> recursive call: “rea”
    - Otherwise, the string is not a palindrome
        - Input: “rea” -> return False
- **Test your code and compare with the Screenshots Below**

```
Task 4 ########################
Fibonacci sequence up to 12:
0 1 1 2 3 5 8 13 21 34 55 89
```

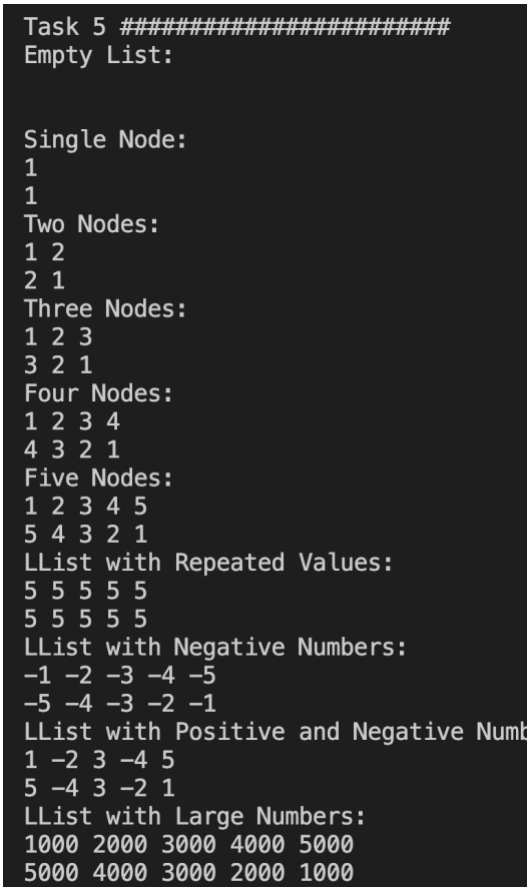
Task 5 –Print the linked list in reverse order

- Complete the printReverse recursive method where given a singly linked list, print the elements of the list in reverse order.
    - Original list: 0 --> 1 --> 1 --> 2 --> 3 --> 5 --> 8
    - Printed list: 8 --> 5 --> 3 --> 2 --> 1 --> 1 --> 0
- The **list should remain unchanged** after the operation.
- The class ListNode.java and the methods to build a singly linked lists of a 1D array will be

provided to create singly linked lists for the problem. Your focus should be in implementing the recursive method, not in creating a singly linked list.

Instructions:

- The **base case** is if the list is empty (head == null), simply return (end of the list).
- The **recursive case** first calls printReverse with the next node to continue moving towards the end of the list.
    - This ensures that the rest of the list is printed before the current node.
- After the recursive call, print the current node's value.
- **Test your code and compare with the Screenshots Below**



Deliverables: You are expected to submit code files in Blackboard (in your lab section). Please use PULSE to develop your solution, then either

(1) Press **Submit** in PULSE, OR

(2) download the .java file (from the **Folder**), rename it as shown below, and submit it in Blackboard. **NOTE: This will also be your class name in java**

Lab4_Lastname.java --- the java file of your program.

Grading Criteria:

- [10 points] The program is indented correctly.
- [10 points] The program is documented properly.
- [10 points] The program uses correct variable types and meaningful variable names.
- [20 points] Program compiles and runs.
- [50 points] The program has correct logic and generates correct output.
  - 10 points per cases 2-5 and 5 points for cases 1 and 6

**** Note that in case of academic dishonesty the grade for this lab will be a 0 and disciplinary actions will be taken ****

- Late submission: [**-10**] points for every 24 hours after the deadline.

If you need any clarification, please ask your TA' for further details.