20.17 Program 8: Recursion

Objectives

* Use recursion
* Use ArrayList class methods
* Use class level variables
* Use method overloading
* Code in an IDE and upload to grading program
* Use documentation and code structure according to class guidelines
* [Link to Piazza Post: Documentation for Java Methods](https://piazza.com/class/kalprp72t55b0?cid=116)

Program 8: Recursion with Lists

**Due Date: Thursday, June 24, 2021 at 11:59 PM PDT**

In this assignment, you will need to implement a Java class and submit through Zybooks: Recursion.java.

The Recursion class will have one class level variables: an ArrayList of Integers that will store a data set on which you will be performing different operations using recursive function. All these functions can also be done using iteration, however, you are restricted to only use recursion for full credit. Your program might be randomly checked and any non-recursive approach will be marked down.

The following non-static public methods will be declared in your class:

* Constructor (1-arg)
  + The constructor will accept in an 1-D array of type int and add every element to the class level ArrayList. There is no recursion required for this method.
* reverseList( ArrayList <Integer> )
  + The method will accept in an ArrayList of type Integer and return a new ArrayList of type Integer that has all the elements in the reverse order. You are only allowed to use recursion for this method.
* reverseList( )
  + The method will use the class level ArrayList and return a new ArrayList of type Integer that has all the elements in the reverse order. You are only allowed to use recursion for this method.
  + Hint: This method is a special case of reverseList( ArrayList <Integer> ) and you are allowed to use that method if it helps.
* toOddList( ArrayList <Integer> )
  + The method will accept in an ArrayList of type Integer as a parameter and return a new ArrayList of type Integer that contains all the odd indexed numbered elements of the class level ArrayList. You are only allowed to use recursion for this method.
* toOddList()
  + The method will use the class level ArrayList and return a new ArrayList of type Integer that contains all the odd indexed numbered elements of the class level ArrayList. You are only allowed to use recursion for this method.
  + Hint: This method is a special case of toOddList( ArrayList <Integer> ) and you are allowed to use that method if it helps.
* toEvenRevList( ArrayList <Integer> )
  + The method will accept in an ArrayList of type Integer as a parameter and return a new ArrayList of type Integer that contains all the even indexed numbered elements of the class level ArrayList in reverse order. You are only allowed to use recursion for this method.
  + You can use the reverseList( ArrayList<Integer> ) method as a helper method.
* toEvenRevList()
  + The method will use the class level ArrayList and return a new ArrayList of type Integer that contains all the even indexed numbered elements of the class level ArrayList in reverse order. You are only allowed to use recursion for this method.
  + Hint: You can use toEvenRevList( ArrayList <Integer> ) or reverseList( ArrayList <Integer> ) as a helper method.
* retPenultimate( ArrayList <Integer> )
  + The method will accept in an ArrayList of type Integer and return an int which is the last element of the ArrayList.If the list is empty/null, it should return -1. As usual, you are only allowed to use recursion for this method and the use of reverseList() is prohibited.
* getList()
  + The method would return the class level ArrayList. The return type is ArrayList <Integer> .

Example

Original: [2, 4, 6, 8, 10]

reverseList(): [10, 8, 6, 4, 2]

toOddList(): [4, 8]

toEvenRevList(): [10, 6, 2]

retPenultimate(): 10

Turning In Procedure

* You are required to submit the Recusion.java in Zybooks which will be autograded.
* Your program might be randomly checked and any non-recursive approach will be marked down.
* The automatic grading program is very specific. If you feel you have the correct solution but are not receiving full credit, please
  + Carefully review the output -- you might need to scroll all the way to the right to find what is wrong with a particular output.
  + Verify you have the correct names for the program itself and all methods.
  + Check your calculations by hand: was there a logic error?
  + Review the requirements: did you miss a step? misinterpret a requirement?
  + If all these check out, contact the T.A. for assistance.