

Homework 5 Chapter 10: Recursion

1. Consider the method `displayRowOfCharacters` that displays any given character the specified number of times on one line. For example, the call

```
displayRowOfCharacters( '*', 5 );
```

produces the line

```
*****
```

Implement this method in Java by using recursion.

Refer to `Homework5Driver.java`

3. Write a method that asks the user for integer input that is between 1 and 10, inclusive. If the input is out of range, the method should recursively ask the user to enter a new input value.

Refer to `Homework5Driver.java`

7. (a) Write a recursive method that writes a given string backward. Consider the last character of the string first.

Refer to `Homework5Driver.java`

- (b) Write a recursive method that writes a given string backward. Consider the first character of the string first.

Refer to `Homework5Driver.java`

Write two versions of a method to display a String backwards. In one version, the print statement should come before the recursive call. In the second version, the print statement should come after the recursive call. Hint: consider using a helper method for the version where the print statement comes after the recursive call.

8. A palindrome is a string that reads the same forward and backward. For example *deed* and *level* are palindromes. Write an algorithm in pseudocode that tests whether a string is a palindrome.

You **must** write a complete functioning method, not just pseudocode. Your method **must** be recursive.

Refer to `Homework5Driver.java`

11. Write a recursive method that counts the number of nodes in a chain of linked nodes.

Refer to `Homework5Driver.java`

15. Write four different recursive methods that each compute the sum of integers in an array of integers. Model your methods after the `displayArray` methods given in Segments 10.15 through 10.18 and described in Question 5.

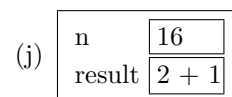
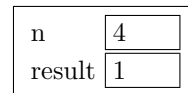
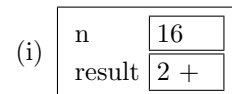
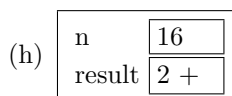
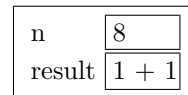
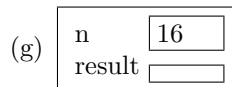
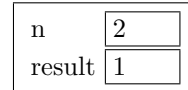
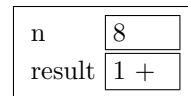
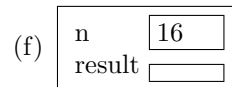
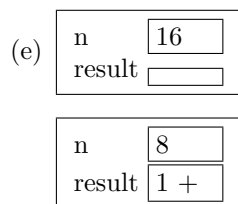
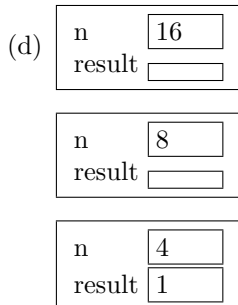
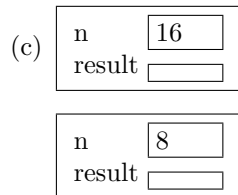
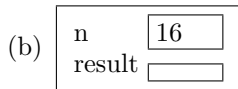
You only need to write three methods.

Refer to `Homework5Driver.java`

17. Trace the call `f(16)` to the following method by showing a stack of activation records:

```
public int f(int n)
{
    int result = 0;
    if (n <= 4)
        result = 1;
    else
        result = f(n / 2) + f(n / 4);
    return result;
} // end f
```

(a) \emptyset



(k) \emptyset

18. Write a recursive algorithm in pseudocode that finds the second smallest object in a list of `Comparable` objects.

Your method can find the second smallest number in an `int[]`. Your method does not have to work for a `ListInterface` object or any `Comparable` object. You can assume the array has at least two elements in it (i.e., the length ≥ 2).

Refer to Homework5Driver.java

Extra Credit

13. Consider the method `contains` of the class `AList`, as given in Segment 5.10 of Chapter 5. Write a private recursive method that `contains` can call, and revise the definition of `contains` accordingly.

Refer to `AList.java`

14. Repeat Exercise 13, but instead use the class `LList` and the method `contains` in Segment 7.12 of Chapter 7.

Refer to `LList.java`

To get extra credit for 13 and 14, you must access the underlying data structure (the array or the linked nodes) directly.

- EC. Write a client-level method `contains` that takes a `ListInterface` and `T` object as parameters and recursively determines if the list contains the object. Do **not** invoke the `contains` method from the `ListInterface` class. The list should not be altered by the method.

In your method header, use `Object` instead of `T`. `T` is funky when working with static methods and I don't want you to get stuck on this, so just use `Object`.

Refer to Homework5Driver.java