

# Lab 11- 20 points

For today's lab you will be doing two exercises pertaining to recursion. The learning objective is to demonstrate your knowledge of recursion by solving two basic problems recursively. These programs do not require a lot of code, but will require some thought and planning beforehand. So, as always, our advice is to work away from the computer. Design your solutions for each specific level, then trace through variables/method calls. Then go to the computer!

You should be able to finish in class, but have until your respective due dates. We recommend finishing these early on but well, and then continuing to work on your projects!

## Part 1 (10 points)

Write a program, RecursiveFinder, that has a **static** method *int largestElement(int[])* .

In that method, using recursion, find the largest element in an array. You may not use/need any loops or any static variables for finding the largest element. If you need to copy arrays for your solution, consider the [helper method in the Arrays class for doing so \(link here\)](#) or you may use a loop to create a new array (but not to determine the answer).

*Hint:* In each step, find the largest element in the subset containing all but the last element. That is, pull out that last element, and recursively determine the largest element in the remaining set. Then compare the maximum of that recursively derived value to the value of the last element you have extracted. Simplify at each step! To help design it, you can/should draw it out on paper with an example array.

In the main method of RecursiveFinder, test this method using at least three arrays showcasing that your method can handle different properties like size and order of elements

## Part 2 (10 points)

Write a program, Reverser, that has a static recursive method `String reverse(String text)` that reverses a string.

For example, `reverse("Hello!")` returns the string `"!olleH"`. Implement a recursive solution by removing the first character, reversing the remaining text, and combining the two (reversed remaining text + first removed character).

In the main method of Reverser, test this method by using at least three Strings of varying complexity and size.

Feel free to model your code after the examples we did in class.

## Submission

Turn in a zip of your “src” folder containing your .java files in a package of your MUID to Canvas by your respective due dates. Ensure you have actually submitted (received confirmation and checked online), since no late labs will be accepted. Grading will be based on conforming to the standards we reviewed in class as well as following the requirements of this lab.