

## Recursion Lab

Please submit a class called **RecursionExercises.java** implementing the following methods:

### **public static int multiply(int a, int b)**

This method takes two numbers  $a$  and  $b$  and recursively multiply them together. Assume that  $a$  and  $b$  are positive integers. The only arithmetic operation allowed in this problem is addition of two integers.

### **public static int findMin(int array[])**

This method takes an array of integers and return the smallest element of the array.

### **public static int reverse(int number)**

This method returns the positive integer obtained when the digits of the parameter is reversed. For example, when called with the parameter 12345, this method would return 54321. (Do this with proper integer arithmetic instead of just simply converting to String, reversing that and then using parseInt to extract the result).

Hint: The number of digits of a number can be obtained by taking the integer part of the logarithm in base 10 of the number + 1 (i.e, num digits =  $\log_{10}(n) + 1$ ).

### **public static countPaths(int n, int m)**

You are standing at the point (n,m) of the grid of positive integers and want to go to origin (0,0) by taking steps either to left or down: that is, from each point (n,m) you are allowed to move either to the point (n-1,m) or the point (n,m-1). Implement the method that based on recursion counts the number of different paths from the point (n,m) to the origin.

### **public String ordString(int number)**

Write a recursive method that given an integer number  $n$  that returns a String based on the following pattern:

```
number = 0, method returns {}  
number = 1, method returns {{}}  
number = 2, method returns {{}}, {{{}}}  
number = 3, method returns {{}}, {{{}}, {{{}}, {{{}}}}
```

In other words, given number  $> 0$ , the ordString method returns all ordStrings from 1 to number. Write a recursive method that accomplish this.