Name: \_Daniel Grimshaw\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Recursion Practice Problems**

1. What is printed by the following statement? System.out.println(m(“GOLD”, “RUSH”)); \_\_\_DH\_\_\_\_\_\_\_\_\_\_

public static String m(String s1, String s2) {

if (s1.length() <= 1 || s2.length() <= 1)

return s2 + s1;

else

return m(s2.substring(1), s1.substring(1));

}

2. What value is returned by the call m1(5)? \_\_243\_\_\_\_\_\_\_\_\_\_\_\_\_

public static int m1(int n) {

if (n == 0)

return 1;

else

return 3 \* m1(n-1);

}

3. What value is returned by the call m1(3)? \_\_16\_\_\_\_\_\_\_\_\_\_\_\_

public int m1(int n) {

if (n == 0)

return 10;

else

return n + m1(n-1);

}

4. What value is returned by the call m2(7)? \_\_\_\_8\_\_\_\_\_\_\_\_

public int m2(int n) {

if (n <= 1)

return 1;

else if (n % 2 == 1)

return n + m2(n-1);

else

return n – m2(n-1);

}

5. What value is returned by the call m3(5,1)? \_\_\_\_8\_\_\_\_\_\_\_\_\_\_\_

public int m3(int a, int b) {

if (a < b)

return a;

else

return b + m3(a-1, b+1);

}

6. What value is returned by the call m4(123, 82)? \_\_41\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public int m4(int a, int b) {

if (a % b == 0)

return b;

else

return m4(b, a % b);

}

7. What does the following method do? \_\_D\_\_\_\_\_\_\_\_\_\_\_\_\_

public int mystery(int a, int b) {

if (a == 1)

return b;

else

return b \* mystery(a-1, b);

}

(A) A + B (B) A \* B (C) AB (D) BA (E) A! (A factorial)

8. Rewrite method, *forwardDisplay*, recursively, without a helper method:

public static void forwardDisplay(int[] primes) {

for (int j = 0; j < primes.length; j++)

System.out.print(primes[j] + " ");

}

public static void forwardDisplay(int [] primes) {  
 if (primes.length > 0) {  
 System.out.print(primes[0] + " ");  
 int [] newPrimes = new int [primes.length-1];  
 for (int i = 1; i < primes.length; i++)  
 newPrimes[i-1] = primes[i];  
   
 forwardDisplay(newPrimes);  
 }  
 }

9. Rewrite method, *backwardDisplay*, recursively, with a helper method:

public static void backwardDisplay(int[] primes) {

for (int j = primes.length - 1; j >= 0; j--)

System.out.print(primes[j] + " ");

}

public static void backwardDisplay(int [] primes) {  
 if (primes.length > 0) {  
 reverse(primes);  
 System.out.print(primes[0] + " ");  
 reverse(primes);  
   
 int [] newPrimes = new int [primes.length-1];  
 for (int i = 0; i < primes.length-1; i++)  
 newPrimes[i] = primes[i];  
   
 backwardDisplay(newPrimes);  
 }  
 }  
  
 private static void reverse(int [] toReverse) {  
 for (int i = 0; i < toReverse.length/2; i++) {  
 int tmp = toReverse[i];  
 toReverse[i] = toReverse[toReverse.length-1-i];  
 toReverse[toReverse.length-1-i] = tmp;  
 }  
 }