Sample Recursion Problems

Tribonacci numbers. The tribonacci numbers are similar to the Fibonacci numbers, except that each term is the sum of the three previous terms in the sequence. The first few terms are 0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81.

Integer partitions. Write a program Partition.java that takes a positive integer N as a command-line argument and prints out all partitions of N. A partition of N is a way to write N as a sum of positive integers. Two sums are considered the same if they only differ in the order of their constituent summands.

```
% java Partition 4      % java Partition 6
4                       6
3 1                     5 1
2 2                     4 2
2 1 1                   4 1 1
1 1 1 1                 3 3
                        3 2 1
                        3 1 1 1
                        2 2 2
                        2 2 1 1
                        2 1 1 1 1
                        1 1 1 1 1 1
```

Write a recursive program GoldenRatio.java that takes an integer input N and computes an approximation to the golden ratio (c.f. en.wikipedia.org/wiki/Golden_ratio) using the following recursive formula:

```
f(N) = 1               if N = 0
     = 1 + 1 / f(N-1)  if N > 0
```

```
public class Partition {

    public static void partition(int n) {
        partition(n, n, "");
    }
    public static void partition(int n, int max, String prefix) {
        if (n == 0) {
            StdOut.println(prefix);
            return;
        }

        for (int i = Math.min(max, n); i >= 1; i--) {
            partition(n-i, i, prefix + " " + i);
        }
    }


    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        partition(N);
    }

}
```

```
class GoldenRatio {
   public static double golden(int n) {
       if (n == 0) return 1;
       return 1.0 + 1.0 / golden(n-1);
   }

   public static void main(String[] args) {
      int N = Integer.parseInt(args[0]);
      System.out.println(golden(N));
   }

}
```