Homework Turnin

Email:	rgalanos@fcps.edu
Section:	6G
Course:	TJHSST APCS 2016-17
Assignment:	02-01
Receipt ID:	668c7cb0b6dc154bf35a85fca59878d2

Turnin Successful!

The following file(s) were received:

Permutations.java (2415 bytes)

```
//name:             date:
import java.util.Scanner;
import java.lang.Math;
import java.io.*;
public class Permutations
{
  // public static int x;

   public static void main(String[] args) throws Exception
   {
      Scanner sc = new Scanner(System.in);
      System.out.print("\nHow many digits? ");
      int n = sc.nextInt();
     // System.out.print("Filename? ");
     // String filename = sc.next();
     // System.setOut(new PrintStream(new FileOutputStream(filename)));
      leftRight("", n);          //when submitting, uncomment all of these
      oddDigits("", n);
      superprime(n);
   }

   public static void leftRight(String s, int n)
   {
      if(s.length()==n)
         System.out.println(s);
      else
      {
         leftRight("L"+s,n);
         leftRight("R"+s,n);
      }
   }


   public static void oddDigits(String s, int n)
   {
      if(s.length()==n)
         System.out.println(s);
      else
      {
         oddDigits("1"+s,n);
         oddDigits("3"+s,n);
         oddDigits("5"+s,n);
         oddDigits("7"+s,n);
         oddDigits("9"+s,n);
      }
   }
   public static void superprime(int n)
   {

    // System.out.println("The superprimes are:" );
```

```java
    // System.out.println(" ");


    //  x = 0;
      recur(2, n); //try leading 2, 3, 5, 7, i.e. all the single-digit primes
      recur(3, n);
      recur(5, n);
      recur(7, n);

     /* if(x==0)
         System.out.println("There are no " + n + " digit superprimes");
      else
         System.out.println("\nThere are " + x + " superprimes." ); */
   }
   private static void recur(int k, int n)
   {
      if(isPrime(k))
      {
         if((int)(Math.log10(k)+1)==n)
         {
            System.out.println(""+k);
           // x++;
         }
         else
         {
            recur(10*k+1,n);
            recur(10*k+3,n);
            recur(10*k+7,n);
            recur(10*k+9,n);
         }
      }

   }
   public static boolean isPrime(int n) {
      if(n < 2)
         return false;
      if(n == 2 || n == 3)
         return true;
      if(n%2 == 0 || n%3 == 0)
         return false;
      int sqrtN = (int)Math.sqrt(n)+1;
      for(int i = 6; i <= sqrtN; i += 6) {
         if(n%(i-1) == 0 || n%(i+1) == 0)
            return false;
      }
      return true;
   }
}
```