Lab 6 (September 27 or September 28)

Instructions: Complete the steps below. Be sure to show your code to one of the lab TAs before you leave, so that you can receive credit for this lab. You must also upload a copy of all your source code (.java) files to the link on Blackboard by 11:59 PM on Thursday, September 28.

1. Implement (define) a Java method `convertMillis()` that converts milliseconds to hours, minutes, and seconds using the following header:

   ```
   public static String convertMillis(long millis)
   ```

   The method returns a `String` in the form hours:minutes:seconds. For example, `convertMillis(5500)` returns the string "0:0:5", while `convertMillis(100000)` returns the string "0:1:40", and `convertMillis(555550000)` returns the string "154:19:10".

   Write a complete Java program that includes the method you just wrote. Your `main()` method should read in a number of milliseconds from the user (as a `long` variable), pass that value to `convertMillis()` as an argument, and print the value that the method returns.

2. Define a Java method `isPrime()` that determines whether an integer is a prime number. Use the following method header:

   ```
   public static boolean isPrime(int num)
   ```

   An integer greater than 1 is a prime number if its only divisors are 1 and itself. For example, `isPrime(11)` returns `true`, and `isPrime(9)` returns `false`.

   Use your `isPrime()` method in a complete Java program that determines and prints the first thousand prime numbers (**NOTE:** this is **NOT** the same as "all of the prime numbers less than or equal to 1000"), displaying every ten prime numbers in a separate row, as follows:

   ```
   2 3 5 7 11 13 17 19 23 29
   31 37 41 43 47 53 59 61 67 71
   73 79 83 89 97 ...
   ```

Grading Guidelines: This lab is graded on a scale of 0-3 points, assigned as follows:

0 points: Student is absent or does not appear to have completed any work for the lab

1 point: Student has written only one program, but it does not compile or run at all due to errors.

2 points: Student has written (or attempted to write) both programs, but only one compiles and runs without error.

3 points: Student has written both programs, and they both compile and run correctly, without any apparent errors.