Lab 01 Solutions for Practice problems

Topic	**Eclipse and Java Setup**
Week	2

Ex. 1.1.19

Develop a better implementation of F(N) that saves computed values in an array.

Solution

```
import java.math.BigInteger;

public class Q_1_1_19 {

	private static int SERIES_LENGTH = 100;

	public static void main(String[] args) {

		BigInteger[] series = new BigInteger[SERIES_LENGTH];

		series[0] = new BigInteger("0");
		series[1] = new BigInteger("1");

		System.out.println("Fibonacci Series: \n0\n1");

		for (int i = 2; i < SERIES_LENGTH; i++) {
			series[i] = series[i - 1].add(series[i - 2]);
			System.out.println(series[i]);
		}
	}
}
```

Ex 1.1.20

Write a recursive static method that computes the value of ln (N !)

Solution

```
public class Q_1_1_20 {

	public static long fact(int n) {
		if (n < 2)
			return 1;
		return n * fact(n - 1);
	}

	public static void main(String[] args) {
		int NUMBER = 10;
		System.out.println("ln(" + NUMBER + "!) = " + Math.log(fact(NUMBER)));
	}
}
```

Ex 1.1. 36

Empirical shuffle check. Run computational experiments to check that our shuffling code on page 32 works as advertised. Write a program ShuffleTest that takes command-line arguments M and N, does N shuffles of an array of size M that is initialized with a[i] = i before each shuffle, and prints an M-by-M table such that row i gives the number of times i wound up in position j for all j. All entries in the array
should be close to N/M.

Solution

```
import java.util.Random;

public class Q_1_1_36 {

	public interface IShuffle {
		public void shuffle(int[] a);
	}

	public static void ShuffleTest(IShuffle shuffle, int m, int n) {
		int[][] s = new int[m][m];
		for (int k = 0; k < n; k++) {
			int[] a = new int[m];
			for (int i = 0; i < m; i++)
				a[i] = i;
			shuffle.shuffle(a);
			for (int i = 0; i < m; i++)
				s[i][a[i]]++;
		}
		for (int i = 0; i < m; i++) {
			for (int j = 0; j < m; j++)
				System.out.printf("%7d", s[i][j]);
			System.out.println();
		}
	}

	public static void main(String[] args) {

		int m = Integer.parseInt("10");
		int n = Integer.parseInt("100000");

		IShuffle shuffle = new IShuffle() {
			public void shuffle(int[] a) {
				for(int i = 0; i < a.length; i++)
				{
					int r = i + new Random().nextInt(a.length-i);
					int temp = a[i];
					a[i] = a[r];
					a[r] = temp;
				}
			}
		};
		ShuffleTest(shuffle, m, n);
	}

}
```