1. Review the Java code below then Re-write the whole program to improve the source code.

```
public class Factorial

{

        public static void main(String[] args)

        { final int NUM = 100;

                for(int i = 0; i < NUM; i++)

                        System.out.println( i + "! is " + factorial(i));

        }


        public static int factorial(int n)

        { int result = 1;

                for(int i = 2; i <= n; i++)

                        result *= i;

                return result;

        }

}
```

```
import java.math.BigInteger;
import java.util.HashMap;
import java.util.Map;

public class Factorial {
    // use memoization to store the value of n! that already calculated
    private static final Map<Integer, BigInteger> memo = new HashMap<>();

    public static void main(String[] args) {
        final int NUM = 100;

        // this code below show the result of 0! to 100!
        // NUM + 1 because the loop i will stop at 99
        for (int i = 0; i < NUM+1; i++) {
            System.out.println(i + "! is " + factorial(i));
        }

        // this show only the result of 100!
        // System.out.println(NUM + "! is " + factorial(NUM));

    }
    // use BigInterger because int cannot store the value of 100!(it will overflow)
    public static BigInteger factorial(int n) {

        // if n = 0 or n = 1, return 1
        if (n == 0 || n == 1) {
            return BigInteger.ONE;
        }

        // if the value of n is already calculated, return the value
        if (memo.containsKey(n)) {
            return memo.get(n);
        }
        // calculate the value of n! by n * (n-1)!
        BigInteger result = BigInteger.valueOf(n).multiply(factorial(n - 1));
        // store the value of n! to the memo
        memo.put(n, result);

        return result;
    }
}
```