Homework Assignments 4
Due date: March 3, 11:55PM EST.

You do not have to write the actual programs for any of the problems, but you may want to experiment with some code to verify your own answers.

Problem 1

Write a **recursive** method that computes the sum of the digits in an integer. Use the following method header:

```java
public static int sumOfDigits ( long n )
```

For example, `sumOfDigits( 234 )` should return 9 (since $2 + 3 + 4 = 9$).

Problem 2

For each of the following recursive methods, rewrite it using iterations instead of recursion.

A)

```java
public int power (int base, int exponent)
{
    if ( exponent == 0 )
        return 1;
    else
        return base * power (base, exponent -1 );
}
```

B)

```java
public int recur ( int n )
{
    if (n < 0 )
        return -1;
    else if ( n < 10 )
        return 1;
    else
        return ( 1 + recur ( n / 10 ) );
}
```
C)    
    public int recur2 ( int n )
    {
        if ( n < 0 )
            return -1;
        else if ( n < 10 )
            return 1;
        else
            return ( n % 10 + recur2 ( n / 10 ) ) ;
    }

Problem 3

Given the following method:

    public int collatz( int num ) {
        if ( num < 2 )
            return 1;
        else
            if ( num % 2 == 0 )
                return collatz ( num / 2 );
            else
                return collatz ( 3 * num + 1 );
    }

How many recursive calls are made by the following initial calls:

A) collatz ( 7 )
B) collatz ( 8 )
C) collatz ( 15 )
D) collatz ( 40 )

THINK ABOUT:
Do you notice anything unusual about this function? Will it terminate for every positive number?

How and what to submit

You can use the template provided at https://docs.google.com/document/d/1PyF6k1QyP9IbsDA42iKNl-mFWx0FjN-0vwZMVG9spzM/edit?usp=sharing to complete your solution.

The completed solutions should be submitted as PDF documents to NYU Classes.