Assignment 5 - Recursion

Nicholas White

DS_Recursion_Assignment.java

sumOfDigits()
recur()
recur2()

Question 4:
1. 16 recursion calls
2. 3 recursion calls
3. 17 recursion calls
4. 8 recursion calls

For any positive integer if it's even divide by two and if it's odd multiply by three and add 1. In my opinion, it looks as though it'll diverge but it converges. Eventually, you will reach a power of 2 going through these steps, and therefore it can terminate to 1 by dividing by 2 until it terminates.

```
public class DS_Recursion_Assignment {

    public DS_Recursion_Assignment() {
    }

    public int sumOfDigits(long n) {
        if (n < 0){
            n = -1 * n;
        }

        if((n/10) == 0){
          return (int)(n);
        }
        else{
            return (int)(((n%10) + sumOfDigits(n/10)));
        }
    }

    public int recur(int n) {
        int counter = 0;
        if (n<0){
            return -1;
        }
        do{
            counter++;
            n = n/10;
        }while (n>0);
```

```
    //counter++;
    return counter;
}



public int recur2(int n) {
    if (n <0){
        n = -1 * n;
    }
    int a = 0;
    int result = 0;
    if (n<0){
        return -1;
    }
    if (n==0){
        return n;
    }
    while (n != 0){
        result = n%10;
        if ((n/10) == 0){
            result = n;
        }
        a = a + result;
        n = n/10;
    }
    return a;
}
```