

Recursion Practice

1) Write a recursive function named **reverse** that prints a positive integer backwards.

reverse(1234) → prints 4321

```
void reverse(int n) {  
    ...  
}
```

2) Given a non-negative integer **num**, repeatedly add all its digits until the result has only one digit. For example:

Given **num = 38**, the process is like: **3 + 8 = 11**, **1 + 1 = 2**. Since **2** has only one digit, return it.

```
int addDigits(int num) {  
    ...  
}
```

3) The Fibonacci numbers is an integer sequence where the next number in the sequence is determined by the sum of the previous two numbers. The sequence starts off as the following:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

where:

$$1 + 1 = 2$$

$$1 + 2 = 3$$

$$2 + 3 = 5$$

$$3 + 5 = 8 \dots \text{etc.}$$

Create a recursive method that calculates the n^{th} number in the sequence.

$$\text{fibonacci}(3) = 2$$

$$\text{fibonacci}(4) = 3$$

$$\text{fibonacci}(5) = 5$$

$$\text{fibonacci}(6) = 8$$

Hint: For the base case, when $n = 1$ or $n = 2$, return 1.

```
int fibonacci(int n) {  
    ...  
}
```