

1. Fill out fibonacci

The Fibonacci sequence is defined as:

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

The first few Fibonacci numbers are:

0, 1, 1, 2, 3, 5, 8, 13, ...

a. What should the two base cases return?

If $n = 0$:

Solution: 0

If $n = 1$:

Solution: 1

b. Complete the recursive return statement.

$$F(n) = \underline{\hspace{2cm}} + \underline{\hspace{2cm}}$$

Solution:

$$F(n) = F(n-1) + F(n-2)$$

c. Fill out the function below.

```
def fibonacci(n):  
    if _____:  
        return _____  
  
    if _____:  
        return _____  
  
    return _____
```

Solution:

```
def fibonacci(n):  
    if n == 0:  
        return 0  
  
    if n == 1:  
        return 1  
  
    return fibonacci(n - 1) + fibonacci(n - 2)
```

2. Count Digits

The function `count_digits(n)` returns the number of digits in a positive integer.

a. What should `count_digits(67)` return?

Solution: 2

b. What should `count_digits(678)` return?

Solution: 3

c. The expression `n // 10` removes the last digit of a positive integer. For example, `678 // 10 = 67` and `7 // 10 = 0`.

If you already know `count_digits(n // 10)`, what do you need to add to get `count_digits(n)`?

`count_digits(n) = _____ + count_digits(n // 10)`

Solution: 1

d. Fill out the function below.

```
def count_digits_recursive(n):  
    if _____:  
        return _____  
    return _____
```

Solution:

```
def count_digits_recursive(n):  
    if n < 10:  
        return 1  
  
    return 1 + count_digits_recursive(n // 10)
```