

1. What does this recursive function do?

Consider the following recursive function:

```
def power_of_two(k):  
    if k == 0:  
        return 1  
  
    return 2 * power_of_two(k - 1)
```

a. Circle the base case in the code.

Solution: Circle if k == 0: return 1

b. What value does power_of_two(0) return? _____

Solution: 1

b. What value does power_of_two(1) return? _____ **Solution:** 2

c. What value does power_of_two(2) return? _____ **Solution:** 4

c. What value does power_of_two(3) return? _____ **Solution:** 8

d. In your own words, what does this function compute? **Solution:** Many possibly solutions, 2 raised to the power of k

2. Fill out product_to_n

The function product_to_n(n) returns the product of all integers from 1 to n, including n.

a. What should product_to_n(1) return?

product_to_n(1) = _____

Solution: 1

b. What should product_to_n(4) return?

product_to_n(4) = 1 · 2 · 3 · 4 = _____

Solution: 24

c. What should `product_to_n(5)` return?

$$\text{product_to_n}(5) = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = \underline{\hspace{2cm}}$$

Solution: 120

d. If you already know `product_to_n(n - 1)`, what do you need to multiply by to get `product_to_n(n)`?

$$\text{product_to_n}(n) = \underline{\hspace{2cm}} \cdot \text{product_to_n}(n - 1)$$

Solution: n

e. Fill out the function for `product_to_n`.

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

Solution:

```
def product_to_n(n):  
    if n == 1:  
        return 1  
  
    return n * product_to_n(n - 1)
```