Recursion

1. (10 points) Consider the following recursive function:

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

   (a) (2 points) What is `foo(5)`?

   (b) (2 points) What is `foo(10)`?

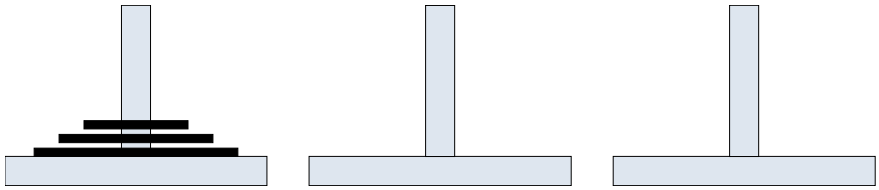
   (c) (3 points) Suppose that `n + foo(n - 1)` is changed to `n * foo(n - 1)`. What is `foo(5)` now?

   (d) (3 points) What happens if `foo(-1)` is called?

2. (10 points) Write a **recursive** function in correct Python syntax that returns the **reverse** of an integer, *n*. For example, if $n = 1234$ (an integer) the function would return 4321 (an integer). Use good coding style. There are many different ways to write this function (e.g., using strings to assist you, etc). Here's the function's header to get you started:

```
def reverse(n):
```

3. (10 points) Consider the Towers of Hanoi problem (as laid out below):



(a) (5 points) If each move takes **1 microsecond**, how long (in **days**) would it take to solve the Towers of Hanoi problem with **37** discs? Provide your answer to a precision of **two** decimal places.

(b) (2 points) What are the **last three** moves of the Towers of Hanoi problem with **343 discs**?

(c) (3 points) What is the Towers of Hanoi's $O()$ performance? Is it good? Explain.