Code Writing

1: Sum of Consecutive Numbers
Write a function `sum_consecutive`. It's argument is a function `f`, which takes in a number and returns either `True` or `False`. `sum_consecutive` should return the sum of every number starting at `0` and going up until the first number for which `f` returns `False`.

```
def sum_consecutive(f):
    i, result = 0, 0
    while f(i):
        result += i
        i += 1
    return result
```

2: The Collatz Sequence
Write a function `collatz_sequence` that takes in a number `n` as its argument, and prints out every number in the Collatz sequence starting at `n` and ending at `1`. The Collatz number after a number `n` is half of `n` if `n` is even, or `3 * n + 1` if `n` is odd.

```
def collatz_sequence(n):
    while n > 1:
        print(n)
        if n % 2 == 0:
            n = n // 2
        else:
            n = 3 * n + 1
    print(1)
```

3: Every Longer Collatz
Write a function `collatz_sequence_length`, which takes in a number `n` and returns the length of the Collatz sequence starting `n`. The Collatz sequence of a number is the sequence printed out by `collatz_sequence`, in the previous problem. After you finish, write a function `every_longer_collatz` that takes no arguments, and prints out every number (starting at 1) whose Collatz sequence is longer than that of every number before it.

```
def collatz_sequence_length(n):
    length = 1
    while n > 1:
        if n % 2 == 0:
            n = n // 2
        else:
            n = 3 * n + 1
        length += 1
    return length
```

This is my favorite question. It's the first Python program I ever wrote in order to solve a real problem. At the time, I was researching the Collatz conjecture. Try copy and pasting it into the terminal!

```
def every_longer_collatz():
    n = 1
    longest_sequence_length = 0
    longest_sequence_number = 1
    while True:
        current_length = collatz_sequence_length(n)
        if current_length > longest_sequence_length:
            longest_sequence_length = current_length
            longest_sequence_number = n
            print(n)
        n += 1
```

4: Amicable Numbers

Write a function `sum_divisors` which takes in a number `n` and returns the sum of all the divisors of `n`. Then write a function `is_amicable`, that returns whether its input `n` is an amicable number. Amicable numbers come in pairs, where each one is the sum of the other's divisors. Lastly, write `every_amicable`, which prints out every amicable number up to `n`, and returns the sum of all those numbers.

```
def sum_divisors(n):
    i, result = 1, 0
    while i < n:
        if n % i == 0:
            result += i
        i += 1
    return result

def is_amicable(n):
    pair = sum_divisors(n)
    return n == sum_divisors(pair)

def every_amicable(n):
    i, result = 0, 0
    while i < n:
        if is_amicable(i):
            result += i
            print(i)
        i += 1
    return result
```