Part 1: From for to while

In lecture, we discussed how every for loop can be translated into a while loop in an algorithmic way. This is your opportunity to practice doing just that.

The file lab11.py contains the following functions:

1.
```
def one ():
    # replace this for-loop with a while loop
    # don't change any more than is necessary
    for j in range(20):
        print ("{0:2} x 2 = {1:<2}".format(j,j*2))
```

2.
```
def two ():
    low = 3
    hi = 50
    total = 0

    # replace this for-loop with a while loop
    # don't change any more than is necessary

    for m in range (hi, low-1,-2):
        total = total + m

    return total
```

3.
```
def three ():
    # replace this for-loop with a while loop
    # don't change any more than is necessary

    for word in "This is a boring sentence".split ():
        print (word,len(word))
```

4.
```
def four ():
    # replace these for-loops with while loops
    # don't change any more than is necessary

    for salute in ["hello","hi","greetings","hey!","<nod>"]:
        for person in ["Jane","Charles","Fitz"]:
            print (salute,person)
```

For each of these functions, **change as little as possible** to convert the for loops into while loops with equivalent behavior. *Do not add break statements, if statements, et cetera: they are not required for this task.*

Part 2: The Syracuse Problem

A popular conjecture in mathematics is the *Collatz Conjecture* (also known as the *Syracuse Problem*), which is concerned with whether every Collatz sequence (explained below) *always* converges to 1.

To generate a Collatz sequence, start with a positive number (say, n), and repeatedly generate new numbers according the following rules:

- If the most recently generated number (say, x) is even, the next number is x divided by 2.
- If the most recently generated number x is odd, the next number is $3x + 1$.

The sequence terminates when the number 1 is generated. For example, the Collatz sequence that starts with 3 is: 3 10 5 16 8 4 2 1

Your task: Add the necessary code (use a while loop!) in the syracuse () function definition in lab11.py to do the following:

- Prompt the user for a positive integer
- Print (on a single line) the Collatz sequence that's generated by the user's input

What and How to Submit

Submit your code through Blackboard. In addition, you should hand in the following items:

- A printout of your code (lab11.py)
- A printout of a shell interaction demonstrating the correctness of all of your answers

  Specifically, you should run each of the functions one (), two (), three (), four (), and syracuse ().