

Problem Wk.3.1.5: Sequential combinations

Part 1: Sum machine

Define a terminating state machine class whose inputs are numbers, which outputs the sum of its inputs so far, and which terminates when the sum is > 100 . The current input should be reflected immediately in the output at that time step.

```
class SumTSM(sm.SM):  
    pass
```

Part 2: Some machine

Make a terminating state machine instance that repeats `SumTSM` four times and then terminates.

```
fourTimes = None
```

Part 3: Counting machine

Define a terminating state machine class that counts from 1 up to specified number and then terminates.

```
>>> m = CountUpTo(3)  
>>> m.run(n=20) # runs machine 20 times, or until termination  
[1, 2, 3]
```

```
class CountUpTo(sm.SM):  
    pass
```

Part 4: Multiple Counting machine

Define a procedure `makeSequenceCounter` that is given a list of numbers and returns a terminating state machine instance that counts from 1 to the first number, then counts from 1 to the next number and so on. It terminates after counting up to the last number.

```
>>> makeSequenceCounter([2,5,3]).run(n=20)  
[1, 2, 1, 2, 3, 4, 5, 1, 2, 3]
```

You can assume that the `CountUpTo` state machine class is already defined.

```
def makeSequenceCounter(nums):  
    pass
```

MIT OpenCourseWare
<http://ocw.mit.edu>

6.01SC Introduction to Electrical Engineering and Computer Science
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.