```
# ShowUpDown.py


"""
As an illustration of the while-loop construction, this
module
checks out the up-down process. Given an integer m, the process generates
the "next
m" according to the rule m/2 if m is even and 3m+1 if m is odd.
Mathematician's conjecture
that the process always "reaches one"  and our
implementations bank on this
fact.
"""

from random import randint as randi

def UpDown(n):

"""
    Returns the number of steps required for
    the up-down process to
reach one when started
    from n.

    Precondition: n is a positive int.

"""

    m=n
    steps=0
    while m>1:
        if m%2 == 0:

# m is even
            m = m/2
        else:
            # m is odd
            m = 3*m+1

   steps+=1;
    return steps

# Test script
if __name__=='__main__':
    """

 Apply UpDown to 10 random integers
    """
    print '\n\n  x     UpDown(x)'

 print '------------------------'
    for k in range(10):
        x = randi(1,10000)
        s
= UpDown(x)
        print '%6d  %6d' % (x,s)
```