HW7

Reading: Chapter 7 and Sections 8.1 and 8.2

Programming:

Submit a single file named hw7.py that contains the solutions to the two problems below. When you are finished, test your solutions using `doctest`. Include the following code at the bottom of your module in order to run the doctest:

```
if __name__=='__main__':
    import doctest
    print( doctest.testfile( 'hw7TEST.py'))
```

1. Develop a class `Volume` that stores the volume for a stereo that has a value between 0 and 11. Usage of the class is listed below the problem descriptions. Throughout the class you must guarantee that:

    - The numeric value of the `Volume` is set to a number between 0 and 11. Any attempt to set the value to greater than 11 will set it to 11 instead, any attempt to set a negative value will instead set it to 0.
    - This applies to the following methods below: `__init__, set, up, down`

    You must write the following methods:

    - `__init__` - constructor. Construct a `Volume` that is set to a given numeric value, or, if no number given, defaults the value to 0. (Subject to 0 <= vol <=11 constraint above.)
    - `__repr__` - converts `Volume` to a `str` for display, see runs below.
    - `set` – sets the volume to the specified argument (Subject to 0 <= vol <=11 constraint above.)
    - `get` – returns the numeric value of the `Volume`
    - `up` – given a numeric amount, increases the `Volume` by that amount. (Subject to 0 <= vol <=11 constraint above.)
    - `down` – given a numeric amount, decreases the `Volume` by that amount. (Subject to 0 <= vol <=11 constraint above.)
    - `__eq__` - implements the operator `==.` Returns `True` if the two `Volumes` have the same value and `False` otherwise.

2. Write a standalone function `partyVolume()` that takes accepts one argument, a string containing the name of a file. The objective of the function is to determine the a Volume object that is the result of many people at a party turning the Volume up and down. More specifically: the first line of the file is a number that indicates the initial value of a `Volume` object. The remaining lines consist of a single character followed by a space followed by a number. The character will be one of `'U"` or `'D'` which stand for "up" and "down" respectively. The function will create a new `Volume` object and then process each line of the file by calling the appropriate method of the `Volume` object which changes the value of the `Volume`. The function then returns the final `Volume` object. Guidelines/hints:

- This is a standalone function, it should NOT be inside (indented within) the class. It should be listed in the module after the `Volume` class and without indentation.

- Note that the first line of the file will be treated differently than all the other lines. Probably the easiest way to do this is to 1) open the file, 2) call the `.readline()` method (no s!) which reads a single line, the initial value of the Volume, then 3) call the `.readlines()` method which reads the rest of the lines. Item 2) will be used to set the initial `Volume` and 3) will be iterated over to represent turning the volume up and down some number of time.

- Make sure you `return` the final `Volume` object.

```
##### Volume #####

# set and get

>>> v = Volume()
>>> v.set(5.3)
>>> v
Volume(5.3)
>>> v.get()
5.3
>>> v.get()==5.3 # return not print
True
>>>

# __init__, __repr__, up, down

>>> v = Volume(4.5) # set Volume with value
>>> v
Volume(4.5)
>>> v.up(1.4)
>>> v
Volume(5.9)
>>> v.up(6) # should max out at 11
>>> v
Volume(11)
>>> v.down(3.5)
>>> v
Volume(7.5)
>>> v.down(10) # minimum must be 0
>>> v
Volume(0)
```

```
# default arguments for __init__

>>> v = Volume() # Volume defaults to 0
>>> v
Volume(0)


# can compare Volumes using ==

>>> # comparisons
>>> v = Volume(5)
>>> v.up(1.1)
>>> v == Volume(6.1)
True
>>> Volume(3.1) == Volume(3.2)
False

# constructor cannot set the Volume greater
# than 11 or less than 0

>>> v = Volume(20)
>>> v
Volume(11)
>>> v = Volume(-1)
>>> v
Volume(0)
>>>


##### partyVolume #####

>>> partyVolume('party1.txt')
Volume(6.35)
>>> partyVolume('party2.txt')
Volume(3.75)
>>> partyVolume('party3.txt')
Volume(0.75)

# make sure return not print

>>> partyVolume('party1.txt')==Volume(6.35) # return not print
True
>>> partyVolume('party2.txt')==Volume(3.75)
True
>>> partyVolume('party3.txt')==Volume(0.75)
True
```