Problem: Duration Tracker Generators

The code below defines a generator that returns the duration of its lifetime when called.

```
from time import sleep
import random
from datetime import datetime
import itertools as it

def producer():
    'produce timestamps'
    starttime = datetime.now()
    while True:
        sleep(random.uniform(0,0.2))
        yield datetime.now()-starttime
```

For example,

```
>>> p = producer()
>>> next(p)
datetime.timedelta(0, 0, 106641)
```

Note that the output of `producer` has a `seconds` attribute. Write a generator that tracks the output of this `producer` and ultimately returns the number of odd numbered seconds that have been iterated over. The usage pattern is the following,

```
>>> t = tracker(p,limit=2)
>>> next(t)
1
>>> list( tracker(p,limit=2))
[1,2]
```

The `limit` keyword argument is the number of odd-numbered seconds to track until completion.

```
>>> list( tracker(p,limit=5))
[0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 5]
```

The last line is interesting because is shows that the producer's `seconds` value output was an even number for the first six iterations. Your `tracker` generator should also receive input that changes the existing limit,

```
>>> t = tracker(p,limit=3)
>>> next(t)
0
>>> next(t)
0
>>> t.send(5)
1
>>> list(t)
[1, 1, 1, 1, 2, 3, 4, 5]
```

Please put your Python code in a Python script file and upload it. Please retain your submitted source files! Remember to use all the best practices we discussed in class. You can use any module in the Python standard library, but third-party modules (e.g., Numpy, Pandas) are restricted to those **explicitly** mentioned in the problem description.

Tips:

- After you have submitted your file, do **not** use the browser back or reload buttons to navigate or open the page in multiple browser tabs, as this may cause your `attempts` to decrease unexpectedly. It may take up to thirty seconds for your code to be processed, so please be **patient**.
- If you find yourself back at the main page without any feedback or change in your `attempts` then it means that your code timed out or crashed in some unexpected way.
- Ensure that your development environment does not presume the existence of certain packages for the autograder. The autograder does not have anything other than the standard library and those third-party libraries **explicitly** named in the problem description.
- Do not leave extraneous statements in your code like test cases, print statements, or anything else besides what is needed to evaluate your submission because the the autograder will spend its limited time executing those lines, which may result in unexpected crashes or timeouts.

浏览... 0304.py

Upload Python source code file

Correct! Back to assignments. functional points = 30 /30 and validation points = 15/15