Thread Exercise

Carried out the python exercise

`code source: https://techmonger.github.io/55/producer-consumer-python/`

```
from threading import Thread
from queue import Queue

q = Queue()
final_results = []

def producer():
    for i in range(100):
        q.put(i)


def consumer():
    while True:
        number = q.get()
        result = (number, number**2)
        final_results.append(result)
        q.task_done()


for i in range(5):
    t = Thread(target=consumer)
    t.daemon = True
    t.start()

producer()

q.join()

print (final_results)
```

**How is the queue data structure used to achieve the purpose of the code? **

The Queue structure is acting as a stack FIFO taking the output of the threads and storing the results into a single structure

What is the purpose of q.put(I)?

This statment takes the current value of the loop itterator (I) and places the value into the queue structure q i.e the value is pushed into the queue if for example the current queue held the value shown below

1 3 4 2

and the value of I was 5 the queue would become

1 3 4 2 5

What is achieved by q.get()? What functionality is provided by q.join()? Extend this producer-consumer code to make the producer-consumer scenario available in a secure way. What technique(s) would be appropriate to apply?

Intersting excerise and shows the issues of predicting sequence and concurency when dealing with indipendent threads.

Regex

Also wrote some code lookign at validation of Postcodes using regex very close to my actual day job this one as its something i have written numerous times over the years as part of data migration projects and FOI requests but never in python 🐍 as I usualy write the regex at the DMBS level using Oralce regex operators so it was nice to see how python does it. And just because it was interesting wrote some code to call a public API to validate postcodes using the public API hosted at https://api.postcodes.io

```
from typing import Dict, List

# Postcodes to test With
POSTCODE = ['M1 1AA', 'M60 1NW', 'CR2 6XH', 'DN55 1PT', 'W1A 1HQ', 'EC1A
1BB']


# Regex Tests
def validate_pcode(pcode: List[str]) -> Dict[str, bool]:
    RESULT = {}
    for x in pcode:
        if any(re.findall(r'\b[A-Z]{1,2}[0-9][A-Z0-9]? [0-9][ABD-HJLNP-UW-
Z]{2}\b', x)):
            RESULT[x] = True
        else:
            RESULT[x] = False
    return RESULT


# Lookup Postcodes using Public API
def api_validate(pcode):
    result = {}
    headers = {'Content-type': 'application/json', 'Accept': 'text/plain'}
    api = requests.post(url='https://api.postcodes.io/postcodes/',
json={"postcodes": POSTCODE}, headers=headers)

    for x in api.json()["result"]:
        if x['result'] is not None:
            result[x["result"]["postcode"]] = True
    return result


print(validate_pcode(POSTCODE))
print(api_validate(POSTCODE))
```