Question A. FindBusinessBasedOnCity(cityToSearch, saveLocation1, collection) –

1. Reflection

This function searches the ‘collection’ given to find all the business present in the city provided in ‘cityToSearch’ and save it to ‘saveLocation1’.

```
In [1]: from unqlite import UnQLite
        db = UnQLite('sample.db')
        data = db.collection('data')
```

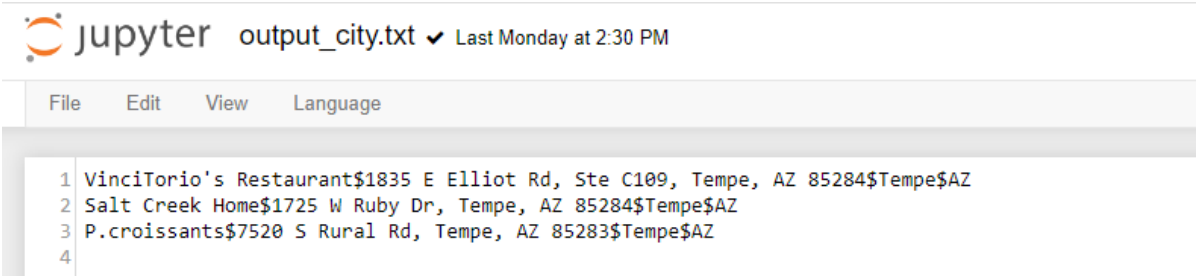
First, import library “unqlite” and import the file “sample.db” which is exported from mongoDB.

```
In [2]: # Graded Cell, PartID: o1flK
        def FindBusinessBasedOnCity(cityToSearch, saveLocation1, collection):
            d = collection.filter(lambda obj: obj['city'].decode() == cityToSearch)
            f = open(saveLocation1, 'w')
            for bus in d:
                txt = bus['name'].decode()+"$"+bus['full_address'].decode()+"$"+bus['city'].decode()+"$"+bus['state'].decode()+"\n"
                f.write(txt)
            f.close()
```

First line shows filtering city that I’d like to search. I used ‘.decode’ to get rid of ‘b’ in front of city name in DB data.

Next, the file name saveLocation1 or 'output_city.txt' is exported to local drive. As you may see on code, each line of the saved file will contain: Name$FullAddress$City$State.

2. Output



Three items have been filtered and saved to text file ‘output_city.txt’ correctly.

3. Result

```
try:
    FindBusinessBasedOnCity('Tempe', 'output_city.txt', data)
except NameError as e:
    print ('The FindBusinessBasedOnCity function is not defined! You must run the cell containing the function before running thi
except TypeError as e:
    print ("The FindBusinessBasedOnCity function is supposed to accept three arguments. Yours does not!")

try:
    opf = open('output_city.txt', 'r')
except FileNotFoundError as e:
    print ("The FindBusinessBasedOnCity function does not write data to the correct location.")

lines = opf.readlines()
if len(lines) != 3:
    print ("The FindBusinessBasedOnCity function does not find the correct number of results, should be 3.")

lines = [line.strip() for line in lines]
if sorted(lines) == sorted(true_results):
    print ("Correct! You FindBusinessByCity function passes these test cases. This does not cover all possible test edge cases, h
```

```
Correct! You FindBusinessByCity function passes these test cases. This does not cover all possible test edge cases, however, so
make sure that your function covers them before submitting!
```

All passed correctly on local machine.

4. Lessons learned

It has been hard time to figure out that decoding the city in mongDB file is necessary in order to search or filter string files. I learned how to search or filter mongoDB file by using python.

Question B. FindBusinessBasedOnLocation(categoriesToSearch, myLocation, maxDistance, saveLocation2, collection) – This function searches the 'collection' given to find the name of all the businesses present in the 'maxDistance' from the given 'myLocation' and save them to 'saveLocation2'

1. Reflection

```
def FindBusinessBasedOnLocation(categoriesToSearch, myLocation, maxDistance, saveLocation2, collection):
    import math
```

First, import math library inside the location function. This is crucial, otherwise, error comes up on autograder. Iterate all of information to find the name.

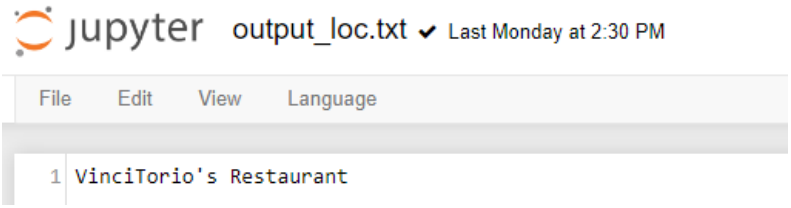
```
    def DistanceFunction(lat2, lon2, lat1, lon1):
        #lat2 = myLocation[0]
        #lon2 = myLocation[1]
        #lat1 = d['latitude']
        #lon1 = d['longitude']
        R = 3959
        o1 = math.radians(lat1)
        o2 = math.radians(lat2)
        do = math.radians(lat2-lat1)
        dl = math.radians(lon2-lon1)
        a = math.sin(do/2) * math.sin(do/2) + math.cos(o1) * math.cos(o2) * math.sin(dl/2) * math.sin(dl/2)
        c = 2 * math.atan2(math.sqrt(a), math.sqrt(1-a))
        distance = R * c
        return distance
```

The above code is for calculating distance between 'my location' and store's location. Since autograder gives me error message when I add the function outside of location function, I just put the distance function inside the location function.

```
    for d in collection.all():
        if DistanceFunction(myLocation[0], myLocation[1], d['latitude'], d['longitude']) < maxDistance:
            list = []
            for de in d['categories']:
                list.append(de.decode())
            if categoriesToSearch[0] in list:
                result = d['name'].decode()
                f = open(saveLocation2, 'w')
                f.write(result)
                f.close()
```

Once calculated distance, if the distance is less than max distance and the category is in the list of DB's categories, the result is saved and exported to saveLocation2 'output_loc.txt'. Decoding the list of categories also is required.

2. Output



Only one item has been filtered and saved to text file ‘output_loc.txt’ correctly.

3. Result

```
try:
    FindBusinessBasedOnLocation(['Buffets'], [33.3482589, -111.9088346], 10, 'output_loc.txt', data)
except NameError as e:
    print ('The FindBusinessBasedOnLocation function is not defined! You must run the cell containing the function before running
except TypeError as e:
    print ("The FindBusinessBasedOnLocation function is supposed to accept five arguments. Yours does not!")

try:
    opf = open('output_loc.txt','r')
except FileNotFoundError as e:
    print ("The FindBusinessBasedOnLocation function does not write data to the correct location.")

lines = opf.readlines()
if len(lines) != 1:
    print ("The FindBusinessBasedOnLocation function does not find the correct number of results, should be only 1.")

if lines[0].strip() == true_results[0]:
    print ("Correct! Your FindBusinessBasedOnLocation function passes these test cases. This does not cover all possible edge cas
```

```
Correct! Your FindBusinessBasedOnLocation function passes these test cases. This does not cover all possible edge cases, so mak
e sure your function does before submitting.
```

All passed correctly on local machine.

4. Lessons learned

I was in trouble of grading using autograder, even though all passed correctly on local machine. This is because I used another function to calculate distance and added ‘import math’ into first line, not inside the location function. Most work I spent was about the errors on autograder. I figured out how to search and filter data in MongoDB files using Python.