CS 251 Inlab9 - Web Scraping, DB and Networking

Instructions :

- Do this lab only in cs251 docker container. All the dependencies needed for this lab are already installed.
- Follow submission and output format strictly otherwise 0 marks will be awarded.
- Untar the resources (**inlab9_resources.tar.gz**) using

  ```
  tar -xvf inlab9_resources.tar.gz
  ```
- Resources contain tester scripts and a csv file for Q2.
- Any changes made to PS will be marked in this color.
- Command to be run on the terminal are marked in this **color** and the output to be printed on terminal is marked in this color.

Q1. Web Scraping [20 points]

(python, bs4, requests, html.parser)

Write a python program **q1.py** which when passed a course code (of IITB course) using **command line arguments** outputs **a semicolon separated list of University:Course_Code** where Course_Code is the mapped course to IITB course code given as input and University is the University which runs that mapped course. See the following sample runs for input and output format.

Strip the newlines, spaces and tabs from the mapped course
Use html5lib parser instead of html.parser

```
$ python3 q1.py "CS228"
KAIST:CS402;NUS:CS3234
```

The course code will be given in double quotes. The university name is to be scraped from the header of the table. The list is sorted based on university name in ascending order (break ties by sorting in ascending order of mapped course code).

```
$ python3 q1.py "CS302"
NUS:CS4215;SFU:CMPT379
```

If the right column has multiple courses, each of individual courses can be given as input and its mapping should appear in the output.

```
$ python3 q1.py "CS416"
NTU:CE4024
```

Thumb rule : Only the **CSXXX** format of input is valid where X is a digit (0 to 9), and if the input is contained in

```
$ python3 q1.py "CS251"
NOT FOUND
```

```
$ python3 q1.py "CS337"
NUS:CS3243 (AI), CS3244
(ML);SFU:CMPT310+CMPT726
```

A file **q1_tester.sh** is included in the resources. To test the above test cases, copy **q1.py** into resources and run `**bash q1_tester.sh q1.py**`.

Q2. Relational Databases [20 points]

(python, sqlite3)

References : https://pynative.com/python-sqlite/
https://www.javatpoint.com/sqlite-tutorial

The file **zipcodes.csv** in the resources contains some US zip codes along with latitude, longitude, city, state and county corresponding to that zipcode.
Your task is to create a python program **q2.py** which when run like

```
$ python3 q2.py <state>
```

does the following :

- Creates a database **zipcodesDB** (to be stored in file **zipcodesDB.db**)
- Creates a table **zipcodesInfo** with the following schema

city	TEXT
state	TEXT
county	TEXT

Set the column **zip_code** as the **PRIMARY KEY**

- Reads the file **zipcodes.csv** (assume that it will be present in the same directory as **q2.py** which is also same as pwd of the terminal from which q2.py will be run) and adds the data into the table **zipcodesInfo**. The **zipcodes.csv** which will be used for evaluation will be similar except for some minor changes in latitudes, longitudes and zip codes (they won't be NULL). So you can assume that the structure of **zipcodes.csv** will not change which includes the order of columns and presence of the same header line in the first line.
- Prints the sorted comma separated list of zip codes which correspond to the northernmost city of the state **<state>** by querying the table. Print **"NOT FOUND"** in case the state doesn't exist in the table. In any other case it should output the zip codes.

Sample test cases :

This test case has been updated

```
$ python3 q2.py AL
35739
```

This is a newly added testcase

```
$ python3 q2.py PR
00603,00604,00605
```

During evaluation, along with the output, whether generated **zipcodesDB.db** contains the table **zipcodesInfo** of the given schema and with the data from **zipcodes.csv** file or not would also be checked.

To check the output of above test cases you can also run `bash q2_tester.sh q2.py` after copying **q2.py** into the same directory as **q2_tester.py**.

Q3. Networking [5 points]

(python, os, ping)

Reference : https://www.geeksforgeeks.org/ping-command-in-linux-with-examples/

Do this question on your native machine connected to the IITB network.

Write a python program **q3.py** which takes in an ip address as command line argument and tells whether the machine is connected to the network and responding to ping or not. Output **YES** if the ping is successful or **NO** if the ping fails.
Use the **os** module of python to run the ping command.
You need to suppress the output of the ping command.
Only **YES** or **NO** should be printed.
Also we don't want ping to run indefinitely, limit the number of packets sent to 5 (and received) using some flag.

Sample test cases :

The above IP corresponds to www.cse.iitd.ac.in

Try to ping these IP addresses from the terminal first.
For more test cases you can ping on IP addresses in the range '10.130.154.1' to '10.130.154.132', most of them would be **YES** except few.
Run the tester script as `**bash q3_tester.sh q3.py**` after copying them in the same directory.

Submission Instructions

The following is what your submission directory must look like. Please follow the submission format strictly. 0 marks will be awarded if the submission format is not correct.

The submission directory must be named **<roll_no>_inlab9**

Compress it into a tarball using the following command

tar -cvzf <roll_no>_inlab9.tar.gz <roll_no>_inlab9

```
<roll_no>_inlab9
├── q1.py
├── q2.py
├── q3.py
.
```

CS 251 Inlab9