Setting up your programming environment

You need to implement the parts of the working scraper system Here are the libraries that you will need:

- For steps 1 and 2: sending GET request and receiving data – python **requests** library or **urllib3** library
- For processing structured data that you receive back: **json** library
- For processing unstructured data: **beautifulsoup4** library which is to be called as **bs4**
- For storing data **sqlite3** library
- For random sampling **random** library

Install each of the libraries as "**pip3 install <libraryname>**" if it's not on your desktop.

TO BE UPLOADED IN MOODLE

- <rollno>_Assgn_6_3.py and <rollno>_Assgn_6_3.txt in home, by friday night.

Problem 3: Using multiple processes to speed up

Now we will convert the above code to be run by multiple processes for speed up.

1. Write a handler function that will do the following (reuse the code from previous example)
    a. Collect the main page of Summer Olympics Wikipedia for this task, the page is here: https://en.wikipedia.org/wiki/Summer_Olympic_Games . Note that you might need to use headers for fetching this page.
    b. Now create a database Create a SQLite database named 'OlympicsData.db' and a table named **'SummerOlympics'** with the following columns:
        i. Name (e.g. "2012 Summer Olympics", in title of wikipedia pages)
        ii. WikipediaURL
        iii. Year (the year when its conducted)
        iv. HostCity (the city where its hosted)
        v. ParticipatingNations (List of the participating nations)
        vi. Athletes (number of athletes)
        vii. Sports (list of sports)
        viii. Rank_1_nation
        ix. Rank_2_nation
        x. Rank_3_nation
        xi. DONE_OR_NOT_DONE (a 1 or 0 variable signifying whether fetched or not respectively)

    c. Parse the html from step 1 and extract the individual summer olympics wiki page urls for **TEN** olympics from the last 50 years, i.e., from 1968 to 2020.
    d. insert the WikipediaURL for each row and set DONE_OR_NOT_DONE as 0 for all.

2. Now the handler code will spawn three processes using os.system call. Example of this call

   ```
   import os
   os.system(“python3 scraper.py&”)
   ```

   This will run “**python3 scraper.py**” in a separate process.

3. This is what **scraper.py** will do
    a. It will check the database for rows where DONE_OR_NOT_DONE flag is 0.
    b. It will pick a row where DONE_OR_NOT_DONE is 0 (if no such row, **scraper.py** will exit).
    c. For the row chosen, **scraper.py** will first set the DONE_OR_NOT_DONE to 1.
    d. Then it will fetch the wikipedia page using URL in the WikipediaURL column
    e. Next using beautifulSoup it will parse the page and populate the columns mentioned in step 1.b. corresponding row in the database
4. Write a **checker.py** code that can check the database and
    a. Report if all the database rows are populated, i.e., there is no DONE_OR_NOT_DONE which is set to 0 and no process is working (figure out how do you check that?)
    b. If all database rows are populated, then print answers to the following:
        i. What are the years you chose?
        ii. Which country was within top 3 for the maximum time in your database?
        iii. What is the average number of athletes?

5. Finally write a small text document documenting what the percentage speed up in time you actually get by running multiple processes. What is your experiment set up and results?