Task 1

Medium.com is an online publishing platform. In Medium, a topic is a broad category that can have multiple posts tagged to it. A section to search Medium posts can be found here.

Your task is to scrape important meta information pertaining to posts belonging to **5** different topics. The output we expect is a JSON file `medium_posts_meta_data.json` that contains a list of objects; each object will have the following keys:

1. topic_name (text)
2. topic_best_all_time_url (text)
3. post_title (text)
4. post_link (text)
5. author_name (text)
6. post_date (datetime)
7. read_time (integer)
8. author_follower_count (integer)
9. post_clap_count (integer)
10. post_comment_count (integer)
11. post_tags (list)

In doing the above, note the following:

1. Choose any 5 topics that interest you
2. Make sure you are only scraping posts which are categorized as the 'Best of All Time'. For example, the best posts of all time pertaining to the topic 'Scraping' can be found in this URL

Also, note the following constraints:

1. You may **not** use Selenium or any other browser automation tool to scrape the data
2. You should choose **5** different topics
3. Each topic that you choose must have at least **9 posts** that are categorized as 'Best of All Time'

For bonus points:

1. Figure out pagination & get more than 9 posts per tag (upto a maximum of 27).
2. Send us an SQLite DB file called `medium_posts_meta_data.db` with the same data that's in the JSON file.

Task 2

Makaan.com is a real estate market website. One of the sections on the website provides a locality-wise price summary. The entry point to this section can be found here.

Your task is to scrape locality-wise price data for the following cities: Chennai, Mumbai, Pune, Puri and Bangalore. The output we expect is a CSV file called `makaan_locality_prices.csv` with the following columns:

1. city_name (text)
2. city_url (text)
3. locality_name (text)

4. price_range_per_sq_ft (text)
5. avg_price_per_sq_ft (float)
6. price_rise (integer)

In doing the above, note the following:

1. The locality-wise price list for a city can be accessed by clicking the city name in the URL above
2. If the locality-wise price list page contains multiple pages, you are required to parse data from all pages

Also, note the following constraints:

1. You may **not** use Selenium or any other browser automation tool to scrape the data
2. You cannot use pandas for parsing.

For bonus points:

1. Store the data in the `price_range_per_sq_ft` column in a smarter way that might make analysis easier
2. Add another column in the CSV called `page_no` indicating which page the locality was scraped from
3. Send us an SQLite DB file called `makaan_locality_prices.db` with the same data that's in the CSV
4. Generate a CSV file called `makaan_localities` with the following columns: city_name, city_url, total_pages, total_localities