Instructions:

1. All 3 questions should be attempted.
2. For question 2 and 3, the solution should be in a file named ***answer_.py.***
3. For questions 1, save the created series/dataframe along with executed notebook (or result screenshots.
4. Output can be submitted by sending the file(s) to aparna.p@scrapehero.com.

Question 1:

Part 01. Create a Pandas series with 100 random dates as it falls between 01-01-20 to 01-01-21 (hint: use Pandas date_range function).

Like:-

_

2020-01-06

2020-06-11

2020-02-18

Part 02. Dedupe it and calculate the number of duplicates and convert it to percentage.

Then by using regex, filter values where either the month is 02,05,09 OR the date is 01,04,07 - the apply function should not be used.

Finally, calculate the percentage of values filtered for the month condition and the date condition.

Question 2:

This question has two parts, a) Sentence Validator and b) Name Reducer

a) Create a sentence ***validator*** function. The validator should return input if it is valid otherwise return False

   Validation Criteria:

1. Start letter must be an uppercase letter and it should follow either a lowercase letter or a single whitespace.
2. All letters in the sentence except the start letter must be in lowercase.

3. The last character (aka terminal character) of the sentence must be any of the following: . (dot) ? (question mark) ! (exclamation mark)
4. Words must be separated with a single whitespace. If there is a hyphen between any two words then there should be one whitespace before and after that hyphen.

    eg: **Lab - 1** is valid, but **Lab- 7** and **Lab -7** are invalid

b) Write a ***reducer*** function to clean the sentences. Reducer takes output from the validator function as input and performs the following cleaning steps,
   - Removes terminal characters (see above validation criteria for list of allowed terminal characters)
   - Removes all duplicated *word groups (see below examples)* but keep its first occurence
   - Removes all leading and trailing whitespaces and hyphens

After completing the functions for part a & b. Create a function ***check_and_clean*** which takes a sentence as input and validates (using validator function) and returns reduced string (output from reducer function) when the sentence is valid, *"<invalid>"* otherwise.

Note: You are not allowed to use regex for this question.

Example 1:

Input: Melo diagnostics melo Labs

Sentence is ***invalid***, failed validation criteria 2 & 3

Output: invalid

Example 2:

Input: Melo diagnostics - southpark east 29th street - southpark east 29th street - free drug testing not offered.

Sentence is ***valid***

Word groups:

Melo diagnostics
southpark east 29th street
free drug testing not offered

Output: Melo diagnostics - southpark east 29th street - free drug testing not offered

Example 3:

Input: Simple labs - covid test available - west side hospital - covid test available!

Sentence is ***valid***

Word groups:

Simple labs
covid test available
west side hospital

Output: Simple labs - covid test available - west side hospital

Question 3:

A sample data of posts of random users are given in this link: click here to download
Post_id, date of post and post caption details are available in the sample dataset. Create a function that extracts posts older than 13/11/2021 and finds the 3 most frequently used special characters out of it. The function should return the 3 most frequently used special characters and the number of times they occurred in the filtered data.

Example

post_id	date	caption
post #1	11/11/2021	@bla bl@ bla! 23🔥
post #2	15/11/2021	Foo b@r foOB!a🙌
post #3	12/11/2021	🔥 aerrt!! Qwe r rr$
post #4	13/11/2021	@momo bati$t@🔥

Output

[(@, 4), (!, 3), (🔥, 2)]

Explanation

The post #2 is eliminated since it is older than the date 13/11/2021. In the remaining 3 rows, the special character “@” occurred the most i.e, 4 times in the posts #1 and #4. The second most frequent special character is “!” which occurred 3 times and then “🔥” occurred twice.

Since only the top 3 most frequent ones are required, the remaining special characters “🙌” and “$” are ignored.