

Part 1: Keyword Search

Please find the attached file **keywords.txt**. In this part of the assignment, we will implement a class for keyword searches: **Given a data string as input, find all instances of the keywords in the input string.**

Instructions

- Find *case insensitive* instances of the keywords.
- Find *whole words* only:
 - If you're searching for *'sport'*, do not find *'transport'* (false positive).
 - If you're searching for *'design'*, do not find *'designated'* (false positive).
- Keywords can contain spaces (for instance: *'Cyber Security'*).
The search algorithm should be able to find minor alterations, such as a dash (*' - '*) instead of space (*' '*), or the elimination of the space character. In this case you should also be able to also find *'cyber-security'*, *'cybersecurity'*.

Output

- The search results should be a list of keywords found without repetitions.

Hint: Use any of the built-in python libraries, especially where it makes your work easier!

Input and Output Examples

[the keywords are read from the file **keywords.txt**]

```
Input: 'Welcome to >>GENERAL-motors! We love programming!'
```

```
Output: ['general motors', 'programming']
```

```
Input: 'Beside being a team focused on cyber-security, we also do software engineering. With good communication we might figure out some unsolved problems in computer-science!'
```

```
Output: ['communication', 'cyber security', 'computer science', 'software engineering', 'unsolved problems in computer science']
```

Part 2: Rate Limiting

Create a multithreaded class with workers that get a random page from Wikipedia and find the keywords from *keywords.txt* anywhere in the response (you can use your search implementation from Part 1).

- The configuration file should have a **parameter** for **number of workers**.
- Each worker runs in its own thread. You must always have this number of workers running.
- You can use additional threads beside workers.

Hint: This URL will get you a random Wikipedia page every time:

`https://en.wikipedia.org/wiki/Special:Random`

Rate Limiting

Since we don't want to burden Wikipedia's servers, we will use **rate limiting** for our requests.

- The configuration file should have a **parameter** for the maximum number of allowed **requests per second** (*across all workers*). Make sure the program doesn't exceed this rate of requests.
- Don't stop or start new workers – the “**number of workers**” parameter still applies.
- There might be more workers than allowed requests per second. The rate limit still applies regardless of how many workers are available. (Hint: Workers can wait or become blocked).
- If you can, don't send workers to 'sleep'. Try avoiding 'Busy Waiting' where possible.

Output

The machine should run until you manually stop it. For each result, print:

- The current worker number.
- The URL.
- The keyword matches found in the page.

Output Example

```
Worker: 1
Random URL: https://en.wikipedia.org/wiki/British_Polar_Engines
Matches: ['construction', 'design', 'manufacturing']
-----
Worker: 5
Random URL: https://en.wikipedia.org/wiki/Nuclear_famine
Matches: ['agriculture', 'biotechnology', 'energy', 'medicine', 'artificial intelligence',
'computer model']
-----
Worker: 2
Random URL: https://en.wikipedia.org/wiki/Cannock_Chase_Railways
Matches: ['construction']
```

GOOD LUCK!