**Assumptions:**

* For any search input, Amazon will only return up to 10 keywords, that have an exact prefix-match with the input.
* Any keyword with a relevant search-volume can be returned by the API
* Whenever the API is called, it operates in 2 following steps:
* Seek: Get all known keywords that match the prefix and create a Candidate-Set
* Sort/Return: Sort the Candidate-Set by search-volume and return the top 10 results.

**\*hint**

* I think the order of the 10 returned keywords is **significant.**
* Because, the resulting candidate set is sorted in descending order from highest searched to lesser. So, I think the order or the order of the 10 returned keywords is important.

The score returned are fairly accurate because of the logic used below,

**Logic:**

1. First thing, I have used the amazon search API link provided to get the response json with the search results.

Given URL: <https://completion.amazon.com/search/complete>

Exact URI: <https://completion.amazon.com/search/complete?search-alias=aps&client=amazon-search-ui&mkt=1&q=search+keyword>

1. A Get request to the above URL with the actual search keyword will respond with a json containing a maximum of 10 top relevant search keywords

**Example:**

<https://completion.amazon.com/search/complete?search-alias=aps&client=amazon-search-ui&mkt=1&q=iphone+charger>

**Response json:**

[

"iphone charger",

[

"iphone charger",

"iphone charger cable",

"iphone charger 10 ft",

"iphone charger 6 ft",

"iphone charger apple certified",

"iphone charger and headphones adapter",

"iphone charger with wall plug",

"iphone charger 3ft",

"iphone charger pack",

"iphone charger wuxian"

], [{"nodes":[{"name":"Electronics","alias":"electronics"}]},{},{},{},{},{},{},{},{},{}],[],"1L0OEDHQT4YDD"]

1. Java 11’s new HTTP Client that takes advantage of async programming concept **CompletableFuture** is used to communicate with Amazon REST API
2. This response json contains a list of different type objects. Hence cannot be directly parsed into a JSON Class.
3. Hence the 10 search keywords are manually parsed and extracted as a list of strings
4. Assigned a direct score in the multiples of 10s between 0 to 100 based on the keyword raking in response JSON.
5. If the exact keyword is not found in the top 10 results, a score of 7 is assigned for each of the partial keyword.