Instructions:

* Problem Definitions:
  + A KWIC index is formed by sorting and aligning the words within an article title to allow each word (except the stop words) in titles to be searchable alphabetically in the index.
  + The KWIC index system accepts an ordered set of lines, each line is an ordered set of words, and each word is an ordered set of characters. Any line may be "circularly shifted" by repeatedly removing the first word and appending it at the end of the line. The KWIC index system outputs a listing of all circular shifts of all lines in alphabetical order.
* Based on the information provided in the paper, as a group, implement KWIC in an assigned style.
  + The instructor will assign the style to your team.
  + Remember that you only need to implement the indexing.
  + Provide one report as a group.

Resources:

* Garlan and Shaw: Pg. 16-22, contains the answer for each implementation type and their comparisons.
* <https://web.archive.org/web/20070820133530/http://www.acm.org/classics/may96/>
* <https://en.wikipedia.org/wiki/Key_Word_in_Context>

Example Input:

* Wikipedia, The Free Encyclopedia
* KWIC is an acronym for Key Word In Context, the most common format for concordance lines.

Expected Output:

|  |  |
| --- | --- |
| KWIC is an | **acronym** for Key Word In Context, ... |
| ... Key Word In Context, the most | **common** format for concordance lines. |
| ... the most common format for | **concordance** lines. |
| ... is an acronym for Key Word In | **Context**, the most common format ... |
| Wikipedia, The Free | **Encyclopedia** |
| ... In Context, the most common | **format** for concordance lines. |
| Wikipedia, The | **Free** Encyclopedia |
| KWIC is an acronym for | **Key** Word In Context, the most ... |
|  | **KWIC** is an acronym for Key Word ... |
| ... common format for concordance | **lines**. |
| ... for Key Word In Context, the | **most** common format for concordance ... |
|  | **Wikipedia**, The Free Encyclopedia |

Minimal Viable Product

1. Input (DONE)
   1. Take an ordered set of lines, comprised of words and characters
2. Circular Shift (DONE)
   1. Repeatedly removing the first word and appending it to the end of the line.
   2. So, each string containing N words gets copied into N different strings, each with a slightly different shift.
   3. We should store each string as a key-value pair with its first word, that way we can just order the keys.
3. Alphabetizing (DONE)
   1. Then we alphabetize all the keys.
4. Output (DONE)
   1. A listing of all circular shifts of all lines in alphabetical order.
5. Master Control (DONE)
   1. Sets up and sequences the subfunctions.
   2. Everything is stored in shared storage.

Additional Stuff (If There’s Time):

* Somehow center the string on its word. (DONE)
* Bold the focused word. (DONE)
* Add ellipses if truncating string (DONE)
* Remove junk words.
* Make it truncate only at the end of the words
  + I.e., if word would exceed length limit, stop before.