**Technical Design Document**

**Name:** Matthew Pocrnic  
**Date Created:** 6/15/2025

### **Program Description:**

This program allows the user to input a paragraph of text. It uses regular expressions to identify sentence boundaries and split the paragraph into individual sentences. The sentences are then displayed on separate lines, each labeled with its sequence number. The total number of sentences identified is also shown. The program is particularly useful for preprocessing text in natural language processing (NLP) tasks or for any application that requires sentence-level parsing.

### **Functions used in the Program:**

#### 1. Function Name: split\_into\_sentences

Description:  
 Extracts individual sentences from a block of text using regular expressions. It is designed to handle basic punctuation (e.g., ., ?, !) and avoid breaking on common non-ending cases like abbreviations or numbers.

Parameters:

* text (str): The input paragraph provided by the user.

Variables:

* sentence\_pattern (str): Regular expression used to match sentences.
* sentences (list): List of sentence strings found using re.findall().

Logical Steps:

* Define a regular expression to capture sentence boundaries based on punctuation and spacing.
* Apply re.findall() with appropriate flags to extract all matches.
* Return the list of matched sentences.

Returns:

* list: A list of sentence strings.

#### 2. Function Name: print\_sentences

Description:  
 Displays each sentence on a new line with a number prefix and prints the total number of sentences found.

Parameters:

* sentences (list of str): The list of sentences to display.

Variables:

* i (int): Index used to number the sentences.
* sentence (str): Individual sentence to print.

Logical Steps:

* Iterate through each sentence using enumerate() to keep track of numbering.
* Strip extra whitespace from each sentence.
* Print the sentence with its corresponding number.
* After all sentences are printed, display the total count.

Returns:

* None

#### 3. Function Name: main

Description:  
 Handles user input and orchestrates the flow of the program by calling the sentence-splitting and printing functions.

Parameters:

* None

Variables:

* paragraph (str): User-provided paragraph of text.
* sentences (list of str): Output from split\_into\_sentences().

Logical Steps:

* Prompt the user to input a paragraph of text.
* Pass the paragraph to split\_into\_sentences() to get a list of sentences.
* Pass the list to print\_sentences() for display.

Returns:

* None

### **Logical Steps (Program Flow):**

1. The program begins by calling the main() function.
2. The user is prompted to enter a paragraph of text.
3. The paragraph is passed to the split\_into\_sentences() function.
4. This function uses a regular expression to extract sentences based on punctuation and formatting rules.
5. The resulting list of sentences is passed to the print\_sentences() function.
6. Each sentence is printed with a sequence number, and the total number of sentences is displayed.
7. The program ends after displaying the output.

**Link to my COP2373 repository:** [**here**](https://github.com/mpocrnic/COP2373)

Screenshot of output from running code

