Text2Compress Documentation

# Challenges

* For decoding, the process of looping through the rules was extremely time consuming and frustrating.
  + Iterating through the sequence and testing different case scenarios led to nearly an hour of debugging, console outputs, etc. to figure out issues.
* As for encoding, doing test cases was nearly impossible, as the sequence would not render properly in the C++ console.

## AI Usage

* Generative AI Usage was limited to clarification on a few paradigms. Generative AI was not used on the main functions/methods but was instead used on helper functions (Ex. shiftLeft(), shiftRight()).
* Prompts used:
  + “With a given array, how do I shift elements to enter two more elements”.
  + “With a given array, how do I shift elements when I want to remove the the element directly next to it”.

# Testing

1. Basic Function Tests

* The resetRules and resetSeq functions make sure arrays get cleared.
* shiftLeft and shiftRight functions check that elements move correctly and length updates.

2. Initialization

* Input a short line (like “ab”). Sequence should store chars and frequency counts update.
* Input multiple lines. Sequence should concatenate.
* Edge case: empty input (no lines). (Utilizes cin.fail())

3. Training

* Text with repeats (e.g., "bbba").
* Text with no repeats (e.g., "abcd"). Should create no rules.
* Training with k=0. This should do nothing.

4. Encode / Decode

* Multiple rules applied which still returns original text after decode.
* If there are no rules then text stays the same.

5. Edge Cases

* Negative or zero inputs for k or lines.
* Non-ASCII characters (outside 127).
* Empty sequence