/\* ---------------------------------------------------------------------------

\*\* I attest that this following code represents my own work and is subject to

\*\* the plagiarism policy found in the course syllabus.

\*\*

\*\* Class: CSC 242

\*\* Assignment: Signature Assignment

\*\* File: source.cpp

\*\* Description: Work with team members to create a spell checker file, and a \*\* program that encrypts a word with a cipher alphabet and decrypts it back \*\* to its original structure

\*\*

\*\* Authors: Jared Lloyd, Joshua Najera, Lorenzo Robles

\*\* Date: August 29, 2025

\*\* -------------------------------------------------------------------------\*/

Pseudocode: Monoalphabetic cipher

Encrypt keyword with Cipher alphabet

1. Enter proper headers
2. // Step 1: Read the keywords from the user
   1. Prompt “Enter a Keyword”
   2. Read out the keyword
3. / /Step 2: Initialize variables
   1. Key 🡨 empty string (will hold the final cipher alphabet)
   2. Enter set “seen” 🡨 empty set of chars (will track letters that are already used)
4. //Step 3: Add unique letters from the keyword (convert letters to uppercase)
   1. For every character ch in the keyword, assign the uppercase version of ch
   2. If the current character is a letter and not yet added to the cipher, then add it and mark it (to prevent repetition)
5. //step 4: Add the remaining letters of the alphabet in reverse order
   1. For each letter from Z to A, if the letter is not seen or used, then add to the cipher
   2. Mark the used letter to prevent repetition
6. // Step 5: Output the cipher alphabet
   1. Print the cipher alphabet and add final cipher alphabet (key)

Decrypt keyword Cipher

1. // Step 1: Generate cipher alphabet ( all same as encryption)
   1. // Step 1: Read the keywords from the user
   2. Prompt “Enter a Keyword”
      1. Read out the keyword
      2. / /Step 2: Initialize variables
         1. Key 🡨 empty string (will hold the final cipher alphabet)
         2. Enter set “seen” 🡨 empty set of chars (will track letters that are already used)
   3. //Step 3: Add unique letters from the keyword (convert letters to uppercase)
      1. For every character ch in the keyword, assign the uppercase version of ch
      2. If the current character is a letter and not yet added to the cipher, then add it and mark it (to prevent repetition)
   4. //step 4: Add the remaining letters of the alphabet in reverse order
      1. For each letter from Z to A, if the letter is not seen or used, then add it to the cipher
      2. Mark the used letter to prevent repetition
   5. // Step 5: Output the cipher alphabet
      1. Print the cipher alphabet and add the final cipher alphabet (key)
2. // Step 2: Read the cipher text to initiate decryption
   1. Enter the cipher text
   2. Read the cipher text to initiate
   3. Decrypted text will enter the string
3. // Step 3: Decrypt each character
   1. For each character (ch) in the cipher text
      1. Insert the uppercase version of ch
      2. If the letter is uppercase
         1. Locate where the letter appears in the cipher alphabet
         2. Return the corresponding letter in the normal alphabet
         3. Take the encrypted letter and add it to the end of the decrypted message (build the final word one character at a time)
      3. Else keep the non-letters unchanged
4. // Step 4: Output the decrypted text
   1. Print out (“Decrypted text: “ + decryptedtext)