OBJECTIVE:

The objective of this program is to convert a sentence of regular text into texting shortcuts. User input is the sentence to be translated, and the output from the program will be the sentence with the translated phrases. The phrases to be translated are “see you” (cu), “see” (c), “you” (u), “later” (l8r), “let me know” (lmk), “good luck” (gl), and “by the way” (btw).

1. Pre-steps: Create 3-word, 2-word, and 1-word dictionary functions. The 3-word dictionary (function) will check if the first 3 words are matched to “Let me know”, or “By the way”. If this matches to either, it will return true, and translate the phrase by writing the substitution into a new string. This new string will be passed as reference so that changes made will persist outside the function. If there are no matches, it will return false. The 2-word dictionary will check for matches between the first two words, and “see you”, or “good luck”. The 1-word dictionary will check for a match between the first word and “see”, you”, or “later”. Set the initial index to 0. This value will increase word by word as we check each part of the sentence.
2. Prompt user for input, and store the input sentence into a character array
   1. Cin.get will be used to allow for white spaces to be included to differentiate words. Up to 200 characters will be stored into the character array, or until a newline is entered.
3. Split this input sentence into words
   1. Use isblank function to determine when each respective work ends. Figure out the ends of each word by incrementing the index each time in a loop and checking for whitespaces. If there is a whitespace, then the loop condition will not be satisfied, and the index of the end of the word will be saved.
   2. Concatenate the words together into a separate array. There will be separate functions for concatenating 3 words and 2 words, depending on which step is required. If you need to concatenate the first 3 words, concatenate the first word with a white space. Then concatenate this with the second word. Repeat until all three words are inputted into the new array with spaces in between. Similarly, do the same with 2-word concatenation, but stop at 2 words. There is no function for concatenating a single word, as it is not necessary.
4. Check for matches from the concatenated phrase to the phrases that need to be translated.
   1. Take the array of the first three words, and check if it is equal to “Let me know”, or “By the way”. If there is a match, then in a new array, add the 3-word phrase’s equivalent text form (“lmk” or “btw”). Change the index value to the size of the 3-word array plus 1. Skip steps 4b and 4c.
   2. If there are no matches, then take the array of the first two words and check for matches to “see you” or “good luck”. If there is a match, then in a new array, add the 2-word phrase’s equivalent text form (“cu” or “gl”) to the end of anything prior. Change the index value to the size of the 2-word array plus 1. Skip step 4c.
   3. If there are no matches so far, then take the first word and check if it is “see”, “you”, or “later”. If there is a match, then in a new array, add the 1-word phrase’s equivalent text form (“c”, “u”, or “l8r”) to the end. Change the index value to the size of the 1-word array plus 1.
   4. If none of the first 3 words match any text shortcut, then advance the index value to the end of the first word plus 1 (same as step 4c). Put in the three words that didn’t match into the translated array.

\*\* All of these text-shortened arrays created within the functions should be passed as reference to remain usable in the main function. \*\*

1. Repeat steps 3b – 4c with the new index value. This new index value should be either at the end of the first 3 words, 2 words, or first word, depending on the phrases that potentially matched. This should be repeated until the end of the sentence is reached.
2. Display the translated sentence to the user, by displaying the translated array that is being appended to in each dictionary function.

Repeat until end of array

Display output to screen

Yes

Yes

No

No

Increase index value

In new array, add “gl” or “cu”, depending on match.

Check if equal to “good luck” or “see you”

In new array, add “lmk” or “btw”, depending on match.

Increase index value

Enter 1-word dictionary function

Enter 2-word dictionary function

Check if equal to “let me know” or “by the way”

Take initial index value and concatenate first word with next 1

Enter concatenate2 function

Take initial index value and concatenate first word with next 2

Enter concatenate3 function

Enter 3-word dictionary function

Input sentence into char array

Prompt user for sentence

Start