MY PSEUDOCODE Lab5

Format of file :

<l or r><shift number (an integer)><first character of plaintext><second character of plaintext>...<last character of plaintext>

1. Reading the Encryption Key from the plaintext file:

   { using **'std::ifstream'** and declaring an object to read the **'input.txt'** file reading from a file letter by letter requires to define a variable as char and then reading from the file with the shovels ' >> ' }

   a. The **first letter denotes the shift direction** ('l' = -ve, 'r' = +ve)
   b. The **next number shows the shift number.**

2. Handling Errors:
   a. Take the inputs until the file ends.
      `while(fileobject >> variable)`

   b. The function `number_from_letter` is used to find the index of the plaintext character.

   c. Add an if else statement after reading the first character that checks if the letter exists and if it does, check if it is 'l' or 'r'. This if structure will be used to print the correct 'Oops' statement.
      `(logic statements input == 'l'|| input == 'r')`

   d. Add one more if else statement after reading the number or second character that reads that a number exists and that the file does not end. Again this statement will be used to print the Oops statement that denotes that the number is not present.
      `if(fileobject >> variable) { code that calls the function }`

   e. Index of ciphertext is longer than the alphabet length.
      ```
      number = (number + shift) % 26;
      ```

   f. Index of ciphertext is shorter than the alphabet length
      ```
      number += 26 (alphabet length);
      ```

   g. `letter_from_number` for to find the encrypted letter using `number` as the parameter and getting the encrypted character

3. Encrypting the character:
   a. Defining a function helps as I will encrypt every character one by one.

      i. The function should have the parameters: character to be encrypted (char), shift direction (int) and the shift number (int).

ii. This function returns the character that is encrypted with the given parameters.

4. Printing the cipher text:
    a. use `std::cout` to print every encrypted character returned by the function and do not use `endl`.