NUM\_LETTRS = 26;

Encrypt formula: Encrypt = (pTextString + shift) mod 26

Decrypt formula: Decrypt = (pTextString - shift + 26) mod 26

Shift formula : Shift = (pTextString - Decrypt + 26) mod 26

**v1** **\_deconstruct1Char:**

* Creates an alphabet array with range [0,25].
* Takes a plaintext string and finds each letters corresponding
* value in the alphabet [0,25] and sets the corresponding letter's array index to a new array.
* Takes an encrypted string and finds each letters corresponding
* value in the alphabet [0,25] and sets the corresponding letter's array index to a new array.
* Uses third letter from plaintext and third letter from cipher and applies
* the Shift formula using the alphabet values of each corresponding letter and
* finds the number of places the plaintext letter was shift. The shift value is
* then converted back to a letter it represents in the key string.

**v2\_deconstruct:**

Loops through the plaintext's length, and compares a letter from plaintext and the corresponding letter from cipher and applies the Shift formula using the alphabet values of each corresponding letter and finds the number of places the plaintext letter was shift. The shift value is then converted back to a letter it represents in the key string.

**v3.1\_createKey\_encrypt1Char:**

* Put all the code to find the key by deconstructing the plaintext and encrypted strings into getKey().
* Takes a plaintext string and finds each letters corresponding value in the alphabet [0,25] and sets the corresponding letter's array index to a new array.
* Takes each letter of the plaintext string and applies the Encrypt formula by adding each plaintext value with a shift value, then modulus 26 to calculates the alphabet value of the encrypted letter. Encrypted value is then converted back into a letter that it represents and saves it to the key string. The key is 5 letters long, and it is repeated to match the length of the plaintext.

**v3.2\_createKey\_encrypt:**

* Takes a plaintext string and finds each letters corresponding value in the alphabet [0,25] and sets the corresponding letter's array index to a new array.
* Loops through the plaintext's length, and applies the Encrypt formula by adding each plaintext value with a shift value, then modulus 26 to calculates the alphabet value of the encrypted letter. Encrypted value is then converted back into a letter that it represents and saves it to the key string.
* The key is 5 letters long, and it is repeated to match the length of the plaintext.

**v4\_decrypt:**

* Takes an encrypted string and finds each letters corresponding value in the alphabet [0,25] and sets the corresponding letter's array index to a new array.
* Takes a given key string and finds each letters corresponding value in the alphabet [0,25] and sets the corresponding letter's array index to a new array. The key is 5 letters long, and it is repeated to match the length of the encrypted.
* Loops through the encrypted string length, and applies the Decrypt formula by subtracting each decrypted value from a shift value, add 26, then modulus 26 to calculates the alphabet value of the decrypted letter. Decrypted value is then converted back into a letter that it represents and saves it to the decrypted string.