Transposition cipher and brute force attack

We had designed a system consisting of transposition encryption, decryption, hashing and brute force attack to find the corresponding key. We had given input of five original strings and developed a program to find the key that decrypts all five ciphertexts using brute force attack by trying out all possible combinations of keys. Transposition requires filling elements of texts in a matrix whose
Columns = key_length, rows = (text_length + key_length- (text_length%key_length))/key_length.

Below is in detail description of each function of our code:

Main: In the main function, we had given five strings of original text as input to the array. Then finding their corresponding hash strings using hash function. Plaintext is generated by concatenating the original string with its corresponding hash string. Then corresponding cipher_texts are generated using encryption of plaintexts. Then store corresponding five hash strings in array hash_str and cipher_texts in ciphertext array. Then call bruteforceattack function to find the key that decrypts all ciphertexts by passing the two arrays ciphertext, hash_str as parameters.

Hash: The function `hash_fun` takes a constant reference to a string (`input`) and calculates a hash value by summing up the ASCII values of its characters. It then generates an 8-character hash string using lowercase alphabets by performing some arbitrary hashing operations on the calculated hash value.The final hash string is constructed and returned by the function.

Join: This function is created for concatenating the original string and its hash_string.

encrypt: This function takes two strings namely plaintext and key as parameters. Transposition encryption involves the given plaintext to be filled row wise in a matrix of column=key_length and then write down column wise according to priority of its corresponding key value starting from least giving us the ciphertext. Using a matrix, filling the elements row-wise and padding leftover positions in the last row with element ‘x’. Then adding elements of matrix to string ciphertext column-wise according to priority of corresponding element of key. The string ciphertext is returned.

decrypt: This function takes two strings namely ciphertext and key as parameters. Transposition decryption involves the given ciphertext to be filled column-wise according to its corresponding key element starting from least value and then finally write down row-wise excluding the padded elements which are present at the last. Using a matrix, filling the elements column-wise and then adding elements of matrix to string plaintext row-wise neglecting the padded elements at the end. The string plaintext is returned.

is_recognizable: This function takes a string plaintext as parameter. It checks if all elements of this string are lowercase and there are no spaces in between and return true or false correspondingly.

Bruteforceattack: This function tries to find the key by taking input as an array of ciphertexts and hash_strings. As we don't know the exact key_length but only know the range (1-9), we try all the lengths by using a loop. As there are 'k!' Keys possible for key_length 'k' so we use a nested loop to try all possible combinations. Then generated possible keys which adds elements in range of (0-9) to make keys. Then try to decrypt all five ciphertexts to generate corresponding plaintexts. We split the plaintext we got to extract its corresponding hash_string to compare it with hash_str which we had already had. If both the hash strings are equal and the plaintext is recognizable, then proceed to the checking of the next plaintext and increment variable 'count'. If at any time the conditions are not satisfied, then restart with a new different key. If for any key count==5 then it means that the key decrypts all ciphertexts successfully and finally prints that key.