For my transposition and substitution cipher I chose to start with coding the substitution part first. I created a variable called alphabet and set it equal to the alphabet in correct order. Then, I created another variable for the key and rearranged the alphabet in random order. For the substitution I coded a for loop that ran through the alphabets lower letters and then replaced the text with its aligned letter in the key variable and printed the result. Moving on to the transposition I created a function that took in the result from the substitution cipher in its parameters. From there I created more variables consisting of a keyword for the matrix, an empty matrix, and the length of the result. Other variables I included were the list of the result the list of the keyword and a starting index of 0. From there I created the col of the matrix with the length of the keyword and the row length using the math.celi function and dividing the length of the word by the column. Following that I accounted for spaces in the word that is being ciphered and this part of the cipher I had to look up but from my understanding it adds a padding to the empty cell of the matrix. Then I created the matrix itself by taking the list of the result and creating a column for every letter or space. I ran a loop after that to fill the empty spaces and create the rows from the length of the column. The loop takes the key and applies It to the matrix and creates the transposition of the result from the substitution cipher. Then all that needed to be done was to return the cipher and print it to the user.