**/\* Experiment No - 02 Program \*/**

import math  
  
  
def encryptMessage(key, message):  
 ciphertext = [''] \* key  
  
 for col in range(key):  
 position = col  
 while position < len(message):  
 ciphertext[col] += message[position]  
 position += key  
 return ''.join(ciphertext) # Cipher text  
  
  
def decryptMessage(key, message):  
 numOfColumns = math.ceil(len(message) / key)  
 print("num of rows = ", len(message) / key)  
 print("num of rows = ", numOfColumns)  
 numOfRows = key  
 print("num of columns = ", numOfRows)  
 numOfShadedBoxes = (numOfColumns \* numOfRows) - len(message)  
 print("numOfRows = ", numOfShadedBoxes)  
 plaintext = [' '] \* numOfColumns  
 col = 0  
 row = 0  
  
 for symbol in message:  
 plaintext[col] += symbol  
 print(" plaintext= ",plaintext)  
 col += 1  
 if (col == numOfColumns) or (col == numOfColumns - 1 and row >= numOfRows - numOfShadedBoxes):  
 col = 0  
 row += 1  
 return ''.join(plaintext)  
  
  
myMessage = 'Transposition Cipher'  
myKey = 3  
print("Encryption Process starts")  
print("Plain Text is: Transposition Cipher ")  
ciphertext = encryptMessage(myKey, myMessage)  
print("Length of message ",len(myMessage))  
print("Cipher Text is==>", ciphertext)  
print("Decryption Process Starts:")  
  
pt = decryptMessage(myKey, ciphertext)  
print("Plain Text is==>", pt)