

```

def do(mode, message, keyword):

    # ERROR HANDLING
    isMode = mode in ('ENCODE', 'DECODE')
    isMessage = type(message) is str and len(message) > 0
    isKeyword = type(keyword) is str and len(keyword) > 0
    if not (isMode and isMessage and isKeyword):
        return 'ERROR'

    # VARIABLES SETUP
    n = len(message)
    m = len(keyword)
    omega = [''] * m
    priority = []
    output = ''

    # GET THE PRIORITY ORDER USING THE KEYWORD
    sorted_keyword = sorted(keyword)
    for char in keyword:
        p = sorted_keyword.index(char)
        priority.append(p)
        sorted_keyword[p] = None

    if mode == 'ENCODE':

        # WRITE OMEGA - PRIORITY ORDER - CHAR AT A TIME
        for i in range(n):
            p = priority[i % m]
            char = message[i]
            omega[p] += char

        # READ OMEGA - ASCENDING ORDER - STRING AT A TIME
        for i in range(m):
            string = omega[i]
            output += string

        # THE END
        return output

    if mode == 'DECODE':

        # WRITE OMEGA - ASCENDING ORDER - STRING AT A TIME
        length = n // m
        deadline = n % m
        start = 0
        for i in range(m):
            extra = priority.index(i) < deadline
            end = start + length + extra
            string = message[start:end]
            omega[i] += string
            start = end

        # READ OMEGA - PRIORITY ORDER - CHAR AT A TIME
        for i in range(n):
            p = priority[i % m]
            string = omega[p]
            char = string[i // m]
            output += char

        # THE END
        return output

```

```

if __name__ == '__main__':

    print('Welcome to the Columnar Transposition Cipher')
    print('* Enter 1 to Encode')
    print('* Enter 2 to Decode')
    print('* Enter anything else to terminate')

    while True:

        entry = input('\n' + 'Entry: ')

        if entry == '1':
            print('=> ' + do('ENCODE', input('Plaintext: '), input('Keyword: ')))
            continue

        if entry == '2':
            print('=> ' + do('DECODE', input('Ciphertext: '), input('Keyword: ')))
            continue

        print('=> thank you for using me bye <3')
        break

```

"""

*Columnar Transposition Cipher Python Implementation*  
*By Mohammad H. Alomar*  
*31 December 2022 @ 4:13 AM*

"""

