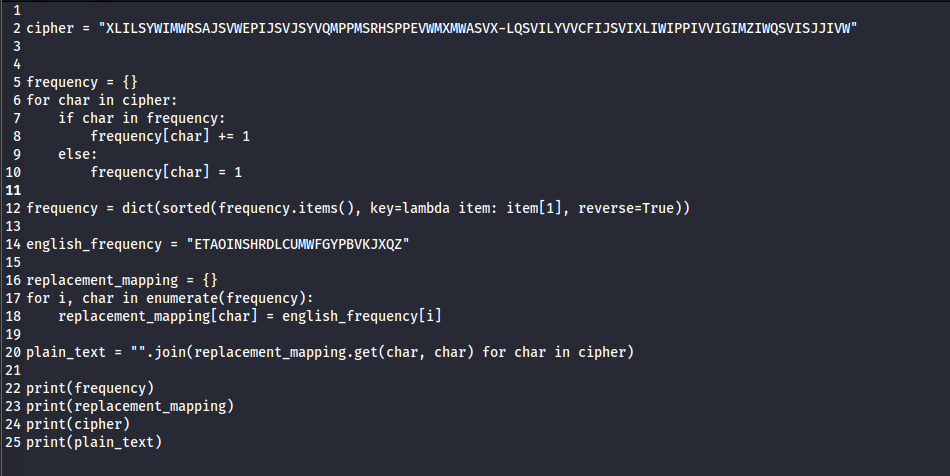
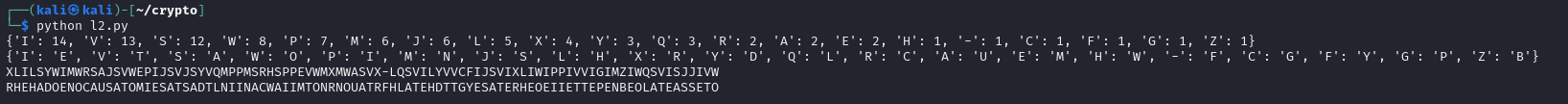
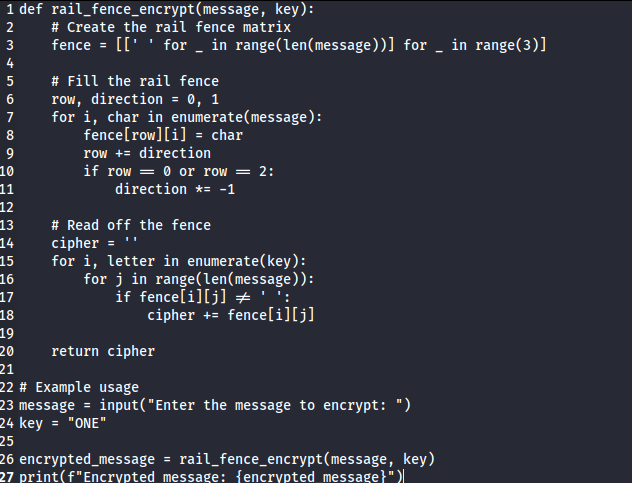
1. Assume you intercepted the following ciphertext. Using a statistical attack, find the plaintext

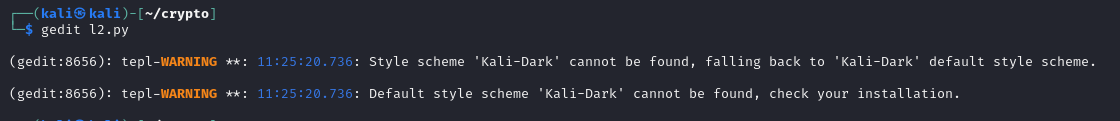
"XLILSYWIMWRSAJSVWEPIJSVJSYVQMPPMSRHSPPEVWMXMWASVX-LQSVILYVVCFIJSVIXLIWIPPIVVIGIMZIWQSVISJJIVW"



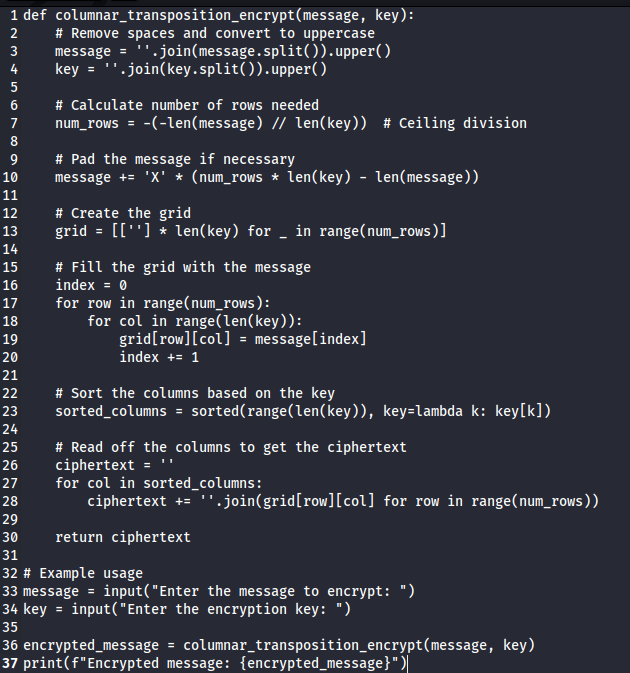


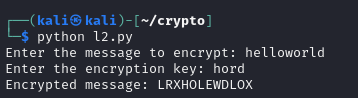
2. Write a Python script to encrypt using Rail Fence (Zig zag ) with three rows and with key (ONE).





3. Write a python script to encrypt columnar transposition





4. Write a Python script to decrypt Rail Fence Cipher

def rail\_fence\_encrypt(message, key):

# Create the rail fence matrix

fence = [[' ' for \_ in range(len(message))] for \_ in range(3)]

# Fill the rail fence

row, direction = 0, 1

for i, char in enumerate(message):

fence[row][i] = char

row += direction

if row == 0 or row == 2:

direction \*= -1

# Read off the fence

cipher = ''

for i, letter in enumerate(key):

for j in range(len(message)):

if fence[i][j] != ' ':

cipher += fence[i][j]

return cipher

def rail\_fence\_decrypt(cipher, key):

# Create the rail fence matrix

fence = [[' ' for \_ in range(len(cipher))] for \_ in range(3)]

# Mark the spots where characters should be

row, direction = 0, 1

for i in range(len(cipher)):

fence[row][i] = '\*'

row += direction

if row == 0 or row == 2:

direction \*= -1

# Fill the fence with the cipher text

index = 0

for i, letter in enumerate(key):

for j in range(len(cipher)):

if fence[i][j] == '\*':

fence[i][j] = cipher[index]

index += 1

# Read off the fence to get the original message

message = ''

row, direction = 0, 1

for i in range(len(cipher)):

message += fence[row][i]

row += direction

if row == 0 or row == 2:

direction \*= -1

return message

# Example usage

choice = input("Enter 'E' for encryption or 'D' for decryption: ").upper()

key = "ONE"

if choice == 'E':

message = input("Enter the message to encrypt: ")

encrypted\_message = rail\_fence\_encrypt(message, key)

print(f"Encrypted message: {encrypted\_message}")

elif choice == 'D':

cipher = input("Enter the message to decrypt: ")

decrypted\_message = rail\_fence\_decrypt(cipher, key)

print(f"Decrypted message: {decrypted\_message}")

else:

print("Invalid choice. Please enter 'E' or 'D'.")

A screen shot of a computer

Description automatically generated

===============================================================================

"It does not matter how slowly you go so long as you do not stop." —Confucius

===============================================================================