Exp 1d : Columnar Transposition Techniques

Code:

import math

key = "HACK"

# Encryption

def encryptMessage(msg):

cipher = ""

k\_indx = 0

msg\_len = float(len(msg))

msg\_lst = list(msg)

key\_lst = sorted(list(key))

col = len(key)

row = int(math.ceil(msg\_len / col))

fill\_null = int((row \* col) - msg\_len)

msg\_lst.extend('\_' \* fill\_null)

matrix = [msg\_lst[i: i + col]

for i in range(0, len(msg\_lst), col)]

for \_ in range(col):

curr\_idx = key.index(key\_lst[k\_indx])

cipher += ''.join([row[curr\_idx]

for row in matrix])

k\_indx += 1

return cipher

# Decryption

def decryptMessage(cipher):

msg = ""

k\_indx = 0

msg\_indx = 0

msg\_len = float(len(cipher))

msg\_lst = list(cipher)

col = len(key)

row = int(math.ceil(msg\_len / col))

key\_lst = sorted(list(key))

dec\_cipher = []

for \_ in range(row):

dec\_cipher += [[None] \* col]

for \_ in range(col):

curr\_idx = key.index(key\_lst[k\_indx])

for j in range(row):

dec\_cipher[j][curr\_idx] = msg\_lst[msg\_indx]

msg\_indx += 1

k\_indx += 1

try:

msg = ''.join(sum(dec\_cipher, []))

except TypeError:

raise TypeError("This program cannot",

"handle repeating words.")

null\_count = msg.count('\_')

if null\_count > 0:

return msg[: -null\_count]

return msg

# Driver Code

msg = input("Enter the message: ")

cipher = encryptMessage(msg)

print("Encrypted Message: {}".format(cipher))

print("Decrypted Message: {}".format(decryptMessage(cipher)))

Output:

