*#----------------------------------------------------------------------------------  
#Rutvik Parmar  
#H-41  
#Date - 25th Feb 2016  
#Assignement No:5 – S-RC4 Cipher implementation  
#----------------------------------------------------------------------------------*

import base64

import sys

#RC4 Implementation

def rc4\_crypt( data , key ):

S = range(256)

j = 0

out = []

#KSA Phase

for i in range(256):

j = (j + S[i] + ord( key[i % len(key)] )) % 256

S[i] , S[j] = S[j] , S[i]

#PRGA Phase

i = j = 0

for char in data:

i = ( i + 1 ) % 256

j = ( j + S[i] ) % 256

S[i] , S[j] = S[j] , S[i]

out.append(chr(ord(char) ^ S[(S[i] + S[j]) % 256]))

return ''.join(out)

# function that encrypts data with RC4 and decodes it in base64 as default

# for other types of data encoding use a different encode parameter

# Use None for no encoding

def encrypt( data , key , encode = base64.b64encode ):

data = rc4\_crypt(data , key)

if encode:

data = encode(data)

return data

# function that decrypts data with RC4 and decodes it in base64 as default

# for other types of data encoding use a different decode parameter

# Use None for no decoding

def decrypt(data , key, decode = base64.b64decode ):

if decode:

data = decode(data)

return rc4\_crypt(data , key)

def main():

if len(sys.argv) != 3:

print (" Usage: ")

return 1

data = tuple(sys.argv[1])

key = tuple(sys.argv[2])

data = encrypt(data, key)

print(data)

data = decrypt(data, key)

print(data)

return 0

if \_\_name\_\_ == '\_\_main\_\_':

main()

Output-

Rutviks-MacBook-Pro:~ rutvikparmar$ python rc4.py rutvikparmar11 rc4cipher

#Encryption - q+JJkm9zpeoLatXpwaA=

#Decryption - rutvikparmar11