#----------------------------------------------------------------------------------

#Rutvik Parmar

#H-41

#Date - 2nd Feb 2016

#Assignement No:1 - Caeser Cipher implementation for alphanumeric input

#----------------------------------------------------------------------------------

#Import datetime for calulating key from time viz. a variant of second

from datetime import datetime

#This function helps us to select the mode either decryption or encryption

def getMode():

while True:

print('Do you wish to encrypt or decrypt a message?')

mode = raw\_input().lower() #raw\_input() is used to take users input from the terminal

if mode in 'encrypt e decrypt d'.split():

return mode

else:

print('Enter either "encrypt" or "e" or "decrypt" or "d".')

#This fynction gets the message from the terminal from the user

def getMessage():

print('Enter your message:')

return raw\_input()

#This function helps to generate the key from the datetime library

def getKey():

key = 0

while True:

key = datetime.now().second #This takes the second part from the datetime.now()

print('Your Key is %s' %(key%26))

if (key >= 1 and key <= 60):

k = key%26 #Key%26 is done to bring the second between 0-26

return (k)

#This is the main function used to translate the message

def getTranslatedMessage(mode, message, key):

if mode[0] == 'd': #selects the first element from the mode and checks whether it is decryption or encryption

key = -key

translated = ''

#This for loop works for encryption of message

for symbol in message:

num = ord(symbol) #ord() function is used to get the ASCII value of the symbol in the message

num += key

#The following if's are used for wrap around

if symbol.isupper():

if num > ord('Z'):

num -= 26

elif num < ord('A'):

num += 26

elif symbol.islower():

if num > ord('z'):

num -= 26

elif num < ord('a'):

num += 26

elif symbol.isdigit():

if num > ord('9'):

num -= 9

elif num < ord('0'):

num += 9

translated += chr(num)

return translated

mode = getMode() #Get mode encrypt or decrypt

message = getMessage() #Get message

key = getKey() #Get key

print('Your translated text is:')

print(getTranslatedMessage(mode, message, key))

encrypted = getTranslatedMessage(mode, message, key)

#Implementation of brute force

print('Brute force :')

for key in range(1, 26 + 1):

print(key, getTranslatedMessage('decrypt', encrypted, key))

#Output -

#Do you wish to encrypt or decrypt a message?

#encrypt

#Enter your message:

#afh8219bqjbehh28313hjfb2321

#Your Key is 7

#Your translated text is:

#hmo6987ixqiloo96181oqmi9198

#Brute force :

#(1, 'gln5876hwphknn85070nplh8087')

#(2, 'fkm4765gvogjmm74868mokg7876')

#(3, 'ejl3654funfill63757lnjf6765')

#(4, 'dik2543etmehkk52646kmie5654')

#(5, 'chj1432dsldgjj41535jlhd4543')

#(6, 'bgi0321crkcfii30424ikgc3432')

#(7, 'afh8210bqjbehh28313hjfb2321')

#(8, 'zeg7108apiadgg17202giea1210')

#(9, 'ydf6087zohzcff06181fhdz0108')

#(10, 'xce5876yngybee85070egcy8087')

#(11, 'wbd4765xmfxadd74/6/dfbx7/76')

#(12, 'vac3654wlewzcc63.5.ceaw6.65')

#(13, 'uzb2543vkdvybb52-4-bdzv5-54')

#(14, 'tya1432ujcuxaa41,3,acyu4,43')

#(15, 'sxz0321tibtwzz30+2+zbxt3+32')

#(16, 'rwy/210shasvyy2/\*1\*yaws2\*21')

#(17, 'qvx.10/rgzruxx1.)0)xzvr1)10')

#(18, 'puw-0/.qfyqtww0-(/(wyuq0(0/')

#(19, "otv,/.-pexpsvv/,'.'vxtp/'/.")

#(20, 'nsu+.-,odworuu.+&-&uwso.&.-')

#(21, 'mrt\*-,+ncvnqtt-\*%,%tvrn-%-,')

#(22, 'lqs),+\*mbumpss,)$+$suqm,$,+')

#(23, 'kpr(+\*)latlorr+(#\*#rtpl+#+\*')

#(24, 'joq\'\*)(kzsknqq\*\'")"qsok\*"\*)')

#(25, "inp&)('jyrjmpp)&!(!prnj)!)(")

#(26, "hmo%('&ixqiloo(% ' oqmi( ('")