'''

Keni Mou

Kmou1@binghamton.edu

Assignment 6, Exercise 1

Lab Section 52

CA Kyle Mille

'''

"""

Analysis

to encrypt or decrypt a cipher

Output to monitor:

processedMessage(str)

Input from keyboard:

operator(str)

rotationKey(int)

Tasks allocated to functions:

operationValidated: Check that requested operation is valid

rotationKeyValidated: Check that rotation key is of form <digits> or -<digits> or +<digits>

convertRotationKey: Convert rotation key to value usable for requested operation

keepInBounds: Perform string modulus operation to prevent processed character

processMessage: Encrypt or decrypt message using rotationKey

"""

#Initialize constants ---------------------------------------------------------

OPERATIONS = "edED"

DECRYPT\_STR= "Dd"

ENCRYPT = 1

DECRYPT = -1

# Min and limit ordinals of printable ASCII

PRINTABLE\_ASCII\_MIN = 32

PRINTABLE\_ASCII\_LIMIT = 127

# Allowable rotation key prefixes

KEY\_PREFIXES = "-+"

# Functions ------------------------------------------------------------------

# Check that requested operation is valid

# param opStr (str) - operation requested

# return True when valid, False otherwise (bool)

def operationValidated(opStr):

return opStr in OPERATIONS and len(opStr)==1

# check the lenth to avoid input "ed" at the same time

# Check that rotation key is of form <digits> or -<digits> or +<digits>

# param rotationKeyStr (str)

# invoke str.isdigit()

# returns: True when valid, False otherwise (bool)

def rotationKeyValidated(rotationKeyStr):

return str.isdigit(rotationKeyStr) or str.isdigit(rotationKeyStr[1:]) and \

rotationKeyStr[0] in KEY\_PREFIXES

# Convert rotation key to value usable for requested operation

# param op (str) - operation requested

# param rotationKeyStr (str)

# invoke int()

# return encryption or decryption rotation key (int)

def convertRotationKey(op, rotationKeyStr):

rotationKey=int(rotationKeyStr)

if op in DECRYPT\_STR:

rotationKey = DECRYPT \* rotationKey

else:

rotationKey = ENCRYPT \* rotationKey

## print (rotationKey) #debug

return rotationKey

# Perform string modulus operation to prevent processed character

# from going out of bounds

# param ordinal (int)

# returns adjusted ordinal of new character (int)

def keepInBounds(ordinal):

while ordinal>=PRINTABLE\_ASCII\_LIMIT:

ordinal=ordinal-PRINTABLE\_ASCII\_LIMIT+PRINTABLE\_ASCII\_MIN

while ordinal <PRINTABLE\_ASCII\_MIN:

ordinal=ordinal+PRINTABLE\_ASCII\_LIMIT-PRINTABLE\_ASCII\_MIN

return ordinal

# Encrypt or decrypt message using rotationKey

# param message (str)

# param rotationKey (int)

# invoke

# keepInBounds()

# return processedMessage (str)

def processMessage(message, rotationKey):

processedMessage=""

for alpha in message:

order=ord(alpha)+rotationKey

while order >= PRINTABLE\_ASCII\_LIMIT or order<PRINTABLE\_ASCII\_MIN:

order=keepInBounds(order)

alpha=chr(order)

processedMessage += alpha

return processedMessage

# Main -----------------------------------------------------------------------

# Gets plain text or cipher code, operation requested (encrypt or decrypt),

# and rotation key for Caesar cipher

# Generates cipher code or plain text

def main():

# Describe program

print("This program encrypts or decrypts messages " + \

"using a Caesar cipher")

# Priming read and repeat

message=input("Please input your message\n"\

"or press ENTER to quit")

while message !="" :

#Get remaining inputs, validate and convert as necessary

operator=input("Please enter either 'e' to encrypt or 'd' to decrypt")

while not operationValidated(operator):

operator=input("Invalid input! Please enter either 'e' or 'd'")

#deal with the rotation key

rotationKeyStr=input("Please enter the rotation Key")

while not rotationKeyValidated(rotationKeyStr):

rotationKeyStr=input("Invalid input! Please try agian.")

rotationKey=convertRotationKey(operator, rotationKeyStr)

# Encrypt or decrypt contents of file

processedMessage=processMessage(message, rotationKey)

# Display result

print("Your prosecced message is \n",\

processedMessage, '\n')

# Continuation read

message=input("Please input your message\n"\

"or press ENTER to quit")

print("You ended this program")

main()

Python 3.3.0 (v3.3.0:bd8afb90ebf2, Sep 29 2012, 10:57:17) [MSC v.1600 64 bit (AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>> ================================ RESTART ================================

>>>

This program encrypts or decrypts messages using a Caesar cipher

Please input your message

or press ENTER to quitz

Please enter either 'e' to encrypt or 'd' to decrypte

Please enter the rotation Key25

Your prosecced message is

4

Please input your message

or press ENTER to quit4

Please enter either 'e' to encrypt or 'd' to decryptd

Please enter the rotation Key25

Your prosecced message is

z

Please input your message

or press ENTER to quit4

Please enter either 'e' to encrypt or 'd' to decryptr

Invalid input! Please enter either 'e' or 'd'e

Please enter the rotation Key-25

Your prosecced message is

z

Please input your message

or press ENTER to quit

You ended this program

>>> ================================ RESTART ================================

>>>

This program encrypts or decrypts messages using a Caesar cipher

Please input your message

or press ENTER to quitapple

Please enter either 'e' to encrypt or 'd' to decrypte

Please enter the rotation Key1

Your prosecced message is

bqqmf

Please input your message

or press ENTER to quitasdhlfksd

Please enter either 'e' to encrypt or 'd' to decryptd

Please enter the rotation Key-215454213

Your prosecced message is

i{lptns{l

Please input your message

or press ENTER to quitlskdjf

Please enter either 'e' to encrypt or 'd' to decrypte

Please enter the rotation Key+444

Your prosecced message is

MTLEKG

Please input your message

or press ENTER to quitjdifn

Please enter either 'e' to encrypt or 'd' to decryptklkdj

Invalid input! Please enter either 'e' or 'd'slkdfj

Invalid input! Please enter either 'e' or 'd'454

Invalid input! Please enter either 'e' or 'd'jdj

Please enter the rotation Keyd

Invalid input! Please try agian.sdfalkj

Invalid input! Please try agian.sdflkj

Invalid input! Please try agian.4654

Your prosecced message is

ichem

Please input your message

or press ENTER to quit

You ended this program

>>> ================================ RESTART ================================

>>>

This program encrypts or decrypts messages using a Caesar cipher

Please input your message

or press ENTER to quitjfl

Please enter either 'e' to encrypt or 'd' to decrypted

Invalid input! Please enter either 'e' or 'd'ed

Invalid input! Please enter either 'e' or 'd'slkdjf

Invalid input! Please enter either 'e' or 'd'd

Please enter the rotation Keyjfl

Invalid input! Please try agian.54

Your prosecced message is

406

Please input your message

or press ENTER to quit

You ended this program