'''

Jack Santini

CSC490.DH1

Dr. Mingxian Jin

Fall 2020

Encryption and Decryption Game: Prototype Versions 1 and 2

'''

'''

Prototype 1 versions: lack of self generating messages/sequences and keys.

Tests different means of encryption, input, and output.

Prototype 2 versions: ### ISSUE: Prototype 2 core code from older version of Python. Proper version

editing currently under way. ###

Implements version 1 base ideas, but now has self generating

messages and keys; simple, short password-like sequences.

User input is no longer used in establishing message or keys.

Instead, users must enter the proper key to decrypt the encrypted messages.

Then input their answer into a prompt.

Protoype 3 versions:

'''

'''

Prototype 1: user inputs message and key, gets back encrypted message

with binary representation.

Pros: Excellent test of encryption method through Python.

User input and answer return concept works well, loops iterate efficiently.

Cons: Binary information irrelevant for current game level design. Still a potential high-difficulty level concept.

Lacks basic goal of game self-generating messages and encryption keys for user to decrypt.

Notes: In next prototype, create

'''

''' Prototype 1.1 '''

#Input 1: key

def proto\_1\_1\_keyEntry():

key = int(input("Enter key: ")) #converts key input to usable integer form

while key > 26: #while loop sets keys above 26 to a proper alphabet

#number iteration

key -= 26

return key

#Input 2: plaintext message

def proto\_1\_1\_messageInput():

plaintext = input("Enter plaintext message: ")

#message = plaintext.lower()

return plaintext

#Program: Encryption

def proto\_1\_1\_mainEncryption():

ciphertext = '' #set empty strings for encrypted characters to be

binarytext = '' #entered into

message = proto\_1\_1\_messageInput()

key = proto\_1\_1\_keyEntry()

#loop to encrypt individual characters and add them to empty strings

for keys in message:

unicode\_a = ord(keys) #convert character to Unicode

unicode\_c = unicode\_a + key

new\_character = chr(unicode\_c) #convert Unicode to cipher character

extra\_enc = format(ord(new\_character), '08b') #convert cipher to binary

ciphertext += new\_character

binarytext += extra\_enc

print("CIPHERTEXT:", ciphertext)

print("BINARY REPRESNTATION:", binarytext)

input("Press Enter to end program.")

''' Prototype 1.2: Version is similar to 1.1, but with a decryption prompt for users to enter.

Users enter a message, a key to encrypt it, and then put the encrypted message into the next prompt.

A list of potential decrypted messages appears, based on a conventional 26 or less integer key.

The user then looks for the most logical message and verifies that the derypted message and key are consistent.

Pro: User prompt further confirmed

'''

#Input 1: key

def proto\_1\_2\_keyEntry():

key = int(input("Enter key: ")) #converts key input to usable integer form

while key > 26: #while loop sets keys above 26 to a proper alphabet

#number iteration

key -= 26

return key

#Input 2: plaintext message

def proto\_1\_2\_messageInput():

plaintext = input("Enter plaintext message: ")

#message = plaintext.lower()

return plaintext

#Program: Encryption

def proto\_1\_2\_mainEncryption():

ciphertext = '' #set empty strings for encrypted characters to be

message = proto\_1\_2\_messageInput()

key = proto\_1\_2\_keyEntry()

#loop to encrypt individual characters and add them to empty strings

for keys in message:

unicode\_a = ord(keys) #convert character to Unicode

unicode\_c = unicode\_a + key

new\_character = chr(unicode\_c) #convert Unicode to cipher character

ciphertext += new\_character

print("CIPHERTEXT:", ciphertext)

def proto\_1\_2\_decrypt():

message = input("Enter encrypted message: ")

key = 0

while key <=26:

messages = ''

for char in message:

new = ord(char)

dec1 = new - key

dec2 = chr(dec1)

messages += dec2

print("Decryption",key,":",messages)

print()

key +=1

input("Press Enter to end program")

''' EXECUTION CHOICE MENU '''

def menu():

print("Please choose a version to test")

print("Enter integer 1 for Prototype 1.1")

print("Enter integer 2 for Prototype 1.2")

key = int(input("Enter choice here: "))

if key == 1:

proto\_1\_1\_mainEncryption()

elif key == 2:

proto\_1\_2\_mainEncryption()

proto\_1\_2\_decrypt()

ans = str("Please press Y to continue, or enter N to end the program.")

if ans == "Y":

menu()

elif ans == "N":

exit()

menu()