# Program:      WK5 Lab - Decrypt an Encrypted Message

# Programmer:   Charlie Ritter

# Date:         5/1/2025

# Purpose:      expand upon the various cypher techniques described in the lecture

A screen shot of a computer program

AI-generated content may be incorrect.

''' Decrypt the message: FSkkYRMgKC9hKC9hEjEgKC9hMjUgODJhLCAoLy04YS4vYTUpJGExLSAoL2A=

Key is likely a single Uppercase letter.

'''

import base64

from itertools import izip, cycle

from string import ascii\_uppercase as uc  # no need to hand type a list of uppercase letters!

# put the uppercase ascii in a list

def split(upper):

    return [char for char in upper]

key\_val\_list = split(uc)

# encrypt and decrypt function

# takes the data to be encrypted or decrypted as the first parameter

# the possible character(s) to be that key as the second parameter.

# and boolean to determin if the function will encode a message or decode a

# a message as the third and fourth parameters.

def xor\_crypt\_string(data, key=' ', encode=False, decode=False):

    if decode:

        # data passes to string variable encrypted and is decoded back into text

        data = base64.decodestring(data)

    # xored = empty string with a join performed on it, ord returns an integer representation

    # of a unicode character, the ints are xored and converted to a character. then that

    # character is put in the xored encrypted variable. Then to be xored as (x,y) passed

    # to the izip function as the data and cycle(key) parameters.

    xored = ''.join(chr(ord(x) ^ ord(y)) for (x,y) in izip(data, cycle(key)))

    if encode:

        return base64.encodestring(xored).strip()

    return xored

secret\_data = "FSkkYRMgKC9hKC9hEjEgKC9hMjUgODJhLCAoLy04YS4vYTUpJGExLSAoL2A="

# function to decrypt message by calling the xor\_crypt\_string function

# takes takes two parameters - the possible keys to decrypt and the message

# to be cracked, loops through each character of possible keys and prints

# the results. This works if the key is one character to crack the code.

def decrypt\_xor(keys, message):

    for d in keys:

        print("for key %s the result is: %s" % (d, xor\_crypt\_string(secret\_data, key=d, decode=True)))

    return message

# call by passing the list of possible keys and the secrete message.

decrypt\_xor(key\_val\_list, secret\_data)