Fall 2024 CIS 3362 Week One Assignment Solution

For questions 1 – 3, decode each message. The techniques used to encrypt the messages are given in parentheses right before the ciphertext. In your write-up, explain the profess you used to decrypt and include any code you might have used as an aid.

Please do NOT use websites that automatically solve ciphers as most of your grade will be based on your description of the decryption process and original code you include in your write up.

1) (shift)

```
eqapqvomdmzgwvmiozmibnittamumabmz
```

We will find the plaintext by attempting each of the 26 possible decryption shifts, printing out what the plaintext would be in each case, and eyeballing which one of the messages makes sense.

Here is a python program that does the task:

```
cipher1 = "eqapqvomdmzgwvmiozmibnittamumabmz"
cipher2 =
"ncdaodnmzgvodqzgtzvntojwmzvfviyvggjrnnopyzionojbzopnzyojoczxtxgdxivopmzjahjy
"

for key in range(26):

    print(key, end=": ")

    # Put either cipher1 or cipher2 here, as needed.
    for x in cipher1:

        # Subtract the key...in other languages we would have to add 26
        # so mod works, but in python it should work without it.
        plain = (ord(x)-ord('a') - key + 26)%26
        print(chr(plain+ord('a')), end="")

    # Go to next line.
    print()
```

The output at line 8, for key = 8 is:

8: wishingeveryoneagreatfallsemester

Cleaning this up, we get: "Wishing everyone a great fall semester." The encryption key was 8. (We can also think of this as adding 18 to decrypt instead of subtracting 8.)

2) (shift)

```
ncdaodnmzgvodqzgtzvntojwmzvfviyvggjrnnopyzionojbzopnzyojoczx
txgdxivopmzjahjy
```

Using the same program as above, but substituting cipher2 for cipher1, we find the line associated with key = 21 is

21: shiftisrelativelyeasytobreakandallowsstudentstogetusedtothecyclicnatureofmod

Cleaning this up we get: "Shift is relatively easy to break and allows students to get used to the cyclic nature of mod." The encryption key is 21. (We can also think of this as adding 5 to decrypt.)

3) (affine)

```
bjdwqbccnwdmhphuizsbcieruxsdurrizcdwsrnbtudxwsrernabnsdfwwsr
erqfwxsrerjdiducptudxnwsrpnhpwsrnrxhwrenheruwxshwwsrpfnriwdq
rqfwbjdwmrdmcrqhztduchuixsdzdfuwdulr
```

For this one, we need to cycle through all possible Affine keys, trying all 26 integers from 0 to 25 for b and all of the 12 integers in the set {1,3,5,7,9,11,15,17,19,21,23,25} for a. This list is small enough, we can just hard code it. Alternatively, we can write or call a built in gcd function to screen out invalid values of a via (if gcd(a, 26) == 1…)

It seems tedious to look at all outputs, so we'll screen the potential output for the words "the" and "and" and only print out the results that have at least one of these words.

We will cycle through the decryption keys. At the very end of the program, I will use the decryption keys to print out the original encryption keys. (I'll take the math process by hand and code it up, just using a chart of modular inverses for now.)

```
# So this fits...
cipher = "bjdwqbccnwdmhphuizsbcieruxsdurrizcdwsrnbtudxwsrernabnsdfwwsrer"
cipher = cipher + "qfwxsrerjdiducptudxnwsrpnhpwsrnrxhwrenheruwxshwwsrpfnr"
cipher = cipher + "iwdqrqfwbjdwmrdmcrqhztduchuixsdzdfuwdulr"

# modinv[x] is the mod inverse of x mod 26 if it exists, -1 otherwise.
modinv = [-1,1,-1,9,-1,21,-1,15,-1,3,-1,19,-1,-1,-1,7,-1,23,-1,11,-1,5,-
1,17,-1,25]

# Possible values of a.
aList = [1,3,5,7,9,11,15,17,19,21,23,25]

# Try all.
for a in aList:
    for b in range(26):

        plain = ""
```

```
        # Just append to plain.
        for x in cipher:
            numX = ord(x) - ord('a')
            p = (a*numX + b)%26
            letP = chr(p+ord('a'))
            plain = plain + letP

        # Look for the and and.
        if "the" in plain or "and" in plain:

            # Print possible decryption keys and potential message
            print(a,b,plain)

            # Here is the math to print out the corresponding encryption
keys.
            print("encryption keys are a=",modinv[a]," b=",(modinv[a]*(26-
b))%26, sep="")
```

Here is the output generated by the program:

```
3 5
igotbillstopayandchildrenwhoneedclothesiknowtheresfishouttherebu
twheregodonlyknowstheysaythesewatersarentwhattheyusedtobebutigot
peoplebackonlandwhocountonme
encryption keys are a=9 b=7
7 9
quehrqxxwhepgkgtncfqxnlytofetyyncxehfywqmteohfylywjqwfeshhfylyrs
hofylyuenetxkmteowhfykwgkhfywyoghylwglythofghhfykswynheryrshqueh
pyepxyrgcmetxgtnofecesthetiy
encryption keys are a=15 b=21
9 16
ztrgeziidgrubvbokhwzikanopwronnkhirgwndzforpgwnandqzdwrjggwnanej
gpwnantrkroivforpdgwnvdbvgwndnpbgnadbanogpwbggwnvjdnkgrenejgztrg
unruinebhfroibokpwrhrjogroln
encryption keys are a=3 b=4
11 8
tdpqcteevqpkhrhusxytesanubypunnsxepqynvtjupbqynanvitvyplqqynancl
qbynandpspuerjupbvqynrvhrqynvnbhqnavhanuqbyhqqynrlvnsqpcnclqtdpq
knpkenchxjpuehusbypxpluqpuzn
encryption keys are a=19 b=4
15 14
dthgudssbghmpfpcezydsewjcvyhcjjezshgyjbdnchvgyjwjbodbyhlggyjwjul
gvyjwjthehcsfnchvbgyjfbpfgyjbjvpgjwbpwjcgvypggyjflbjeghujulgdthg
mjhmsjupznhcspcevyhzhlcghcxj
encryption keys are a=7 b=6
17 18
jprcejaafcrohnhuybmjayivutmruvvybarcmvfjdurtcmvivfsjfmrzccmvivez
ctmvivpryruandurtfcmvnfhncmvfvthcvifhivuctmhccmvnzfvycrevezcjprc
ovroavehbdruahuytmrbrzucruxv
```

```
encryption keys are a=23 b=2
23 12
jldyqjggzydcrtreopkjgoanevkdennopgdyknzjhedvyknanzmjzkdxyyknanqx
yvknanldodegthedvzykntzrtyknznvrynazraneyvkryykntxznoydqnqxyjldy
cndcgnqrphdegreovkdpdxeydefn
encryption keys are a=17 b=4
```

Luckily the first try worked and we get the following output:

igotbillstopayandchildrenwhoneedclothesiknowtheresfishouttherebutwheregodonlyknowstheysaythesewatersarentwhattheyusedtobebutigotpeoplebackonlandwhocountonme

Cleaning it up we get:

I got bills to pay and children who need clothes. I know there's fish out there but where God only knows. They say these waters aren't what they used to be, but I got people back on land who count on me.

The decryption keys were a = 3, b = 5. The corresponding encryption keys were a = 9 and b = 7.

Incidentally, these are lyrics from Billy Joel's song, "Downeaster Alexa"

(https://en.wikipedia.org/wiki/The_Downeaster_%22Alexa%22#:~:text=%22The%20Downeaster%20'Alexa'%22,3%20album%20in%201997.)

4) Using the affine cipher with encryption keys a = 15 and b = 17, here is a sample message to encrypt

```
dearscottandjacobthanksforgradingmyhomeworkireallyappreciateit
```

Here is the program (basically the one above with the a and b loops removed) to generate the ciphertext:

```
# Question 4 here.
plain = "dearscottandjacobthanksforgradingmyhomeworkireallyappreciateit"

# Just encrypt with hard-coded key given.
for x in plain:
    numX = ord(x) - ord('a')
    c = (17*numX + 15)%26
    print(chr(c+ord('a')),end="")
print()
```

The corresponding output was:

ofpsjxtaapcompxtgaepcdjwtsnspovcnlhetlfztsdvsfpuuhpkksfxvpafva