1. 7503 mod 81 = **51**
2. LOOKUPINTHEAIRITSABIRDITSAPLANEITSSUPERMAN
3. 0, 13
   1. = 30
   2. = 4,000
   3. ­ = 1,029,000
   4. a’ = 5, b’ = 21.

**Work**

**Question 2:**

Python script to break encryption with output.

ct = 'BEEAKFYDJXUQYHYJIQRYHTYJIQFBQDUYJIIKFUHCQD' #A = 65

alpha = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

output = open('output.txt', 'w')

for shift in range(1, 27):

  for ch in ct:

    ind = alpha.index(ch)

    output.write(alpha[(ind + shift) % 26])

  output.write('\n')



**Question 3:**

mod = 26

alpha = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

ct = 'THISISATEST'

enc = ['THISISATEST']

dec = [enc[0]]

involutory = []

#Encrypt

for shift in range(1, mod):

  enc.append('')

  for ch in ct:

    ind = alpha.index(ch)

    enc[shift] += alpha[(ind + shift) % mod]

#Shift Again Using Same Shift

for shift in range(1, mod):

  dec.append('')

  currDec = enc[shift]

  for ch in currDec:

    ind = alpha.index(ch)

    dec[shift] += alpha[(ind + shift) % mod]

#Find instances where encrypted ended up back at the start using the same shift

for i in range(len(dec)):

  if dec[i] == ct:

    involutory.append(i)

print(involutory)

**Output:** [0, 13]

**Question 4:**

I wrote this script to find all the invertible elements and their inverses given a modulo. I used this and multiplied the number *n* that came out by the given modulo to get the keys possible.

import math

mod = 1225

invertibleAndInverses = {'Invertible': [], 'Inverse': []}

for i in range(mod):

    if math.gcd(i, mod) == 1:

        #A number is invertible if the gcd of the number and the modulo is 1.

        invertibleAndInverses['Invertible'].append(i)

        for j in range(mod):

            if (i \* j) % mod == 1:

                invertibleAndInverses['Inverse'].append(j)

print(f'Invertibles: {invertibleAndInverses["Invertible"]}')

print(f'Inverses:    {invertibleAndInverses["Inverse"]}')

print(f'Number n:    {len(invertibleAndInverses["Invertible"])}')