#1.5

```
from itertools import count
from string import ascii_lowercase

cipher_text = 'beeakfydjxuqyhyjiqryhtyjiqfbqduyjiikfuhcqd'
results_file = open('results.txt', 'w')
letters_to_numbers = dict(zip(ascii_lowercase, count(0)))
numbers_to_letters = dict(zip(count(0), ascii_lowercase))
cipher_text_numbers = [letters_to_numbers[letter] for letter in
cipher_text]
for i in range(1,26):
    shifted_cipher = [(num - i)%26 for num in cipher_text_numbers]
    plain_text = [numbers_to_letters[num] for num in shifted_cipher]
    results_file.write('\n')
    results_file.write('Shift Amount: ' + str(i))
    results_file.write('\n')
    results_file.write(''.join(plain_text))
```

```

Shift Amount: 1
addzjexciwtpxgxihpqxgsxihpeapctxihhjetgbpc

Shift Amount: 2
zccyidwbhvsowfwhgopwfrwhgodzobswhggidsfaob

Shift Amount: 3
ybbxhcvagurnvevgfnoveqvgfncynarvgffhcrezna

Shift Amount: 4
xaawgbuzftqmudufemnudpufembxmzqufeegbqdymz

Shift Amount: 5
wzzvfatyespltctedlmtcotedlawlypteddfapcxly

Shift Amount: 6
vyyuezsxdroksbsdcklsbnsdckzvkxosdccezobwkx

Shift Amount: 7
uxxtdyrwcqnjrarcbjkramrcbjyujwnrcbbdynavjw

Shift Amount: 8
twwscxqvbpmiqzqbaijqzlqbaixtivmqbaacxmzuiv

Shift Amount: 9
svvrbwpuaolhpypazhipykpazhwshulpazzbwlythu

Shift Amount: 10
ruuqavotznkgoxozyghoxjozygvrgtkozyyavkxsgt

Shift Amount: 11
```

```
qttpzunsymjfnwnyxfgnwinyxfuqfsjnyxxzujwrfs

Shift Amount: 12
pssoytmrxliemvmxwefmvhmxwetperimxwwytivqer

Shift Amount: 13
orrnxslqwkhdlulwvdeluglwvdsodqhlwvvxshupdq

Shift Amount: 14
nqqmwrkpvjgcktkvucdktfkvucrncpgkvuuwrgtocp

Shift Amount: 15
mpplvqjouifbjsjutbcjsejutbqmbofjuttvqfsnbo

Shift Amount: 16
lookupintheairitsabirditsaplaneitssuperman

Shift Amount: 17
knnjtohmsgdzhqhsrzahqchsrzokzmdhsrrtodqlzm

Shift Amount: 18
jmmisnglrfcygpgrqyzgpbgrqynjylcgrqqsncpkyl

Shift Amount: 19
illhrmfkqebxfofqpxyfoafqpxmixkbfqpprmbojxk

Shift Amount: 20
hkkgqlejpdawenepowxenzepowlhwjaepooqlaniwj

Shift Amount: 21
gjjfpkdioczvdmdonvwdmydonvkgvizdonnpkzmhvi

Shift Amount: 22
fiieojchnbyuclcnmuvclxcnmujfuhycnmmojylguh

Shift Amount: 23
ehhdnibgmaxtbkbmltubkwbmltietgxbmllnixkftg

Shift Amount: 24
dggcmhaflzwsajalkstajvalkshdsfwalkkmhwjesf

Shift Amount: 25
cffblgzekyvrzizkjrsziuzkjrgcrevzkjjlgvidre
```

Therefore the shift amount is 16 and the plain text is: look up in the air its a bird its a plane its superman

#1.6

```
from itertools import count
from string import ascii_lowercase
```

```
plain_text = 'july'
results_file = open('results.txt', 'w')
letters_to_numbers = dict(zip(ascii_lowercase, count(0)))
numbers_to_letters = dict(zip(count(0), ascii_lowercase))
plain_text_numbers = [letters_to_numbers[letter] for letter in plain_text]
for i in range(0, 26):
    #encrypt the plain text by shifting by some number
    cipher_numbers = [(num + i)%26 for num in plain_text_numbers]
    #try to decrypt the plain text by shifting forward by the same number (encrypt function = decrypt function)
    decrypted_cipher_numbers = [(num + i)%26 for num in cipher_numbers]
    attempted_plain_text = [numbers_to_letters[num] for num in decrypted_cipher_numbers]

    if ''.join(attempted_plain_text) == plain_text: #if we decrypt print which key values work
        print('At shift = ' + str(i) + ':')
        print('Plain text: ' + plain_text)
        print('Attempted Plain Text Decrypt: ' + ''.join(attempted_plain_text))
```

```
At shift = 0:

Plain text: july

Attempted Plain Text Decrypt: july

At shift = 13:

Plain text: july

Attempted Plain Text Decrypt: july
```

Therefore the involutory keys are 0 and 13

#1.7

```
from math import gcd

results_file = open('results.txt', 'w')
user_input = input('Enter the m value: ')
while user_input != 'q':
    m = int(user_input)
    results_file.write('Enter the m value: ' + str(m) + '\n')
    keys = 0
    for i in range(1,m):
        if gcd(i, m) == 1:
            keys = keys + 1
    keys = keys * m
    results_file.write(str(keys) + '\n')
    user_input = input('Enter the m value: ')
```

```
Enter the m value: 30

240

Enter the m value: 100

4000

Enter the m value: 1225

1029000
```

Therefore the number of keys are 240 at m = 30, 4000 at m = 100, and 1029000 at 1225