Prob.1

$(a,b) = (9,5)\ ;\ \gcd(9,26) = 1\ \rightarrow\ (a^{-1}, -b) = (3,11)$

- $26 = 2(9) + 8\ ,\ 9 = 1(8) + 1$
- $\rightarrow 3(9) - 26 = 1 \rightarrow 9^{-1} = 3$

$(a,b) = (8,12)\ ;\ \gcd(8,26) = 2 \neq 1\ \rightarrow$ cant be a key for affine cipher

In [13]:
```
(3*-5)%26
```

Out[13]: 11

Prob.2

(a) $e_k(P) = d_k(C)$
For shift cipher to be self-inverse : $\{x+b = y-b\} \rightarrow \{b = -b\} \rightarrow \{b=0\}$.
For affine cipher to be self-inverse :
$\{ax+b = a^{-1}(y-b)\} \rightarrow \{ax+b = a^{-1}y - a^{-1}b\} \rightarrow \{a = a^{-1}\}\&\{b = -a^{-1}b\} \rightarrow \{a^2 = 1\}, \{ab =$ -
,
$\rightarrow \{a = \{1,25\}\}\ ,\ \{ab+b=0\} \rightarrow \{b(a+1) = 0\} \rightarrow \{b = \{0,13\}\ \&\ a = 1\ or\ b \in \mathbb{Z}_{26}\ \&\ a = 25\}$
:

In [16]:
```
from math import gcd

ab_pairs =[]
for a in range(26):
    if gcd(a,26) != 1:
        continue
    if ((a * a) % 26 != 1):
        continue
    for b in range(26):
        if(((a + 1)*b) % 26 == 0):
            ab_pairs.append((a,b));

print("(a,b) =", ab_pairs, "\n len(a,b) =", len(ab_pairs))
```

```
(a,b) = [(1, 0), (1, 13), (25, 0), (25, 1), (25, 2), (25, 3), (25, 4), (25, 5), (25, 6), (25, 7), (25, 8), (25, 9), (25, 10), (25, 11), (25, 12), (25, 13), (25, 14), (25, 15), (25, 16), (25, 17), (25, 18), (25, 19), (25, 20), (25, 21), (25, 22), (25, 23), (25, 24), (25, 25)]
 len(a,b) = 28
```

(b)
If $a^2 \mod pq = 1$ then $a \mod p = \pm 1$ or $a \mod q = \pm 1$
For $\{b(a+1) \mod n = 0\} \rightarrow \{a \mod n = (n-1)\}$ or
$\{b \mod n = 0\} \rightarrow \#\gcd(a+1,n) = \#\gcd(2,pq) = 1$
For $\{a \mod n = 1\}$ and $\{b \mod n = 0\}$ and for $\{a \mod n = (n-1)\}$,
$b \in \mathbb{Z}_{pq} \rightarrow \#\gcd(a+1,n) = n$
For $\{a \mod p = 1\}$ and $\{a \mod q = -1\} \rightarrow \#\gcd(a+1,n) = q$
For $\{a \mod p = -1\}$ and $\{a \mod 1 = 1\} \rightarrow \#\gcd(a+1,n) = p$
$\rightarrow \#total\ solutions = n + q + p + 1\ \square$

(c) For $n=0$ and $n=1$ there is only one permutation and for $n=2$ there is 2 permutation that are self-inverse, it is proven that for n alphabetic substitution cipher there is $I_n = I_{n-1} + (n-1)I_{n-2}$ possible self-inverse permutation:

In [15]:
```python
def I(n):
    if(n==0 or n==1):
        return 1
    return I(n-1) + (n-1)*I(n-2)

print("total number of 26 alphabet self-inverse cipher: ",I(26))
```

```
total number of 26 alphabet self-inverse cipher:  532985208200576
```

Prob.3

Shift cipher:

1. for all $0 < k < 25$
2. decrypt massage using shift cipher method
3. if frequency match its shift cipher

Affine:

1. for all possible pairs of (a,b) that gcd(a,26) is 1:
2. decrypt massage using affine method
3. if frequency match its affine cipher

Substitution:

1. for all words check the distribution
2. if the distribution of two word perserved then its substitution cipher

Permutation:

1. for all words check the distribution
2. if the distribution of two word is not perserved then its permutation cipher

Vigenere:

1. if Index of coincedence suggest a polyalphabetic cipher:
2. split cipher to m group of words
3. compute I.C. to reach 0.066 and return m
4. check the frequency at each group if it match its Vigenere cipher

Prob.4

(a)

- In shift shift if we consider k_1 and k_2 we can write encryption function as $e_{K_2}(e_{k_1}(x))$ which is just $(x + k_1) + k_2 = e_{k_1+k_2}$ so it would not be more secure than a shift cipher.

- Lets first validate two step affine cipher:

1. $\gcd(a_1, m) = 1$, $\gcd(a_2, m) = 1$ should result in $\gcd(a_1 a_2, m) = 1$ scince m is not measurable by either a_1 or a_2 divisors.
2. new encryption function will become: $\{a_2(a_1 x + b_1) + b_2 = a_2 a_1 x + a_2 b_1 + b_2\} = a_3 x + b_3$ the new encryption function is also an affine cipher.

- for subtitution cipher also $e_{\pi_1}(e_{\pi_2}(x)) = e_{\pi_3(x)}$

(b)

- Applying Kasiski will indicate the least common multiple of two periods $\gcd(d_i) = lcm(k_1, k_2)$. so for guessing k_1 and k_2 we need to try all $\gcd(d_i)$ divisor and check the frequency to approve our guess.

Prob.5

(a)

- $c_1 \parallel c_2 = (m_1 \oplus k) \parallel (m_2 \oplus k) \rightarrow c_1 \oplus c_2 = m_1 \oplus m_2$ so attacker can distinguish between two massages:

- if adversery chooses $m^0 = m_1^0 \parallel m_2^0$ and $m_1^1 \parallel m_2^1$ challanger choose randomly to encrypt one of m^0 or m^1

- attacker will now from $c_1 \oplus c_2 = m_1 \oplus m_2$ that which massage was encrypted.

- so the **Indistinguishability Advantage** becomes:
  $Adv = \Pr[experiment\ success] - 0.5 = 1 - 0.5 = 0.5$. (b)

- it is impossible $c_1 \oplus c_2 \neq m_1 \oplus m_2$ to happen after observing "c" that is $\Pr[M = m|C = c] = 0$ hence $\Pr[C = c|M = m_1] = \Pr[C = c|M = m_2]$ not necesserly holding.

In [ ]: