Question 2

{Note that if there is not an earlier description of affine ciphers in the book, it must be included here.}

Implement Sage functions that perform affine cipher Encryption / Decryption, given a key that consists of a pair of integers (a,b), both in {1,2,..,25} with a not divisible by 2 or 13. The functions should work on strings, and leave any non-alphabetic characters unchanged. Show the operation of your functions on an example.

Solution to Question 2

Implement Sage functions that perform affine cipher Encryption / Decryption, given a key that consists of a pair of integers (a,b), both in {1,2,..,25} with a not divisible by 2 or 13. The functions should work on strings, and leave any non-alphabetic characters unchanged. Show the operation of your functions on an example.

def AffineEncrypt(k, plaintext):

(a, b) = k

ciphertext = ""

for j in xrange(len(plaintext)):

p = plaintext[j]

if is\_alphabetic\_char(p):

x = (a\*char\_to\_num(p) + b) % 26

c = num\_to\_char(x)

else:

c = p

ciphertext += c

return ciphertext

def AffineDecrypt(k, ciphertext):

(a, b) = k

ainv = xgcd(a,26)[1]

plaintext = ""

for j in xrange(len(ciphertext)):

c = ciphertext[j]

if is\_alphabetic\_char(c):

x = ainv\*(char\_to\_num(c) - b) % 26

p = num\_to\_char(x)

else:

p = c

plaintext += p

return plaintext

Here is an example of this code running on an example:

sage: plaintext = "john smith is the culprit!"

sage: k = (17, 8)

sage: ciphertext = AffineEncrypt(k, plaintext); ciphertext

'fmxv ceotx oc txy qkndlot!'

sage: AffineDecrypt(k, ciphertext)

'john smith is the culprit!'