Question 4)

Implement Sage functions to perform encryption/decryption with 2x2 Hill Cipher. The key should be an invertible Sage matrix over the integers mod 26. Do not just call the built in sage functionality for the Hill cipher. Show the operation of your functions on a plaintext of your choice.

Solution to Question 4)

Implement Sage functions to perform encryption/decryption with 2x2 Hill Cipher. The key should be an invertible Sage matrix over the integers mod 26. Do not just call the built in sage functionality for the Hill cipher. Show the operation of your functions on a plaintext of your choice.

def validate\_2x2\_hill\_key(M):

if (M not in MatrixSpace(IntegerModRing(26), 2)):

raise ValueError, "M must be a 2x2 matrix over the integers mod 26."

if (not M.is\_invertible()):

raise ValueError, "M must be an invertible matrix."

return True

def string\_to\_2x2\_hill\_block(in\_string):

if (2 != len(in\_string)):

raise ValueError, "in\_string must be of length 2."

pair = (char\_to\_num(in\_string[0]), char\_to\_num(in\_string[1]))

FM = FreeModule(R, 2)

return FM(pair)

def hill\_2x2\_process\_text(intext, M):

# strip out all nonalphabtic characters

intxt\_alpha\_only = ""

for j in xrange(len(intext)):

if is\_alphabetic\_char(intext[j]):

intxt\_alpha\_only += intext[j]

# pad if necessary

if (1 == len(intxt\_alpha\_only) % 2):

intxt\_alpha\_only += 'a'

num\_blocks = len(intxt\_alpha\_only)/2

outtext = ""

for j in xrange(num\_blocks):

str\_block = intxt\_alpha\_only[2\*j] + intxt\_alpha\_only[2\*j+1]

vec = string\_to\_2x2\_hill\_block(str\_block)

out\_vec = M\*vec

outtext += num\_to\_char(out\_vec[0]) + num\_to\_char(out\_vec[1])

return outtext

def hill\_2x2\_encrypt(plaintext, K):

validate\_2x2\_hill\_key(K)

return hill\_2x2\_process\_text(plaintext, K)

def hill\_2x2\_decrypt(ciphertext, K):

validate\_2x2\_hill\_key(K)

Kinv = K.inverse()

return hill\_2x2\_process\_text(ciphertext, Kinv)

The following is the operation of these functions on a plaintext:

sage: pt = 'ilikepurpledinosaurs'

sage: ct = hill\_2x2\_encrypt(pt, K); ct

'nvembfdxtixbfnsgyyct'

sage: hill\_2x2\_decrypt(ct, K)

'ilikepurpledinosaurs'