Question 1

In the examples functions for the Euclidean and extended Euclidean GCD, the first input must be greater than the second. Furthermore, each argument must be a positive integer. Implement these functions such that these assumptions need not be made about the input, also for the extended Euclidean GCD, if the gcd of a and b is 1, return a inverse mod b and b inverse mod a [Your Sage functions may call the example Sage functions, or you may write these implementations from scratch. Do not merely call the built in Sage functionality.] Show your functions work on a few inputs.

Solution to Question 1

In the examples functions for the Euclidean and extended Euclidean GCD, the first input must be greater than the second. Furthermore, each argument must be a positive integer. Implement these functions such that these assumptions need not be made about the input, also for the extended Euclidean GCD, if the gcd of a and b is 1, return a inverse mod b and b inverse mod a [Your Sage functions may call the example Sage functions, or you may write these implementations from scratch. Do not merely call the built in Sage functionality.] Show your functions work on a few inputs.

def euclidean\_gcd(a,b):

A = abs(a)

B = abs(b)

if (A == B):

return A

if (B > A):

temp = B

B = A

A = temp

return EUCLID(A, B)

def extended\_euclidean\_gcd(a,b):

sign\_coef\_A = 1

sign\_coef\_B = 1

if (a < 0):

sign\_coef\_A = -1

if (b < 0):

sign\_ceof\_B = -1

A = abs(a)

B = abs(b)

switched = False

if (A == B):

return (1, 1, 0)

if (B > A):

switched = True

temp = B

B = A

A = temp

(g, x) = EXTENDED\_EUCLID(A, B)

if (None == x):

w = None

else:

w = (g - x\*B)/A

if (switched):

temp = x

x = w

w = temp

if (None != x):

w = sign\_coef\_A \* w

x = sign\_coef\_B \* x

return (g, w, x)