Question 2

Suppose that polynomials are represented by lists of coefficients, where the coefficient at index i is the coefficient of xi. Using this representation, write Sage functions that perform the following polynomial operations (don’t just call the underlying sage functions):

1. Scalar multiply, given a scalar c, and a polynomial f, computes c∙f.
2. Addition, given two polynomials f, g, computes h = f + g.
3. Subtraction, given two polynomials f, g computes h = f – g.
4. Multiplication, given two polynomials f, g computes h = f\*g.
5. For each of the above functions that you wrote show the output of the function on 1 set of inputs.

Solutions to Question 2

Suppose that polynomials are represented by lists of coefficients, where the coefficient at index i is the coefficient of xi. Using this representation, write Sage functions that perform the following polynomial operations (don’t just call the underlying sage functions):

1. Scalar multiply, given a scalar c, and a polynomial f, computes c∙f.

def polynomial\_scalar\_multiply(f, c):

g = [ c\*f[j] for j in xrange(len(f)) ]

return g

1. Addition, given two polynomials f, g, computes h = f + g.

def polynomial\_addition(f, g):

max\_len = max(len(f), len(g))

temp\_f = [f[j] for j in xrange(len(f))]

temp\_g = [g[j] for j in xrange(len(g))]

if (len(f) < max\_len):

temp\_f.extend([0 for j in xrange(max\_len - len(f))])

if (len(g) < max\_len):

temp\_g.extend([0 for j in xrange(max\_len - len(g))])

h = [temp\_f[j] + temp\_g[j] for j in xrange(max\_len)]

return h

1. Subtraction, given two polynomials f, g computes h = f – g.

def polynomial\_subtraction(f, g):

return polynomial\_addition(f, polynomial\_scalar\_multiply(g, -1))

1. Multiplication, given two polynomials f, g computes h = f\*g.

def polynomial\_multiplication(f, g):

len\_prod = len(f) + len(g) - 2

h = [0 for j in xrange(len(f) + len(g) - 1)]

for j in xrange(len(f)):

for k in xrange(len(g)):

h[j+k] += f[j]\*g[k]

return h

1. For each of the above functions that you wrote show the output of the function on 1 set of inputs.

sage: f = [GF(17).random\_element() for j in xrange(6)]; f

[0, 13, 2, 16, 0, 1]

sage: g = [GF(17).random\_element() for j in xrange(5)]; g

[5, 14, 14, 9, 13]

sage: c = GF(17).random\_element(); c

6

sage: polynomial\_scalar\_multiply(f, c)

[0, 10, 12, 11, 0, 6]

sage: polynomial\_addition(f, g)

[5, 10, 16, 8, 13, 1]

sage: polynomial\_subtraction(f, g)

[12, 16, 5, 7, 4, 1]

sage: polynomial\_multiplication(f, g)

[0, 14, 5, 1, 12, 8, 14, 1, 9, 13]