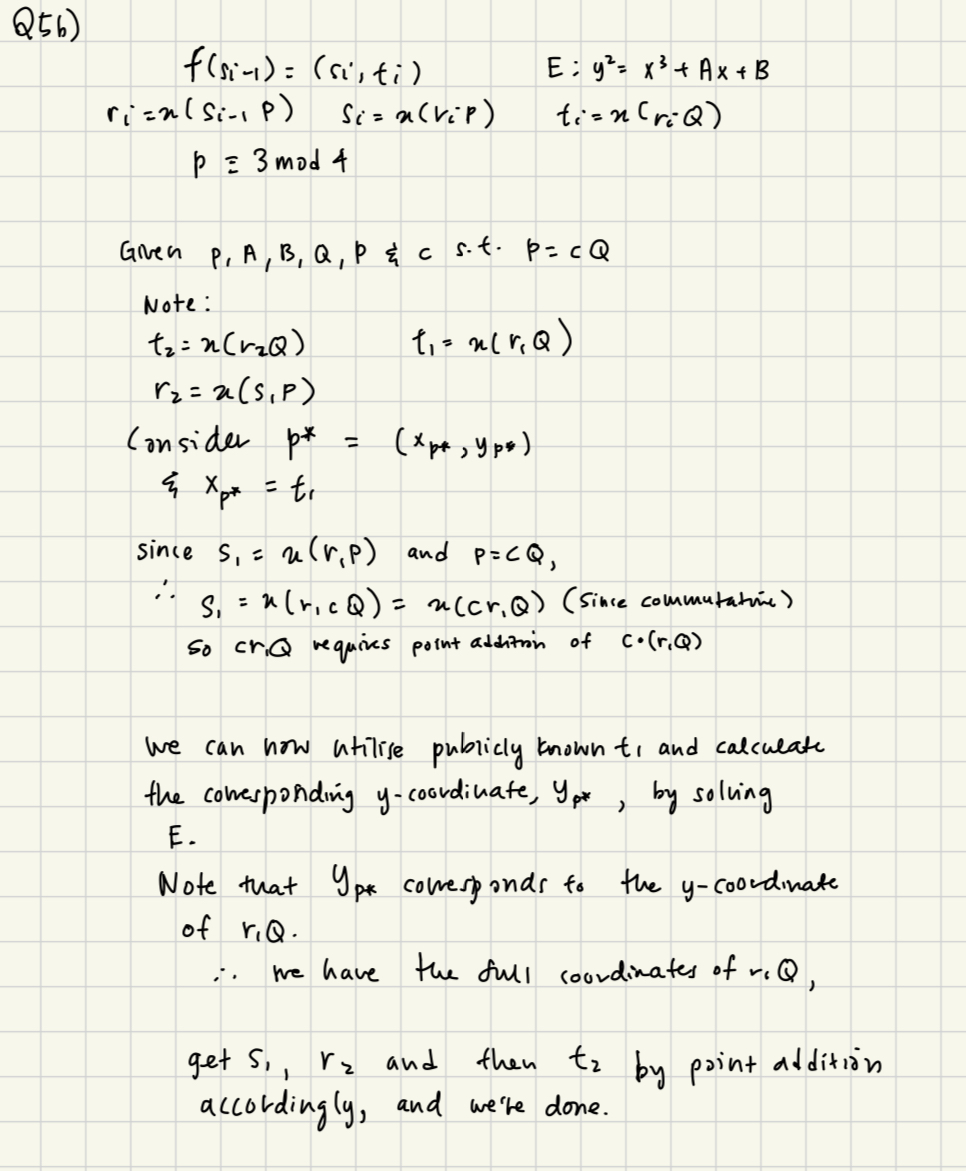
watIAM: sh2yap

student ID: 21111395

1. 15, 18, 3 (Refer to appendix for full codes)

A blue background with white numbers

Description automatically generated

1. 
2. s1 = 102, t2 =37 <https://andrea.corbellini.name/ecc/interactive/modk-add.html>
3. Use well-established elliptic curves.

Appendix:

class EllipticCurve:

def \_\_init\_\_(self, a, b, p):

self.a = a

self.b = b

self.p = p

def is\_point\_on\_curve(self, point):

x, y = point

return (y\*\*2) % self.p == (x\*\*3 + self.a \* x + self.b) % self.p

def point\_addition(self, p, q):

if p == (float('inf'), float('inf')):

return q

if q == (float('inf'), float('inf')):

return p

x\_p, y\_p = p

x\_q, y\_q = q

if p != q:

# Calculate the slope

s = ((y\_q - y\_p) \* pow(x\_q - x\_p, -1, self.p)) % self.p

else:

# Point doubling

s = ((3 \* x\_p\*\*2 + self.a) \* pow(2 \* y\_p, -1, self.p)) % self.p

# Calculate the new x-coordinate

x\_r = (s\*\*2 - x\_p - x\_q) % self.p

# Calculate the new y-coordinate

y\_r = (s \* (x\_p - x\_r) - y\_p) % self.p

return (x\_r, y\_r)

def scalar\_multiply(self, k, point):

result = (float('inf'), float('inf'))

for \_ in range(k.bit\_length()):

if k & 1:

result = self.point\_addition(result, point)

point = self.point\_addition(point, point)

k >>= 1

return result

# Example usage

a = 2

b = 3

p = 19 # Prime modulus

# Create an elliptic curve instance

curve = EllipticCurve(a, b, p)

# Define points P and Q on the curve

P = (1, 14)

Q = (3, 13)

# Check if points are on the curve

print(f"Is P on the curve? {curve.is\_point\_on\_curve(P)}")

print(f"Is Q on the curve? {curve.is\_point\_on\_curve(Q)}")

# Initializations

s = 2

# Do iteration

for i in range(3):

r = curve.scalar\_multiply(s, P)[0]

s = curve.scalar\_multiply(r, P)[0]

t = curve.scalar\_multiply(r, Q)[0]

print("i: ", i, "t: ", t)